DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning

Min Du, Feifei Li, Guineng Zheng, Vivek Srikumar
University of Utah
Background

081111 083419 35 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock: /user/root/rand7/_temporary/_task_20081101024_0014_m_001575_0/part-01575. blk_5214640714119373081
081111 083422 24621 INFO dfs.DataNode$DataXceiver: writeBlock blk_5214640714119373081 received exception java.io.IOException: Could not read from stream
081111 104136 35 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock: /user/root/randtxt9/_temporary/_task_20 0811101024_0016_m_001470_0/part-01470. blk_-3208483482800741142
081111 104233 26437 INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-3208483482800741142 terminating

......
Background

System Event Log

08111 083419 35 INFO fs.FileSystemBlock: Writing block /user/root/rand7/_temporary/blk_5214640714119373081
08111 083422 24621 INFO dfs.DataNode$DataXceiver: writeBlock blk_5214640714119373081 received exception java.io.IOException: Could not read from stream
08111 104136 35 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock: /user/root/randtxt9/_temporary/_task_20 0811101024_0016_m_001470_0/part-01470. blk_-3208483482800741142
08111 104233 26437 INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-3208483482800741142 terminating

......
Background

System Event Log

Available practically on every computer system!

081111 083419 35 INFO dfs.DataNode$DataXceiver: writeBlock blk_5214640714119373081 /user/root/rand7/_temporary/_task_20_0811101024_0016_m_001470_0/part-001470.blk_-3208483482800741142 src: /10.251.111.209:34510 dest: /10.251.121.224:50010
081111 104136 35 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock: /user/root/randtxt9/_temporary/_task_20 0811101024_0016_m_001470_0/part-001470.blk_-3208483482800741142
081111 104233 26437 INFO dfs.DataNode$PacketResponder: PacketResponder 1 for block blk_-3208483482800741142 terminating

......
Background

System Event Log

Available practically on every computer system!

Automatic Analysis?
Background
Started service A on port 80
Executor updated: app-1 is now LOADING
......
Started service A on port 80
Executor updated: app-1 is now LOADING

……

printf("Started service %s on port %d", x, y);
Background

System
Event
Log

Structured Data
Message type
Log key
printf(“Started service %s on port %d”, x, y);

Started service A on port 80
Executor updated: app-1 is now LOADING

Started service * on port *
Executor updated: * is now LOADING
Background

System Event Log

Structured Data
Message type
Log key
......
printf("Started service %s on port %d", x, y);

Anomaly Detection

LOG PARSING

LOG ANALYSIS
Background

System

Event

Log

Structured Data

Message type

Log key

......

printf("Started service %s on port %d", x, y);

Anomaly

Detection

LOG ANALYSIS

- Message count vector:
  Xu’SOSP09, Lou’ATC10, etc.
Structured Data
- Message type
- Log key
- ...

printf("Started service \%s on port \%d", x, y);

LOG ANALYSIS

- Message count vector:
  Xu’SOSP09, Lou’ATC10, etc.

- Problem: Offline batched processing
Background

Structured Data
Message type
Log key

```
printf("Started service \%s on port \%d", x, y);
```

Anomaly Detection

LOG ANALYSIS

- **Message count vector:**
  Xu’SOSP09, Lou’ATC10, etc.
  
  *Problem: Offline batched processing*

- **Build workflow model:**
  Lou’KDD10, Beschastnikh’ICSE14, Yu’ASPLOS16, etc.
Background

System
Event
Log

Structured Data
Message type
Log key
……
printf(“**Started service** %s on port %d”, x, y);

Anomaly
Detection

LOG ANALYSIS

- **Message count vector:**
  Xu’SOSP09, Lou’ATC10, etc.
  *Problem: Offline batched processing*

- **Build workflow model:**
  Lou’KDD10, Beschastnikh’ICSE14, Yu’ASPLOS16, etc.
  *Problem: Only for simple execution path anomalies*
Background

System Event Log

Structured Data
Message type
Log key
......
printf("Started service\n%s on port %d", x, y);

Anomaly Detection

LOG ANALYSIS

- Message count vector: Xu’SOSP09, Lou’ATC10, etc.
  Problem: Offline batched processing

- Build workflow model: Lou’KDD10, Beschastnikh’ICSE14, Yu’ASPLOS16, etc.
  Problem: Only for simple execution path anomalies

Common problem: Only Log keys (Message types) are considered.

LOG PARSING
### DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network …</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

SPELL
A streaming log parser published in ICDM'16
### DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

---

**SPELL**
A streaming log parser published in ICDM’16
**DeepLog**

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deletion of <strong>file1</strong> complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**SPELL**
A streaming log parser published in *ICDM’16*
**DeepLog**

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network …</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**log message** -> **log key** -> **parameters**

Deletion of file1 complete.

Deletion of * complete.

SPELL
A streaming log parser published in ICDM’16

[file1]
# DeepLog

- **Deletion of file1 complete.**  
  - Log message: Deletion of file1 complete.  
  - Log key: $k_1$  
  - Parameter value vector: $[t_1 - t_0, \text{file1}]$  

- **Took 0.61 seconds to deallocate network...**  
  - Log message: Took 0.61 seconds to deallocate network...  
  - Log key: $k_2$  
  - Parameter value vector: $[t_2 - t_1, 0.61]$  

- **VM Stopped (Lifecycle Event)**  
  - Log message: VM Stopped (Lifecycle Event)  
  - Log key: $k_3$  
  - Parameter value vector: $[t_3 - t_2]$  

- **VM Stopped (Lifecycle Event)**  
  - Log message: BM Stopped (Lifecycle Event)  
  - Log key: $k_3$  
  - Parameter value vector: $[t_3 - t_2]$  

---

The Deletion of file2 complete message is also included in the log, but the details are not shown in this table.

---

**SPELL**  
A streaming log parser published in ICDM'16  

**Deletion of * complete.**  
- Parameter value vector: [file1]
DeeplLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Deletion of file1 complete.  
Deletion of file2 complete.  
Deletion of * complete.  
Deletion of * complete.  
Deletion of * complete.
## DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Log Key Anomaly Detection model**

**Workflows**
DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Diagram

- **Log Key Anomaly Detection model**
- **Workflows**
- **Parameter Values Anomaly Detection model**

### Anomaly Detection
## DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1}]$</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>$[t_2 - t_1, 0.61]$</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>$[t_3 - t_2]$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Diagram:

- **Log Key Anomaly Detection model**
- **Workflows**
- **Parameter Values Anomaly Detection model**
- **Anomaly Detection**
- **Diagnosis**
## DeepLog

<table>
<thead>
<tr>
<th>log message (log key underlined)</th>
<th>log key</th>
<th>parameter value vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$ Deletion of file1 complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1]</td>
</tr>
<tr>
<td>$t_2$ Took 0.61 seconds to deallocate network ...</td>
<td>$k_2$</td>
<td>[$t_2 - t_1$, 0.61]</td>
</tr>
<tr>
<td>$t_3$ VM Stopped (Lifecycle Event)</td>
<td>$k_3$</td>
<td>[$t_3 - t_2$]</td>
</tr>
</tbody>
</table>

---

### Anomaly Detection

**Log Key Anomaly Detection model**

**Workflows**

**Parameter Values Anomaly Detection model**

---

### Diagnosis

Anomaly Detection
DeepLog Architecture

**Training Stage**
- Each log entry = log key + parameter value vectors
- Training model
- Construct workflow
- \( t_1 : \text{log entry}_1 \)
- \( t_2 : \text{log entry}_2 \)
- \( t_3 : \text{log entry}_3 \)
- \( t_4 : \text{log entry}_4 \)
- \( t_5 : \text{log entry}_5 \)
- \( t_6 : \text{log entry}_6 \)

**Detection Stage**
- Log key anomaly detection model
- Workflows
- \( k_1 \)
- \( k_2 \)
- \( k_i \)
- Parameter value anomaly detection model for each log key
- A new log entry
- \( \text{Log Parser} \)

MODELS

29
DeepLog Architecture

Training Stage

Normal execution log file:
- $t_1$: log entry 1
- $t_2$: log entry 2
- $t_3$: log entry 3
- $t_4$: log entry 4
- $t_5$: log entry 5
- $t_6$: log entry 6

Log Parser

Detect model Construct workflow

Detection Stage

A new log entry

Log Key Anomaly Detection model

Parameter Value Anomaly Detection model for each log key

Detection Stage

Diagnose false positive

Update model

Anomaly?
DeepLog Architecture

Each log entry = log key + parameter value vector

Training Stage

Log Parser

- $t_1$: log entry1
- $t_2$: log entry2
- $t_3$: log entry3
- $t_4$: log entry4
- $t_5$: log entry5
- $t_6$: log entry6

- $k_1$ + $[t_{d1}, v_{11}, \ldots]$ (parameter value vector)
- $k_2$ + $[t_{d2}, v_{21}, \ldots]$ (parameter value vector)
- $k_i$ + $[t_{di}, v_{i1}, \ldots]$ (parameter value vector)

Train model

Construct workflow

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

Update model if false positive

Diagnosis

Ano

Ano
DeepLog Architecture

Log Parser

Train model

Construct workflow

Workflows

Log Key
Anomaly Detection
model

Train models

Parameter Value
Anomaly Detection
model

for each log key

Normal execution log file

$\bar{t}_1$ : log entry 1
$\bar{t}_2$ : log entry 2
$\bar{t}_3$ : log entry 3
$\bar{t}_4$ : log entry 4
$\bar{t}_5$ : log entry 5
$\bar{t}_6$ : log entry 6

......

$\bar{k}_1$ + $[\bar{t}_{d1}, \bar{v}_{11}, \ldots]$  
$\bar{k}_2$ + $[\bar{t}_{d2}, \bar{v}_{21}, \ldots]$  
$\bar{k}_i$ + $[\bar{t}_{di}, \bar{v}_{i1}, \ldots]$  
$\bar{k}_1$ + $[\bar{t'}_{d1}, \bar{v}_{11}, \ldots]$  
$\bar{k}_2$ + $[\bar{t'}_{d2}, \bar{v}_{21}, \ldots]$  
$\bar{k}_i$ + $[\bar{t'}_{di}, \bar{v}_{i1}, \ldots]$  

......

each log entry = log key + parameter value vector

Diagnosis

Update model if false positive

Ano

Ano

Ano

32
DeepLog Architecture

Each log entry = log key + parameter value vector

Training Stage

- Normal execution log file
  - \( t_1 \): log entry1
  - \( t_2 \): log entry2
  - \( t_3 \): log entry3
  - \( t_4 \): log entry4
  - \( t_5 \): log entry5
  - \( t_6 \): log entry6
  - ...

- Log Parser

- Train model
  - Construct workflow

- Log Key Anomaly Detection model

- Workflows

- Train models
  - Parameter Value Anomaly Detection model for each log key

- Diagnosis
  - Update model if false positive

- Ano
DeepLog Architecture

Each log entry = log key + parameter value vector

Log Key
Anomaly Detection
model

Train model
Construct workflow

Workflows

Parameter Value
Anomaly Detection
model for each log key

Training Stage

normal execution log file

$t_1$ : log entry 1
$t_2$ : log entry 2
$t_3$ : log entry 3
$t_4$ : log entry 4
$t_5$ : log entry 5
$t_6$ : log entry 6
......
DeepLog Architecture

Each log entry = log key + parameter value vector

Training Stage

Normal execution log file
- $t_1$: log entry 1
- $t_2$: log entry 2
- $t_3$: log entry 3
- $t_4$: log entry 4
- $t_5$: log entry 5
- $t_6$: log entry 6

Log Parser

Train models

Construct workflow

Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key
DeepLog Architecture

MODELS

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

Training Stage

Detection Stage

A new log entry

Parse

Log

Anomaly?

Yes

Train model

Construct workflow

normal execution log file

$t_1$, $t_2$, $t_3$, $t_4$, $t_5$, $t_6$, ...

$k_1$, $k_2$, $k_i$, ...

$k_1$: log entry

$k_2$: log entry

$k_i$: log entry

......

$k_i$: log entry

......

$[r^1_{d1}, v^1_{i1}, \ldots]$

$[r^2_{d2}, v^2_{i2}, \ldots]$

$[r^i_{di}, v^i_{di}, \ldots]$

Train models

Diagnoses

Update model if false positive

Anomaly?

No

$\text{Train} \rightarrow \text{Identify} \rightarrow \text{Diagnoses} \rightarrow \text{Update} \rightarrow \text{Detect} \rightarrow \text{Train}$

each log entry = log key + parameter value vector
DeepLog Architecture

Training Stage

MODELS

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

Diagnosis

Update model if false positive

A new log entry

log key $k_i$

+ parameter value vector $[t_{d1}, v_{i1}, \ldots]$

Anomaly?

Yes

No, check vector

Anomaly?

Yes

No

Parser

normal execution log file

$\begin{align*}
    t_1 &: \text{ log entry}_1 \\
    t_2 &: \text{ log entry}_2 \\
    t_3 &: \text{ log entry}_3 \\
    t_4 &: \text{ log entry}_4 \\
    t_5 &: \text{ log entry}_5 \\
    t_6 &: \text{ log entry}_6
\end{align*}$

Training Stage

Detection Stage

each log entry = log key + parameter value vector

Train model

Construct workflow

Train models

......

37
DeepLog Architecture

A new log entry

Log Parser

log key $k_i$

+ parameter value vector $[t_{di}, v_{i1}, \ldots]\n
Diagnosis

Update model if false positive

Anomaly?

Yes

No, check vector

Anomaly?

Yes

No
DeepLog Architecture

A new log entry

- log key $k_i$
  
  + parameter value vector $[t_{di}, v_{i1}, \ldots]$
DeepLog Architecture

Detection Stage

- A new log entry
  - Log Parser
  - log key $k_i$
  - + parameter value vector $[t_{di}, v_{i1}, \ldots]$

- Diagnosis
  - Update model if false positive
    - Yes
    - No, check vector

- Anomaly detection model for each log key
  - $k_1$
  - $k_2$
  - $k_i$

- Workflows
  - value vector
    - model
    - uct workflow
      - \ldots

- Train models
      - \ldots
DeepLog Architecture

A new log entry

Log Parser

log key $k_i$

+ parameter value vector $[t_{di}, v_{i1}, \ldots]$

Anomaly?

Yes

No, check vector

Diagnosis

Update model if false positive

Yes

Anomaly?

No

Train models

$k_1$

$k_2$

$k_i$

......

Parameter Value Anomaly Detection model for each log key

Workflows

Log Key Anomaly Detection model

value vector model

uct workflow

Detection Stage
DeepLog Architecture

A new log entry

Log Parser

log key $k_i$

+ parameter value vector 

[$t_{di}, v_{i1}, \ldots$]

Anomaly?

Yes

No, check vector

Diagnosis

Update model if false positive

Anomaly?

Yes

No

Train models

value vector

uct workflow

model

Log Key

Anomaly Detection model

Workflows

$k_1$

$k_2$

$k_i$

Parameter Value Anomaly Detection model for each log key

Detection Stage
DeepLog Architecture

- Log Key
  - Anomaly Detection model

- Workflows

- A new log entry
  - Log
  - Parser
  - log key $k_i$
  - +
  - parameter value vector $[t_{di}, v_{i1}, \ldots]$

- Diagnosis
  - Update model if false positive

- Parameter Value
  - Anomaly Detection model
  - for each log key

- Train models

- No, check vector

- Yes

- No

- Smiley face

- value vector

- model

- act workflow

- , , , ,

- , , , ,

- , , , ,

- , , , ,
DeepLog Architecture

A new log entry

Log Parser

log key $k_i$

+ parameter value vector

$[t_{di}, v_{i1}, \ldots]$
DeepLog Architecture

Log Key
Anomaly Detection model

Workflows

Parameter Value
Anomaly Detection model
for each log key

A new log entry

Log
Parser

log key $k_i$

+ parameter
value vector

$t_{di}, v_{i1}, \ldots$

Diagnosis

Update model if
false positive

Anomaly?

Yes

No, check
vector

Anomaly?

Yes

No

Train models
DeepLog Architecture

Training Stage:
- Normal execution log file
  - \( t_1 \): log entry 1
  - \( t_2 \): log entry 2
  - \( t_3 \): log entry 3
  - \( t_4 \): log entry 4
  - \( t_5 \): log entry 5
  - \( t_6 \): log entry 6

Log Parser:
- \( k_1 \) + \([t_{d1}, v_{i1}, \ldots]\)
- \( k_2 \) + \([t_{d2}, v_{i2}, \ldots]\)
- \( k_i \) + \([t_{di}, v_{i1}, \ldots]\)

Train model:
- Construct workflow

Detection Stage:
- A new log entry
- Log key \( k_i \) + parameter value vector \([t_{di}, v_{i1}, \ldots]\)
- Diagnosis
  - Update model if false positive
- Anomaly?
  - Yes
  - No, check vector

Parameter Value Anomaly Detection model for each log key

MODELS:
- Log Key Anomaly Detection model
- Workflows

Log Key Anomaly Detection model

Example log key sequence:
25 18 54 57 18 56 ... 25 18 54 57 56 18 ...

➢ a rigorous set of logic and control flows
➢ a (more structured) natural language
Log Key Anomaly Detection model

Example log key sequence:
25 18 54 57 18 56 … 25 18 54 57 56 18 …

- a rigorous set of logic and control flows
- a (more structured) natural language

natural language modeling

multi-class classifier: history sequence => next key to appear
Example log key sequence:
25 18 54 57 18 56 … 25 18 54 57 56 18 …

➢ a rigorous set of logic and control flows
➢ a (more structured) natural language

natural language modeling

multi-class classifier: history sequence => next key to appear

A log key is detected to be abnormal if it does not follow the prediction.
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: $h$ recent log keys up to $m_{t-1}$

Output: conditional probability of next log key given the input recent sequence

$w = \{m_{t-h}, \ldots, m_{t-2}, m_{t-1}\}$

$\Pr(m_t = k_i | w)$ for each log key $k_i$
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: $h$ recent log keys up to $m_{t-1}$

$w = \{m_{t-h}, \ldots, m_{t-2}, m_{t-1}\}$

Output: conditional probability of next log key given the input recent sequence

$\Pr(m_t = k_i | w) \quad k_i \in K(i = 1, \ldots, n)$

Training:

log key sequence:

$h=3 \quad 25\ 18\ 54\ 57\ 18\ 56 \ldots\ 25\ 18\ 54\ 57\ 56\ 18 \ldots$
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: $h$ recent log keys up to $m_{t-1}$

Output: conditional probability of next log key given the input recent sequence

Training:

log key sequence:

$h=3$ 25 18 54 57 18 56 ... 25 18 54 57 56 18 ...
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: $h$ recent log keys up to $m_{t-1}$

Output: conditional probability of next log key given the input recent sequence

Training:

log key sequence:

$h=3$ 25 18 54 57 18 56 ... 25 18 54 57 56 18 ...
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: \( h \) recent log keys up to \( m_{t-1} \)

\[ w = \{m_{t-h}, \ldots, m_{t-2}, m_{t-1}\} \]

Output: conditional probability of next log key given the input recent sequence

Pr(\( m_t = k_i | w \)) \( k_i \in K(i = 1, \ldots, n) \)

Training:

log key sequence:

\( h=3 \) 25 18 54 57 18 56 ... 25 18 54 57 56 18 ...
Log Key Anomaly Detection model

Use long short-term memory (LSTM) architecture

Input: $h$ recent log keys up to $m_{t-1}$

Output: conditional probability of next log key given the input recent sequence

$w = \{m_{t-h}, \ldots, m_{t-2}, m_{t-1}\}$

Detection:

In detection stage, DeepLog checks if the actual next log key is among its top $g$ probable predictions.
Log Key Anomaly Detection model

Output of last state is forwarded as current input state

$k_1$
$k_2$
$k_i$

Parameter Value
Anomaly Detection model
for each log key

Workflows
Log Key Anomaly Detection model

Parameter Value
Anomaly Detection model for each log key

$k_1$
$k_2$
$k_i$

Workflows

Output of last state is forwarded as current input state

$\text{LSTM block}$

$m_{t-i}$

$\text{Roll out}$

$LSTM block$

$LSTM block$

$LSTM block$

$H_{t-h}$

$C_{t-h}$

$m_{t-h}$

$H_{t-2}$

$C_{t-2}$

$m_{t-2}$

$H_{t-1}$

$m_{t-1}$
Log Key Anomaly Detection model

Log Key Anomaly Detection model

Parameter Value Anomaly Detection model for each log key

Workflows

$k_1$

$k_2$

$k_i$

......

Output of last state is forwarded as current input state

Roll out

Stack up

DeepLog

Input

Output
Workflow Construction

Input: log key sequence
25 18 54 57 18 56 … 25 18 54 57 56 18 …

Output:
Workflow Construction

Method 1: Using Log Key Anomaly Detection model
--- LSTM prediction probabilities
Workflow Construction

Method 1: Using Log Key Anomaly Detection model

--- LSTM prediction probabilities

An example of concurrency detection:
Workflow Construction

Method 1: Using Log Key Anomaly Detection model
--- LSTM prediction probabilities

An example of concurrency detection:

1. [25, 18, 54] -> [57: 1.00]
Workflow Construction

Method 1: Using Log Key Anomaly Detection model
--- LSTM prediction probabilities

An example of concurrency detection:

1. [25, 18, 54] -> {57: 1.00}

2. [18, 54, 57] -> {18: 0.8, 56: 0.2}
Workflow Construction

Method 1: Using Log Key Anomaly Detection model
--- LSTM prediction probabilities

An example of concurrency detection:

1. [25, 18, 54] -> {57: 1.00}
2. [18, 54, 57] -> {18: 0.8, 56: 0.2}
3. [54, 57, 18] -> {56: 1.00}
   [54, 57, 56] -> {18: 1.00}
Method 1: Using Log Key Anomaly Detection model
--- LSTM prediction probabilities

An example of concurrency detection:

1. [25, 18, 54] -> {57: 1.00}
2. [18, 54, 57] -> {18: 0.8, 56: 0.2}
3. [54, 57, 18] -> {56: 1.00}
   [54, 57, 56] -> {18: 1.00}

Workflow Construction

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

$k_1$

$k_2$

$k_i$
Workflow Construction

Method 2: A density-based clustering approach
Workflow Construction

Method 2: A density-based clustering approach

Co-occurrence matrix of log keys \((k_i, k_j)\) within distance \(d\)

\[
\begin{array}{cccc}
  & k_1 & \ldots & k_j & \ldots & k_n \\
k_1 & p_d(1, 1) & & p_d(1, j) & & \\
\vdots & & & & & \\
k_i & p_d(i, 1) & & p_d(i, j) = \frac{f_d(k_i, k_j)}{d \cdot f(k_i)} & & \\
\vdots & & & & & \\
k_n & p_d(n, 1) & & p_d(n, j) & & \\
\end{array}
\]

- \(f_d(k_i, k_j)\): the frequency of \((k_i, k_j)\) appearing together within distance \(d\)
- \(f(k_i)\): the frequency of \(k_i\) in the input sequence
- \(p_d(i, j)\): the probability of \((k_i, k_j)\) appearing together within distance \(d\)
Parameter Value Anomaly Detection model

Example:

Log messages of a particular log key:

\( t_2: \text{Took 0.61 seconds to deallocate network} \ldots \)

\( t'_2: \text{Took 1.1 seconds to deallocate network} \ldots \)

\ldots
Parameter Value Anomaly Detection model

Example:

Log messages of a particular log key:
\[ t_2: \text{Took 0.61 seconds to deallocate network} \quad ... \]
\[ t'_2: \text{Took 1.1 seconds to deallocate network} \quad ... \]

Parameter value vectors overtime:
\[ [t_2 - t_1, 0.61], [t'_2 - t'_1, 1.1], ... \]
Parameter Value Anomaly Detection model

Example:

Log messages of a particular log key:
- \( t_2: \text{Took 0.61 seconds to deallocate network} \ldots \)
- \( t’_2: \text{Took 1.1 seconds to deallocate network} \ldots \)

Parameter value vectors overtime:
- \([t_2 - t_1, 0.61], [t’_2 - t’_1, 1.1], \ldots\)

Multi-variate time series data anomaly detection problem!
Multi-variate time series data anomaly detection problem

✓ Leverage LSTM-based approach;
✓ A parameter value vector is given as input at each time step;
✓ An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

✓ Leverage LSTM-based approach;
✓ A parameter value vector is given as input at each time step;
✓ An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

- Leverage LSTM-based approach;
- A parameter value vector is given as input at each time step;
- An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

- Leverage LSTM-based approach;
- A parameter value vector is given as input at each time step;
- An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

- Leverage LSTM-based approach;
- A parameter value vector is given as input at each time step;
- An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.

![Diagram showing time series data with anomaly detection logic](image)
Parameter Value Anomaly Detection model

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

Multivariate time series data anomaly detection problem

- Leverage LSTM-based approach;
- A parameter value vector is given as input at each time step;
- An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

- Leverage LSTM-based approach;
- A parameter value vector is given as input at each time step;
- An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

✓ Leverage LSTM-based approach;
✓ A parameter value vector is given as input at each time step;
✓ An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
Parameter Value Anomaly Detection model

Multi-variate time series data anomaly detection problem

✓ Leverage LSTM-based approach;
✓ A parameter value vector is given as input at each time step;
✓ An anomaly is detected if the mean-square-error (MSE) between prediction and actual data is too big.
LSTM model online update

Q: How to handle false positive?
LSTM model online update

Q: How to handle false positive?

Log sequence: history
LSTM model online update

Q: How to handle false positive?

Log sequence:

history

model
Q: How to handle false positive?

Log sequence:

- History
- Model
- Prediction
LSTM model online update

Q: How to handle false positive?
LSTM model online update

Q: How to handle false positive?
LSTM model online update

Q: How to handle false positive?
Q: How to handle false positive?

Update model using this case: "history -> current"
Evaluation results on HDFS log data \[^1\].

(over a million log entries with labeled anomalies)

\[^1\] PCA (SOSP’09), IM (UsenixATC’10), N-gram (baseline language model)
Evaluation – parameter value anomaly detection

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)

MSE: mean square error
Evaluation – parameter value anomaly detection

MSE: mean square error

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)

generated on CloudLab;
VM creation/deletion operations;
injected performance anomalies.
Evaluation – parameter value anomaly detection

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)

MSE: mean square error
Evaluation – parameter value anomaly detection

MSE: mean square error

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)
Evaluation – parameter value anomaly detection

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)

MSE: mean square error

False Positive

ANOMALY

thresholds
Evaluation – LSTM model online update

Evaluation on Blue Gene/L log, with and without online model update.
Evaluation – LSTM model online update

Evaluation on Blue Gene/L log, with and without online model update.

HPC log with labeled anomalies; Available at https://www.usenix.org/cfdr-data
Evaluation – case study: network security log

Dataset: IEEE VAST Challenge 2011

(Mini Challenge 2 – Computer Networking Operations)

The dataset contains firewall log, IDS log, etc.
## Dataset: IEEE VAST Challenge 2011

*(Mini Challenge 2 – Computer Networking Operations)*

The dataset contains firewall log, IDS log, etc.

<table>
<thead>
<tr>
<th>suspicious activity</th>
<th>detected?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1: Denial of Service attack</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 1: port scan</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: port scan 1</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: port scan 2</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: socially engineered attack</td>
<td>Yes, log key anomaly in firewall log</td>
</tr>
<tr>
<td>Day 3: undocumented IP address</td>
<td>No</td>
</tr>
</tbody>
</table>

**Detection results.**
Evaluation – case study: network security log

Dataset: IEEE VAST Challenge 2011

(Mini Challenge 2 – Computer Networking Operations)

The dataset contains firewall log, IDS log, etc.

<table>
<thead>
<tr>
<th>suspicious activity</th>
<th>detected?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1: Denial of Service attack</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 1: port scan</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: port scan 1</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: port scan 2</td>
<td>Yes, log key anomaly in IDS log</td>
</tr>
<tr>
<td>Day 2: socially engineered attack</td>
<td>Yes, log key anomaly in firewall log</td>
</tr>
<tr>
<td>Day 3: undocumented IP address</td>
<td>No</td>
</tr>
</tbody>
</table>

Detection results.

Could be fixed with prior knowledge of “documented IP”
Evaluation – workflow construction

**Constructed workflow of VM Creation.**

(previously generated OpenStack cloud log)
How does it help to diagnose anomalies?

44: instance: * Attempting claim: memory * disk * vcpus * CPU
51: instance: * Claim successful
23: instance: * GET * HTTP/1.1" status: * len: * time: *
52: instance: * Creating image
53: instance: * VM Started (Lifecycle Event)
32: instance: * VM Paused (Lifecycle Event)
18: instance: * VM Resumed (Lifecycle Event)

56: instance: * Took * seconds to build instance

**Constructed workflow of VM Creation.**
(previously generated OpenStack cloud log)
Evaluation – workflow construction

How does it help to diagnose anomalies?

Parameter value anomaly

44: instance: *Attempting claim: memory * disk * vcpus * CPU
51: instance: *Claim successful
23: instance: *GET * HTTPV1.1" status: *len: *time: *
52: instance: *Creating image
53: instance: *VM Started (Lifecycle Event)
32: instance: *VM Paused (Lifecycle Event)
18: instance: *VM Resumed (Lifecycle Event)

56: instance: *Took *seconds to build instance

Constructed workflow of VM Creation.
(previously generated OpenStack cloud log)
Evaluation – workflow construction

How does it help to diagnose anomalies?

Parameter value anomaly

Time difference (performance) anomaly

44: instance: * Attempting claim: memory * disk * vcpus * CPU
51: instance: * Claim successful
23: instance: * GET * HTTPv1.1" status: * len: * time: *
52: instance: * Creating image
53: instance: * VM Started (Lifecycle Event)
32: instance: * VM Paused (Lifecycle Event)
18: instance: * VM Resumed (Lifecycle Event)

56: instance: * Took * seconds to build instance

Constructed workflow of VM Creation.
(previously generated OpenStack cloud log)
Evaluation – workflow construction

How does it help to diagnose anomalies?

Identified anomaly:
Instance took too long to build because of the transition from 52 -> 53

Constructed workflow of VM Creation.
(previously generated OpenStack cloud log)
Evaluation – workflow construction

How does it help to diagnose anomalies?

44: instance: * Attempting claim: memory * disk * vcpus * CPU
51: instance: * Claim successful
23: instance: * GET * HTTP/1.1" status: * len: * time: *
52: instance: * Creating image
53: instance: * VM Started (Lifecycle Event)
32: instance: * VM Paused (Lifecycle Event)
18: instance: * VM Resumed (Lifecycle Event)

56: instance: * Took * seconds to build instance

Identified anomaly:
Instance took too long to build because of the transition from 52 -> 53

Injected anomaly:
During VM creation, network speed from controller to compute node is throttled.

Constructed workflow of VM Creation.
(previously generated OpenStack cloud log)
DeepLog

➢ A realtime system log anomaly detection framework.
➢ LSTM is used to model system execution paths and log parameter values.
➢ Workflow models are built to help anomaly diagnosis.
➢ It supports online model update.

Thank you!

Min Du  Feifei Li
mind@cs.utah.edu  lifeifei@cs.utah.edu