DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning
(to appear in CCS’17)

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

15/07/31 12:20:17 INFO SparkContext: Running Spark version 1.3.0
15/07/31 12:20:18 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using built-in-java classes where applicable
15/07/31 12:20:18 INFO SecurityManager: Changing view acts to: zhoulang
15/07/31 12:20:18 INFO SecurityManager: Changing modify acts to: zhoulang
15/07/31 12:20:18 INFO SecurityManager: SecurityManager: authentication disabled; ui acts disabled; users with view permissions: Set(zhoulang); users with modify permissions: Set(zhoulang)
15/07/31 12:20:18 INFO Slf4jLogger: Slf4jLogger started
15/07/31 12:20:18 INFO Remoting: Starting remoting
15/07/31 12:20:18 INFO Remoting: Remoting started; listening on addresses: [akka.tcp://sparkDriver@head:60626]
15/07/31 12:20:18 INFO Utils: Successfully started service 'sparkDriver' on port 60626.
15/07/31 12:20:18 INFO SparkEnv: Registering MapOutputTracker
15/07/31 12:20:18 INFO SparkEnv: Registering BlockManagerMaster
15/07/31 12:20:18 INFO DiskBlockManager: Created local directory at /tmp/spark-3799bc3c-5275-499c-8b89-fe93e68031e3/blockmgr-7fe0b97-c8b3-4fa9-b6e3-2af1620db1e3
15/07/31 12:20:18 INFO MemoryStore: MemoryStore started with capacity 10.4 GB
15/07/31 12:20:19 INFO HttpFileServer: HTTP file server directory is /tmp/spark-c01a992b-d9d3-4751-8f2e-05ca6a4cb329/httpd-b9f5f8c6-0f7c-434c-aed4-20f27b9b3731
15/07/31 12:20:19 INFO HttpServer: Starting HTTP Server
15/07/31 12:20:19 INFO Server: jetty-8.y.z-SNAPSHOT
15/07/31 12:20:19 INFO AbstractConnector: Started SocketConnector@0.0.0.0:43664
15/07/31 12:20:19 INFO HttpUtils: Successfully started service 'HTTP file server' on port 43664.
15/07/31 12:20:19 INFO SparkEnv: Registering OutputCommitCoordinator
15/07/31 12:20:19 INFO Server: jetty-8.y.z-SNAPSHOT
15/07/31 12:20:19 INFO AbstractConnector: Started SelectChannelConnector@0.0.0.0:4040
15/07/31 12:20:19 INFO HttpUtils: Successfully started service 'SparkUI' on port 4040.
15/07/31 12:20:19 INFO AppClient$ClientActor: Connecting to master akka.tcp://sparkMaster@head:7077/user/Master...
15/07/31 12:20:19 INFO SparkDeploySchedulerBackend: Connected to Spark cluster with app ID
System Event Log

Exist practically on every computer system!

Automatic Analysis?
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

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

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

Started service * on port *
Executor updated: * is now LOADING
......
## 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 <strong>file1</strong> complete</td>
<td>$k_1$</td>
<td>$[t_1 - t_0, \text{file1Id}]$</td>
</tr>
<tr>
<td>$t_2$ Took <strong>0.61</strong> 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><em>t</em>&lt;sub&gt;1&lt;/sub&gt; Deletion of <strong>file1</strong> complete</td>
<td><em>k</em>&lt;sub&gt;1&lt;/sub&gt;</td>
<td>[<em>t</em>&lt;sub&gt;1&lt;/sub&gt; – <em>t</em>&lt;sub&gt;0&lt;/sub&gt;, file1Id]</td>
</tr>
<tr>
<td><em>t</em>&lt;sub&gt;2&lt;/sub&gt; Took <strong>0.61</strong> seconds to deallocate network ...</td>
<td><em>k</em>&lt;sub&gt;2&lt;/sub&gt;</td>
<td>[<em>t</em>&lt;sub&gt;2&lt;/sub&gt; – <em>t</em>&lt;sub&gt;1&lt;/sub&gt;, 0.61]</td>
</tr>
<tr>
<td><em>t</em>&lt;sub&gt;3&lt;/sub&gt; VM Stopped (Lifecycle Event)</td>
<td><em>k</em>&lt;sub&gt;3&lt;/sub&gt;</td>
<td>[<em>t</em>&lt;sub&gt;3&lt;/sub&gt; – <em>t</em>&lt;sub&gt;2&lt;/sub&gt;]</td>
</tr>
</tbody>
</table>

log message                          log key                          parameters

*Deletion of file1 complete.*  ➔  SPELL  ➔  *Deletion of file1 complete.*  ➔  []

*Deletion of file2 complete.*  ➔  A streaming log parser published in ICDM’16  ➔  *Deletion of * complete.*  ➔  [file2]
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 <strong>file1</strong> complete</td>
<td>$k_1$</td>
<td>[$t_1 - t_0$, file1Id]</td>
</tr>
<tr>
<td>$t_2$ Took <strong>0.61</strong> 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 Architecture

**Training Stage**

**Detection Stage**

MODELS

- Log Key Anomaly Detection model
- Parameter Value Anomaly Detection model for each log key
DeepLog Architecture

 MODELS

Detection Stage

A new log entry

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

train models

Train model

Construct workflow

Log Key

each log entry = log key + parameter value vector

normal execution log file

Training Stage

log entry 1

log entry 2

log entry 3

log entry 4

log entry 5

log entry 6

...
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

Detection Stage

- each log entry = log key + parameter value vector

MODELS

- Train model
- Construct workflow

Log Key Anomaly Detection model

- Workflows

- A new log entry
- Log Key
- Anomaly?

Parameter Value Anomaly Detection model for each log key

Anomaly?

- Yes
  - Diagnosis data model
  - False positive
- No
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

- Train model
- Construct workflow

Detection Stage

- MODELS
  - Log Key Anomaly Detection model
  - Workflows

- Detection Stage
  - Parameter Value Anomaly Detection model for each log key
  - A new log entry
  - Log key $k_i$
DeepLog Architecture

**Training Stage**

- Log Parser
- Normal execution log file
- Each log entry = log key + parameter value vector
- Train models
- Construct workflow
- Log Key Anomaly Detection model

**Detection Stage**

- A new log entry
- Detection Stage
- Anomaly? (Yes/No)
- Parameter Value Anomaly Detection model for each log key
- Update model
- False positive
- Yes/No
- No

**MODELS**

- Log Key Anomaly Detection model
- Workflows
DeepLog Architecture

Training Stage

MODELS

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

**Training Stage**

- Each log entry = log key + parameter value vector
- Normal execution log file
- Train model
- Construct workflow
- $t_1$, $t_2$, $t_3$, $t_4$, $t_5$, $t_6$, etc.

**MODELS**

- Log Key Anomaly Detection model
- Workflows
  - $k_1$, $k_2$, $k_3$, $k_i$, etc.
  - Parameter Value Anomaly Detection model for each log key

**Detection Stage**

- A new log entry
- Log Parser
- Log key $k_i$ + parameter value vector $[t_{d1}, v_{i1}, \ldots]$
- Diagnosis
- Update model if false positive
- Anomaly?
DeepLog Architecture

Training Stage

MODELS

Log Key Anomaly Detection model

Workflows

Parameter Value Anomaly Detection model for each log key

A new log entry

Log Parser

Anomaly?

Diagnosis Update model if false positive

No, check vector

Anomaly?

{\log key k_i} + parameter value vector \{t_{d1}, v_{i1}, \ldots\}
DeepLog Architecture

Training Stage

MODELS

A new log entry

Log Parser

parameter value vector \([t_{di}, v_{i1}, \ldots]\)

log key \(k_i\)

Anomaly?

Yes

No

Update model if false positive

Diagnosis

No, check vector

Anomaly?

Yes

No

Parameter Value Anomaly Detection model for each log key

Training Stage

normal execution log file

\(t_1: \text{log entry}_1\)

\(t_2: \text{log entry}_2\)

\(t_3: \text{log entry}_3\)

\(t_4: \text{log entry}_4\)

\(t_5: \text{log entry}_5\)

\(t_6: \text{log entry}_6\)

----------

Train model

Construct workflow

each log entry = log key + parameter value vector

Train models
DeepLog Architecture

**Training Stage**

- Training log file
  - $t_1$: log entry
  - $t_2$: log entry
  - $t_3$: log entry
  - $t_4$: log entry
  - $t_5$: log entry
  - $t_6$: log entry

- Normal execution workflow

- Each log entry = log key + parameter value vector

**ModeLS**

- Log Key Anomaly Detection model
- Workflows
- Parameter Value Anomaly Detection model for each log key

**Detection Stage**

- A new log entry
  - Log Parser
  - Log key $k_i$
  - + parameter value vector $[t_{d1}, v_{i1}, \ldots]$
DeepLog Architecture

**Training Stage**

- Each log entry = log key + parameter value vector
- Normal execution log file
- Train model
- Construct workflow
- Train models
- $t_1, t_2, \ldots, t_n$ : log entry
- $k_1, k_2, \ldots, k_n$ : log key
- $[v_{i1}, v_{i2}, \ldots]$ : parameter value vector

**Detection Stage**

- A new log entry
- Log Parser
- Log key $k_i$
- + Parameter value vector $[t_{di}, v_{i1}, \ldots]$
- Anomaly Detection model
- Workflows
- Diagnosis
- Update model if false positive
- Anomaly?
- Yes
- No, check vector
- Anomaly?
- Yes
- No

**MODELS**

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

In detection stage, DeepLog checks if the actual next log key is among its top $g$ probable predictions.
Workflow Construction

Method 1: Using LSTM prediction probabilities

Method 2: Using co-occurrence matrix
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:
\([t_2 - t_1, 0.61], [t'_2 - t'_1, 1.1], \ldots\)
Parameter Value Anomaly Detection model

Example:

Log messages of a particular log key:
t₂: Took 0.61 seconds to deallocate network ...
t₂’: Took 1.1 seconds to deallocate network ...
....

Parameter value vectors overtime:
[t₂ - t₁, 0.61], [t₂’ - t₁', 1.1], ....

Multi-variate time series data anomaly detection problem!
--- Leverage LSTM to check reconstruction error.
Evaluation results on HDFS log data.
(over a million log entries with labeled anomalies)

PCA (SOSP’09), IM (UsenixATC’10), N-gram (baseline language model)
Evaluation results on Blue Gene/L log, with and without online model update.
Evaluation – parameter value anomaly detection

Evaluation results on OpenStack cloud log with different confidence intervals (CIs)
Diagnosis using constructed workflow.

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

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.

Min Du
mind@cs.utah.edu