LogOptPlus: Learning to Optimize Logging in Catch and If Programming Constructs

Sangeeta Lal¹, Neetu Sardana¹, and Ashish Sureka²

¹JIIT, Noida, India
²ABB, Bangalore, India

{sangeeta, neetu.sardana}@jiit.ac.in,
ashish.sureka@in.abb.com

June 11, 2016
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
**INTRODUCTION**

- Software Logging is an important software development practice which is used to trace important execution points in the source code.

- Example:

```java
1. if (ssoId != null)
2. {
3.     if (log.isDebugEnabled())
4.         {
5.             log.debug("SSO Id" + ssoId + "set; attempting")
6.         }
```
WHY SOFTWARE LOGGING IS IMPORTANT?

- Logging statements are helpful in debugging
  - Yuan et al. [1] reported that bug reports having logging statements are fixed faster (1.8 times)
  - Survey results show that 96% of Microsoft developers consider logging statements important for debugging[2]

- Logging is the only information available for debugging many times
  - Privacy concerns related to user input
  - Difficulty of creating the same execution environment (same hardware, software version)

- Better than commonly used “printf” statements for debugging
  - Provide customizable support
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
RESEARCH MOTIVATION

- **Cost and benefit** tradeoff in inserting logging statements in the source code
  - Sparse logging can lessen the benefits of logging by leaving important information
  - Excessive logging can cause performance and cost overhead

- Developers often **face difficulty** in inserting logging statement in the source code
  - Lack of training required to make strategic logging decision, logging mostly done based of knowledge and expertise of the developers
  - Several times developer modify logging statement as afterthought

- **IF** blocks are one of the **most frequent logged** code snippets
  - Fu et al.[2] shows catch blocks frequently logged
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
**State of the art of automated logging/logging prediction**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Author &amp; Year</th>
<th>Aim</th>
<th>Dataset</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Lal et al.[2016]</td>
<td>Logging Prediction (catch-blocks)</td>
<td>Tomcat, CloudStack (Java)</td>
<td>85.50-93.34%</td>
</tr>
<tr>
<td>2.</td>
<td>Zhu et al. [2015]</td>
<td>Logging prediction (return-value-check/catch block)</td>
<td>Proprietary Microsoft software, MonoDev, SharpDev(C#)</td>
<td>84.6-93.4% (catch-blocks), 86.5-92.7% (return-value-check-blocks)</td>
</tr>
<tr>
<td>3.</td>
<td>Fu et al.[2014]</td>
<td>Logging prediction (return-value-check/catch block)</td>
<td>Proprietary Microsoft software (C#)</td>
<td>75-81% (catch-blocks), 88-89% (return-value-check-blocks)</td>
</tr>
<tr>
<td>4.</td>
<td>Yuan et al.[2012]</td>
<td>Verbosity level prediction</td>
<td>Apache httpd, OpenSSH, PostgreSQL, Squid(C/C++)</td>
<td>24 code constructs</td>
</tr>
</tbody>
</table>
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
Research Contribution

1. We identified **28 distinguishing** features for if block logging prediction

2. We propose **LogOptPlus**, a machine learning tool based on static features from source code for catch block logging prediction

3. We present results of **comprehensive evaluation** of LogOptPlus on two large open source projects with five machine learning algorithms
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
We extracted 28 distinguishing features

Each feature have three properties

- **Domain**: Part of the source code from where the feature is extracted
- **Type**: Feature is textual, boolean and numeric
- **Class**: Feature belongs to positive or negative class

Example of a if block from Apache Tomcat project showing **three domains** of the extracted features

```java
package org.apache.catalina.authenticator;

public class BasicAuthenticator extends AuthenticatorBase {
    public boolean authenticate(Request request, ...) throws IOException {
        Principal principal = request.getUserPrincipal();
        String ssold = (String) request.getAttribute(Conf.REQ_SSO_ID, Constants.REQ_SSO_ID, Constants.REQ_SITE_ID);
        if (principal != null) {
            if (log.isDebugEnabled()) {
                log.debug("Already authenticated " + principal.getName() + ");
            }
            if (ssold != null) { associate(ssold, request.getSessionInternal(true));
        }
        ...
        if (ssold != null) {
            if (log.isDebugEnabled()) {
                log.debug("SSOId " + ssold + " set: attempting " + "reauthentication");
            }
        }
        ...
```
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - Domain: Method_BI
   - Type: Numeric
   - Class: Positive

2. If Condition is Null [NC]
   - Domain: IF
   - Type: Boolean
   - Class: Positive

3. Return Statement in If-block [RI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative

4. Throw Statement in IF Block [TTI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - Domain: Method_BI
   - Type: Numeric
   - Class: Positive

2. If Condition is Null [NC]
   - Domain: IF
   - Type: Boolean
   - Class: Positive

3. Return Statement in IF block [RI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative

4. Throw Statement in IF Block [TTI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - Domain: Method_BI
   - Type: Numeric
   - Class: Positive

2. If Condition is Null [NC]
   - Domain: IF
   - Type: Boolean
   - Class: Positive

3. Return Statement in If-block [RI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative

4. Throw Statement in IF Block [TTI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - Domain: Method_BI
   - Type: Boolean
   - Class: Negative

2. If Condition is Null [NC]
   - Domain: IF
   - Type: Boolean
   - Class: Positive

3. Return Statement in If-block [RI]
   - Domain: IF
   - Type: Numeric
   - Class: Positive

4. Throw Statement in IF Block [TTI]
   - Domain: IF
   - Type: Boolean
   - Class: Negative
## LogOptPlus Model Feature

- **28 Features for catch-block logging predictions (T: Type, C:Class)**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Feature (Acronym)</th>
<th>T</th>
<th>C</th>
<th>S.No</th>
<th>Feature (Acronym)</th>
<th>T</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Size of Method_BI [LOC] [SM]</td>
<td>N</td>
<td>P</td>
<td>15</td>
<td>Method Parameters (type)</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>2</td>
<td>Logged Method_BI [LM]</td>
<td>B</td>
<td>P</td>
<td>16</td>
<td>Method Parameters (name)</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>3</td>
<td>Log Count in Method_BI [LCM]</td>
<td>N</td>
<td>P</td>
<td>17</td>
<td>Container Package Name [CPN]</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>4</td>
<td>Log Levels in Method_BI [LLM]</td>
<td>T</td>
<td>P</td>
<td>18</td>
<td>Container Class Name [CCN]</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>5</td>
<td>Operators in Method_BI [OM]</td>
<td>T</td>
<td>P</td>
<td>19</td>
<td>Container Method Name [CMN]</td>
<td>T</td>
<td>P</td>
</tr>
<tr>
<td>7</td>
<td>Variable Declaration Count in Method_BI [VCM]</td>
<td>N</td>
<td>P</td>
<td>21</td>
<td>Null Condition [NC]</td>
<td>B</td>
<td>P</td>
</tr>
<tr>
<td>8</td>
<td>Variable Declaration Name in Method_BI [VNM]</td>
<td>T</td>
<td>P</td>
<td>22</td>
<td>InstanceOf Condition [IOC]</td>
<td>B</td>
<td>P</td>
</tr>
<tr>
<td>9</td>
<td>Method have Parameter [PM]</td>
<td>B</td>
<td>P</td>
<td>23</td>
<td>Throw/Throws in IF Block [TTI]</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>10</td>
<td>Method Parameter Count [PCM]</td>
<td>N</td>
<td>P</td>
<td>24</td>
<td>Throw/Throws in Method_BI [TTM]</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>11</td>
<td>Method Call Count in Method_BI [MCM]</td>
<td>N</td>
<td>P</td>
<td>25</td>
<td>Return in IF Block [RI]</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>12</td>
<td>Method Call Name in Method_BI [MNM]</td>
<td>T</td>
<td>P</td>
<td>26</td>
<td>Return in Method_BI [RM]</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>13</td>
<td>IF in Method_BI [IM]</td>
<td>B</td>
<td>P</td>
<td>27</td>
<td>Assert in Method_BI [AM]</td>
<td>B</td>
<td>N</td>
</tr>
<tr>
<td>14</td>
<td>IF Count in Method_BI [ICM]</td>
<td>N</td>
<td>P</td>
<td>28</td>
<td>Assert in IF Block [AI]</td>
<td>B</td>
<td>N</td>
</tr>
</tbody>
</table>
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
**LogOptPlus Model Feature Analysis Metric**

- **Metric for boolean features**
  
  1. \( CLX_{f_k} (\%) = \frac{|CLX_{f_k}| \times 100}{|TXP_{f_k}|} \)
  
  2. \( PTLX_{f_k} (\%) = \frac{|CLX_{f_k}| \times 100}{|LXS|} \)

- **Metric for numeric features**
  
  1. \( AVLX_{Y,f_k} = \frac{\sum_{i=1}^{LXS} FV(X_i, f_k)}{|LXS|} \)

  2. \( AVNLX_{Y,f_k} = \frac{\sum_{i=1}^{NLXS} FV(X_i, f_k)}{|NLXS|} \)
LogOptPlus Model Features Example

- $AVLI_{SM} = 26.96$
- $AVNLI_{SM} = 20.90$

1. LOC of Method_BI [SM]
   - Total = 2435
   - $CLI_{TTI} = 3.66\%$
   - $CNLI_{TTI} = 96.35\%$
   - $PTLI_{TTI} = 6.35\%$
   - $PTNLI_{TTI} = 14.62\%$

2. If Condition is Null [NC]
   - Total = 6212
   - $CLI_{NC} = 8.62\%$
   - $CNLI_{NC} = 91.39\%$
   - $PTLI_{NC} = 38.16\%$
   - $PTNLI_{NC} = 35.38\%$

3. Return Statement in If-block [RI]
   - Total = 3950
   - $CLI_{RI} = 5.12\%$
   - $CNLI_{RI} = 94.89\%$
   - $PTLI_{RI} = 14.4\%$
   - $PTNLI_{RI} = 23.36\%$

4. Throw Statement in IF Block [TTI]
   - Total = 20
   - $CLI_{TTI} = 5.12\%$
   - $CNLI_{TTI} = 94.89\%$
   - $PTLI_{TTI} = 14.4\%$
   - $PTNLI_{TTI} = 23.36\%$
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - AVLI_{SM} = 26.96
   - AVNLI_{SM} = 20.90

2. If Condition is Null [NC]
   - Total = 6212
   - CLI_{NC} = 8.62%
   - CNLI_{NC} = 91.39%
   - PTLI_{NC} = 38.16%
   - PTNLI_{NC} = 35.38%

3. Return Statement in If-block [RI]
   - Total = 3950
   - CLI_{RI} = 5.12%
   - CNLI_{RI} = 94.89%
   - PTLI_{RI} = 14.4%
   - PTNLI_{RI} = 23.36%

4. Throw Statement in IF Block [TTI]
   - Total = 2435
   - CLI_{TTI} = 3.66%
   - CNLI_{TTI} = 96.35%
   - PTLI_{TTI} = 6.35%
   - PTNLI_{TTI} = 14.62%
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - Total = 2435
   - CLI_TTI = 3.66%
   - CNLI_TTI = 96.35%
   - PTLI_TTI = 6.35%
   - PTNLI_TTI = 14.62%

2. If Condition is Null [NC]
   - Total = 3950
   - CLI_RI = 5.12%
   - CNLI_RI = 94.89%
   - PTLI_RI = 14.4%
   - PTNLI_RI = 23.36%

3. Return Statement in If-block [RI]
   - Total = 6212
   - CLI_NC = 8.62%
   - CNLI_NC = 91.39%
   - PTLI_NC = 38.16%
   - PTNLI_NC = 35.38%

4. Throw Statement in IF Block [TTI]
   - Total = 2005
   - CLI_TTI = 5.12%
   - CNLI_TTI = 94.89%
   - PTLI_TTI = 14.4%
   - PTNLI_TTI = 23.36%

- AVLI_{SM} = 26.96
- AVNLI_{SM} = 20.90
LogOptPlus Model Features Example

1. LOC of Method_BI [SM]
   - AVLI_{SM} = 26.96
   - AVNLI_{SM} = 20.90

2. If Condition is Null [NC]
   - Total = 6212
   - CLI_{NC} = 8.62%
   - CNLI_{NC} = 91.39%
   - PTLI_{NC} = 38.16%
   - PTNLI_{NC} = 35.38%

3. Return Statement in If-block [RI]
   - Total = 3950
   - CLI_{RI} = 5.12%
   - CNLI_{RI} = 94.89%
   - PTLI_{RI} = 14.4%
   - PTNLI_{RI} = 23.36%

4. Throw Statement in IF Block [TTI]
   - Total = 2435
   - CLI_{TTI} = 3.66%
   - CNLI_{TTI} = 96.35%
   - PTLI_{TTI} = 6.35%
   - PTNLI_{TTI} = 14.62%
LogOptPlus Model Feature Analysis Results

- Metric for Boolean features

<table>
<thead>
<tr>
<th>S.No</th>
<th>Feature (Acronym)</th>
<th>Class (Pos/Neg.)</th>
<th>Apache Tomcat</th>
<th>CloudStack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TIP</td>
<td>CLI (%)</td>
</tr>
<tr>
<td>1</td>
<td>[LM]</td>
<td>P</td>
<td>2121</td>
<td>625(29.47)</td>
</tr>
<tr>
<td>2</td>
<td>[IM]</td>
<td>P</td>
<td>11633</td>
<td>1042(8.96)</td>
</tr>
<tr>
<td>3</td>
<td>[PM]</td>
<td>P</td>
<td>13043</td>
<td>1031(7.91)</td>
</tr>
<tr>
<td>4</td>
<td>[NC]</td>
<td>P</td>
<td>6212</td>
<td>535(8.62)</td>
</tr>
<tr>
<td>5</td>
<td>[IOC]</td>
<td>P</td>
<td>984</td>
<td>80(8.14)</td>
</tr>
<tr>
<td>6</td>
<td>[TTI]</td>
<td>N</td>
<td>2435</td>
<td>89(3.66)</td>
</tr>
<tr>
<td>7</td>
<td>[TTM]</td>
<td>N</td>
<td>2222</td>
<td>111(5)</td>
</tr>
<tr>
<td>8</td>
<td>[RI]</td>
<td>N</td>
<td>3950</td>
<td>202(5.12)</td>
</tr>
<tr>
<td>9</td>
<td>[RM]</td>
<td>N</td>
<td>3278</td>
<td>187(5.71)</td>
</tr>
<tr>
<td>10</td>
<td>[AM]</td>
<td>N</td>
<td>81</td>
<td>0(0)</td>
</tr>
<tr>
<td>11</td>
<td>[AI]</td>
<td>N</td>
<td>86</td>
<td>4(4.66)</td>
</tr>
</tbody>
</table>
## LogOpt Model Feature Analysis Results

- Metric for numeric features

<table>
<thead>
<tr>
<th>S.No</th>
<th>Feature (Acronym)</th>
<th>Apache Tomcat</th>
<th>CloudStack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AVL</td>
<td>AVNLI</td>
</tr>
<tr>
<td>1</td>
<td>[SM]</td>
<td>26.96</td>
<td>20.90</td>
</tr>
<tr>
<td>2</td>
<td>[LCM]</td>
<td>1.84</td>
<td>0.22</td>
</tr>
<tr>
<td>3</td>
<td>[COM]</td>
<td>67.78</td>
<td>53.71</td>
</tr>
<tr>
<td>4</td>
<td>[VCM]</td>
<td>2.68</td>
<td>1.94</td>
</tr>
<tr>
<td>5</td>
<td>[MCM]</td>
<td>13.61</td>
<td>9.57</td>
</tr>
<tr>
<td>6</td>
<td>[ICM]</td>
<td>5.19</td>
<td>4.06</td>
</tr>
<tr>
<td>7</td>
<td>[PCM]</td>
<td>1.62</td>
<td>1.54</td>
</tr>
</tbody>
</table>
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
LogOptPlus Model Framework
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
## Experimental Dataset

<table>
<thead>
<tr>
<th>S.No</th>
<th></th>
<th>Apache Tomcat</th>
<th>CloudStack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Version</td>
<td>8.0.9</td>
<td>4.3.0</td>
</tr>
<tr>
<td>2</td>
<td>LOC (Java Code)</td>
<td>276081</td>
<td>1142928</td>
</tr>
<tr>
<td>3</td>
<td>Number of Java Files</td>
<td>2037</td>
<td>5351</td>
</tr>
<tr>
<td>4</td>
<td>Total If Blocks</td>
<td>17449</td>
<td>96885</td>
</tr>
<tr>
<td>5</td>
<td>Logged If Blocks</td>
<td>1402 (8.03%)</td>
<td>5611 (5.79%)</td>
</tr>
</tbody>
</table>
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
LogOptPlus Model Prediction Results

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier</td>
<td>DT : (Max-Depth=86), KNN : (Neighbor=3)</td>
</tr>
<tr>
<td></td>
<td>GNB , ADA: (Estimators=99)</td>
</tr>
<tr>
<td></td>
<td>Random Forest (Estimators=82)</td>
</tr>
<tr>
<td>Feature Set</td>
<td>28 Features</td>
</tr>
<tr>
<td>Train – Test</td>
<td>70-30%</td>
</tr>
<tr>
<td>Classifier Iterations</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
<th>Class</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apache Tomcat</strong></td>
<td>ADA</td>
<td>74.08</td>
<td>74.58</td>
<td>72.85</td>
<td>73.64</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>74.15</td>
<td>73.44</td>
<td>75.10</td>
<td>74.32</td>
</tr>
<tr>
<td></td>
<td>GNB</td>
<td>70.95</td>
<td>65.11</td>
<td>89.69</td>
<td>75.43</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>60.84</td>
<td>63.54</td>
<td>50.11</td>
<td>55.99</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>81.42</td>
<td>83.98</td>
<td>78.78</td>
<td>80.70</td>
</tr>
<tr>
<td><strong>Cloud-Stack</strong></td>
<td>ADA</td>
<td>89.77</td>
<td>87.21</td>
<td>93.23</td>
<td>90.12</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>89.10</td>
<td>87.97</td>
<td>90.55</td>
<td>89.27</td>
</tr>
<tr>
<td></td>
<td>GNB</td>
<td>78.83</td>
<td>88.16</td>
<td>66.66</td>
<td>75.91</td>
</tr>
<tr>
<td></td>
<td>KNN</td>
<td>83.01</td>
<td>84.27</td>
<td>81.21</td>
<td>82.71</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>91.67</td>
<td>86.53</td>
<td>98.77</td>
<td>92.25</td>
</tr>
</tbody>
</table>
OUTLINE

- Introduction
- Research Motivation
- Related Work
- Research Contribution
- Feature Details
- Feature Analysis
- Proposed Framework
- Experimental Dataset
- Results
- Conclusion
CONCLUSIONS

- Machine learning based approach is effective in catch block logging prediction giving highest F1-score of 92.25% (CloudStack project)

- Random forest model give best results as compared to other machine learning algorithms
REFERENCES


Thank You

sangeeta@jiit.ac.in
neetu.sardana@jiit.ac.in
ashish.sureka@in.abb.com