LogTransfer: Cross-System Log Anomaly Detection for Software Systems with Transfer Learning

Rui Chen¹, Shenglin Zhang¹, Dongwen Li¹, Yuzhe Zhang¹, Fangrui Guo¹, Weibin Meng², Dan Pei², Yuzhi Zhang¹, Xu Chen¹, Yuqing Liu¹

¹Nankai University, ²Tsinghua University
Software systems are playing important roles in daily life.

Reliability as well as availability are highly demanded in modern software systems.

Traffic will increase more than three times.

System anomalies will cause degradation, impact revenue and user experience.

How to comprehensively and precisely detect anomalies has brought about widespread attention!

System anomaly

An instance of a system anomaly
System logs are unstructured text.

A log message is composed of constant part and variable part.

```
......
[SIF pica_sif] Interface te-1/1/11, changed state to down.
......
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable Part</strong></td>
<td>Te-1/1/11</td>
</tr>
<tr>
<td><strong>Constant Part</strong></td>
<td>Interface *** changed state to down</td>
</tr>
</tbody>
</table>
### Previous methods

<table>
<thead>
<tr>
<th>Type</th>
<th>Input of Template</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>Template index</td>
<td>Decision tree</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SVM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CNN-based model</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linear regression</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>Template count</td>
<td>PCA</td>
</tr>
<tr>
<td></td>
<td>Template index</td>
<td>DeepLog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LogCluster</td>
</tr>
<tr>
<td></td>
<td>Template index</td>
<td>Isolation forest</td>
</tr>
<tr>
<td></td>
<td>Template embedding</td>
<td>Invariant mining</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LogAnomaly</td>
</tr>
</tbody>
</table>

**Supervised methods**: Massive labelling efforts & no semantics information of logs.

**Unsupervised methods**: Suffering from low accuracy in real-world service systems.
System logs in anomaly detection

**Service Type A**

Nov 6 20:06:32 192.168.190.65 %10IFNET/3/LINK_UPDOWN(l):
GigabitEthernet1/0/10 link status is UP.

......

Nov 6 20:06:32 192.168.190.65 %10IFNET/3/LINK_UPDOWN(l):
GigabitEthernet1/0/10 link status is DOWN.

**Service Type B**

Feb 14 00:37:36 192.168.191.79 2018: [SIF pica_sif]Interface te-1/1/11,
changed state to down

......

Feb 14 00:37:38 192.168.191.79 2018: [SIF pica_sif]Interface te-1/1/11,
changed state to up

Different type of systems/services event generate log sequences with similar semantics.
Can we transfer anomalous patterns from one software system to another one?
Different types of systems are different in log syntax.

Service Type A

Service Type B
Challenges

Noises in anomalous log sequences

Service Type A

*** logined the switch
*** logouted from the switch
PICALIBCOMM pica_login Fan is plugged in
PICALIBCOMM pica_login RPSU is plugged in serial number ***
Redundancy power supply unit RPSU is plugged in serial number ***
Receive SFP_PRE message module plugged into port ***

10SHELL SHELL_LOGINFAIL TELNET user *** failed to log in from ***
10DEVM POWER REMOVED Trap cPowerRemoved power ID is ***
10SHELL LOGIN Trap cLogIn *** login from ***
10SHELL LOGOUT Trap cLogOut *** logout from ***
10SHELL SHELL_CMD Task IPAddr User *** Command is ***
10LLDP LLDP_CREATE_NEIGHBOR New neighbor created on Port *** ID is ***
Framework of LogTransfer

**Representation construction**

- Logs of source system
- Logs of target system
- Template vectors

**Template vector sequences of source system**

**Template vector sequences of target system**

**An accurate representation construction method**

**Transfer learning**

- Labels of source system
- Labels of target system

**Offline model training process**

- Anomaly detection model for target system
• **Building the template embedding:**
  Containing semantic and syntactic information of a log entry.

<table>
<thead>
<tr>
<th>Raw system log:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SIF pica_sif] Interface te-1/1/11, changed state to down</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Log template (with FT-tree):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface *, changed state to down</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word embeddings (with Glove):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface: $V_1 \rightarrow (v_{11}, v_{12}, v_{13}, \ldots, v_{1n})$</td>
</tr>
<tr>
<td>changed: $V_2 \rightarrow (v_{21}, v_{22}, v_{23}, \ldots, v_{2n})$</td>
</tr>
<tr>
<td>state: $V_3 \rightarrow (v_{31}, v_{32}, v_{33}, \ldots, v_{3n})$</td>
</tr>
<tr>
<td>to: $V_4 \rightarrow (v_{41}, v_{42}, v_{43}, \ldots, v_{4n})$</td>
</tr>
<tr>
<td>down: $V_5 \rightarrow (v_{51}, v_{52}, v_{53}, \ldots, v_{5n})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Template embedding:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface *, changed state to down:</td>
</tr>
<tr>
<td>t $\rightarrow$ average($V_1 + V_2 + V_3 + V_4 + V_5$)</td>
</tr>
</tbody>
</table>

**Example of how to generate a template embedding.**

**the objective function of Glove:**

\[
J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2
\]
Transfer learning

A novel transfer learning method.

Representation construction

Logs of source system

Template vector sequences of source system

Template vector sequences of target system

Labels of source system

Transfer learning

Anomaly detection model for target system

Logs of target system
Transfer learning

- Fully Connected Network for anomaly detection
- LSTM networks to extract the pattern of log sequences
- Template vector sequences with syntactic and semantic info.
Transfer learning

Source system

Target system

Labels of source system

Shared fully connected networks

Labels of target system

Source LSTM networks

Target LSTM networks

Template vector sequences of source system

Template vector sequences of target system

Source system

Target system
Framework of LogTransfer

Online anomaly detection process

- New logs of target system
- Template vector sequences
- Target LSTM networks
- Shared fully connected networks
- Anomaly detection result
### Dataset introduction

Three switch system log from a top cloud service provider

The Hadoop application dataset & the HDFS dataset

<table>
<thead>
<tr>
<th>Source of dataset</th>
<th>Type of system</th>
<th>Source of labels</th>
<th># chunks</th>
<th># anomalous chunks</th>
<th># switches/log files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch</td>
<td>Type A</td>
<td>Real-world anomalies</td>
<td>2,345,646</td>
<td>6,406</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Type B</td>
<td></td>
<td>49,946</td>
<td>1,096</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Type C</td>
<td></td>
<td>525,427</td>
<td>4,939</td>
<td>21</td>
</tr>
<tr>
<td>Hadoop application</td>
<td>PageRank &amp; WordCount</td>
<td>Manually injected</td>
<td>121,878</td>
<td>73,936</td>
<td>1008</td>
</tr>
<tr>
<td>Hadoop file system</td>
<td>HDFS</td>
<td>Real-world anomalies</td>
<td>3,725,203</td>
<td>108,024</td>
<td>575,061</td>
</tr>
</tbody>
</table>
Overall Performance

Evaluation results on switch logs dataset with supervised learning-based model

Switch log A -> Switch log B accuracy comparison

Switch log C -> Switch log B accuracy comparison
Overall Performance

Evaluation results on switch logs dataset with unsupervised learning-based model

Switch log A -> Switch log B accuracy comparison

Switch log C -> Switch log B accuracy comparison
## Word embedding method evaluation

### Comparison between LogTransfer with word2Vec & LogTransfer

<table>
<thead>
<tr>
<th>Method</th>
<th>F1-score</th>
<th>AUC score</th>
<th>#False positive</th>
<th>#False negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogTransfer w/ word2Vec</td>
<td>0.8368</td>
<td>0.8881</td>
<td>88</td>
<td>84</td>
</tr>
<tr>
<td>LogTransfer</td>
<td>0.8606</td>
<td>0.9243</td>
<td>80</td>
<td>59</td>
</tr>
</tbody>
</table>
Evaluation of the transfer learning method
Contributions

A new anomaly detection method
- We are the first to apply transfer learning for log anomaly detection
- We identify the challenges lying in that for a large software service.

A new log representation method
- We propose to use Glove to construct logs' representations to accurately measure the similarities of cross-system logs.

A new transfer learning approach
- We propose a novel transfer learning approach that shares fully connected networks between source and target systems, addressing the impact induced by the noises in log sequences.

An extensive evaluation
- We have conducted extensive evaluation experiments using real-world logs to demonstrate LogTransfer's performance.
Thanks

Q&A