Rapid Construction of Accurate Automatic Alert Handling System: Architecture and Prototype

Problem
Static analysis alerts for security-related code flaws require too much manual effort to triage, and there is little use of automated alert classifier technology because of barriers of cost, expertise, and lack of labeled data.

Solution
Develop extensible architecture for classification and advanced prioritization, building on novel test-suite data method we developed.
- Implement prototype
- Enable organizations to quickly start using classifiers and advanced prioritization by making API calls from their alert auditing tools
- Develop adaptive heuristics for classifier to adapt as it learns from test suite and "natural program" data

Approach
1. Design architecture
2. Develop API definition
3. Implement prototype system
4. Develop adaptive heuristics
5. Test adaptive heuristics with datasets combining test suite and real-world (DoD) data
6. Collaborators test architecture and prototype

Juliet test suite classifiers: initial results (hold-out data)
All four classification methods had high accuracy.

Table:

<table>
<thead>
<tr>
<th>CLASSIFIER</th>
<th>ACCURACY</th>
<th>PRECISION</th>
<th>RECALL</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.938</td>
<td>0.893</td>
<td>0.875</td>
<td>0.991</td>
</tr>
<tr>
<td>Lightgbm</td>
<td>0.942</td>
<td>0.902</td>
<td>0.882</td>
<td>0.992</td>
</tr>
<tr>
<td>Xgboost</td>
<td>0.932</td>
<td>0.941</td>
<td>0.798</td>
<td>0.987</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.925</td>
<td>0.886</td>
<td>0.831</td>
<td>0.985</td>
</tr>
</tbody>
</table>

Artifacts

Code and Test Results
- API definition (Swagger, RESTful)
- SCALe v3 static analysis alert auditing tool with new features (collaborators-only) with advanced prioritization schemes and features for classification
- Code development for prototype system
- Expanded archive of auto-labeled alerts
- Test results from cross-taxonomy test suite classifiers using precise mappings
- Code enabling novel "speculative mapping" method for tools without mappings to test suite metadata's code flaw taxonomy
- Adaptive heuristic development and testing results (in progress)

Non-Code Publications + Papers

Architecture API definition and new SCALe features
- Special Report: "Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools" (Aug. 2018)

Classifier development research methods and results:
- Paper "Prioritizing Alerts from Multiple Static Analysis Tools, using Classification Models," SQUADE (ICSE workshop)
- SEI blog post: "Test Suites as a Source of Training Data for Static Analysis Alert Classifiers" (Apr. 2018)
- SEI Podcast (video): "Static Analysis Alert Classification with Test Suites" (Sep. 2018)
- In-progress conference papers (4): precise mapping, architecture for rapid alert classification, test suites for classifier training data, API development

Precise mappings on CERT C Standard wiki
- Metadata for Juliet (created to test CWES) to test CERT rule coverage
- Per-rule precise CWE mapping

Continuing in FY19
Using test suite data for classifiers, research:
Adaptive heuristics
- How classifiers incorporate new data
- Test suite vs. non-test-suite data
- Weighting recent data

Semantic features for cross-project prediction
- Test suites as different projects

This project developed an architecture and API definition for static analysis alert classification and advanced alert prioritization, plus major parts of a prototype system.