Applied Machine Learning in Software Security

Eliezer Kanal

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® and CERT® are registered marks of Carnegie Mellon University.

DM-0004563
What is Machine Learning?

Tom Mitchell, former CMU Machine Learning department chair:

The field of Machine Learning asks the question, “How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?”

Machine Learning seeks to automate data analysis and inference.
What is Machine Learning?

If your problem can be stated as either of the following:

I would like to use _____ data to guess what _____ is.

I would like to use _____ data to predict _____.

…you would likely benefit from machine learning.
What is Machine Learning?

Sample Techniques:

• Regression

• K-Means Clustering
What is Machine Learning?

Feature Engineering:
Using existing data to create more informative data

Data Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Static Video</td>
</tr>
<tr>
<td>Time series</td>
<td>Financial data Event counts</td>
</tr>
<tr>
<td>Structured text</td>
<td>Web forms Structured data (JSON, XML) Source code</td>
</tr>
<tr>
<td>Free text</td>
<td>News Tweets Email</td>
</tr>
<tr>
<td></td>
<td>many more...</td>
</tr>
</tbody>
</table>
What is Machine Learning?

Examples:

- I would like to use **incident ticket** data to predict **customer needs**.

- I would like to use **publicly available code** to predict **what code I will write**.

- I would like to use **bug report** data to guess the location of **undetected bugs in my code**.
Autocomplete from Stack Overflow

by Emil Schutte

Tired of writing code? Me too! Let's have Stack Overflow do it.

```javascript
// Boss wants this function done by tomorrow :(
function contains(needle, haystack) {
    var
```

(Try typing a space. JavaScript only, for now.)

**How it works**

I grabbed a Stack Overflow data dump from [https://archive.org/details/stackexchange](https://archive.org/details/stackexchange) and scraped out any code snippets from

- accepted answers
- with more than 50 points
- on posts tagged "javascript"

Then I processed it by walking the ASTs of those snippets and creating a "completion" fragment for each node, pairing a trace of the left-hand context with the code snippet for the right-hand side.

To complete at run time, it uses the same logic to find the left-hand trace at the current cursor position, and tries to match that up against the database of completion fragments. Available completions are sorted by a proprietary blend of post score, left-hand context similarity, and nearby identifiers.
What is Machine Learning?

Examples:

- I would like to use **incident ticket** data to predict **customer needs**.

- I would like to use **publicly available code** to predict **what code I will write**.

- I would like to use **bug report** data to guess the location of **undetected bugs in my code**.
Applied ML: Vulnerability Detection
Applied ML: Vulnerability Detection

Many alerts left unaudited!
Applied ML: Vulnerability Detection

Automated Statistical Classifier

- Expected True Positive (e-TP)
- Expected False Positive (e-FP)
- Indeterminate (I)

Our Goal

Codebases

Analyzer

Analyzer

Analyzer

Alerts

Today

Graphs showing statistics for e-TP, e-FP, and I

Graphs showing statistics for TP, FP, and Susp
## Applied ML: Vulnerability Detection

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Some of the features used</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lasso Logistic Regression</strong></td>
<td>Analysis tools used: Tokens in func/method</td>
</tr>
<tr>
<td><strong>CART</strong></td>
<td>Significant LOC: Alerts in func/method</td>
</tr>
<tr>
<td><strong>Random Forest</strong></td>
<td>Complexity: Alerts in file</td>
</tr>
<tr>
<td><strong>Extreme Gradient Boosting (XGBoost)</strong></td>
<td>Coupling: Methods in file</td>
</tr>
<tr>
<td></td>
<td>Cohesion: SLOC in file</td>
</tr>
<tr>
<td></td>
<td>SEI coding rule: Avg Tokens</td>
</tr>
<tr>
<td></td>
<td>Function/method length: Avg SLOC</td>
</tr>
<tr>
<td></td>
<td>SLOC in func/method: Depth in code repository</td>
</tr>
<tr>
<td></td>
<td># parameters in func/method: Cyclomatic complexity (func/meth)</td>
</tr>
</tbody>
</table>
Applied ML: Vulnerability Detection

Significant improvement!

- 91% Classifier accuracy overall
- Specific rule accuracy at right
- 10x developer time saved!

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>INT31-C</td>
<td>97%</td>
</tr>
<tr>
<td>EXP01-J</td>
<td>74%</td>
</tr>
<tr>
<td>OBJ03-J</td>
<td>83%</td>
</tr>
<tr>
<td>FIO04-J*</td>
<td>80%</td>
</tr>
<tr>
<td>EXP33-C*</td>
<td>83%</td>
</tr>
<tr>
<td>EXP34-C*</td>
<td>72%</td>
</tr>
<tr>
<td>DCL36-C*</td>
<td>100%</td>
</tr>
<tr>
<td>ERR08-J*</td>
<td>100%</td>
</tr>
<tr>
<td>IDS00-J*</td>
<td>96%</td>
</tr>
<tr>
<td>ERR01-J*</td>
<td>100%</td>
</tr>
<tr>
<td>ERR09-J*</td>
<td>88%</td>
</tr>
</tbody>
</table>

* Small quantity of data
Applied ML: Malware family classification
Applied ML: Malware family classification

1. **File**
2. **Reverse Engineering**
   - Signature 1
   - Signature 2
   - Signature 3
   - ...
3. **Discovery**
4. **Refinement**
   - Files 1a 1b 1c 1d ...
   - Files 2a 2b 2c 2d ...
   - Files 3a 3b 3c 3d ...
   - ...
5. **Artifact Catalog**

Which files hang together?
Applied ML: Malware family classification

Program instruction analysis shows similarity and diversion of behavior

Static Analysis identifies programs with similar source code

Signal Flow graph highlights behavior relating different malware families
Applied ML: Malware family classification

Simplify visualization of extremely complex data through the use of dimensionality reduction and associated visualization techniques.
Applied ML: Malware family classification

- **Ground Truth**: SVM trained with expert ground truth labels.
- **Turkers Avg**: Classifier trained with layperson labels.

Performance surprisingly similar!
Applied ML:
Software cost estimation

Table 2: Cost Overruns in DoD Acquisitions

Funding Shortfalls at the Start of Development for Five Major Weapon System Programs

<table>
<thead>
<tr>
<th>Program</th>
<th>Percentage of development funding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-mission</td>
<td></td>
</tr>
<tr>
<td>Warfighter Information</td>
<td></td>
</tr>
<tr>
<td>Future Combat Systems</td>
<td></td>
</tr>
<tr>
<td>Joint Strike Fighter</td>
<td></td>
</tr>
<tr>
<td>Global Hawk</td>
<td></td>
</tr>
</tbody>
</table>

Source: DOD (data); GAO (analysis and presentation).

Applied ML: Software cost estimation

Information Flow for Early Lifecycle Estimation

Proposed Material Solution & Analysis of Alternatives

Information from Analogous Programs/Systems

Program Execution Cost Drivers

System Characteristics
- Trade-offs
  - KPP selection
  - Systems Design
  - Sustainment issues

Operational Capability
- Trade-offs
  - Mission / CONOPS
  - Capability Based Analysis

Technology Development
- Strategy
  - Production Quantity
  - Acquisition Mgt
  - Scope definition/responsibility
  - Contract Award

Driver States & Probabilities

Probabilistic Modeling (BBN) & Monte Carlo Simulation

Cost Estimates
- analogy
- parametric
- engineering
- others

Plans, Specifications, Assessments

Program Execution Scenarios with conditional probabilities of drivers/states
# Applied ML: Software cost estimation

<table>
<thead>
<tr>
<th>Causes</th>
<th>Effects</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope Responsibility</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Scope Definition</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Mission / CONOPS</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Capability Definition</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Funding Schedule</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Program Mgt - Contractor Relations</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Systems Design</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Standards/Certifications</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Acquisition Management</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Program Mgt - Contractor Relations</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Project Social / Dev Env</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>Supply Chain Vulnerabilities</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Information Sharing</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>PO Process Performance</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Sustainment Issues</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Contract Award</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Contractor Performance</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Production Quantity</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Data Ownership</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Change in Strategic Vision</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Advocacy Change</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Industry Company Assessment</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Cost Estimate</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Test &amp; Evaluation</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Closing Technical Gap (CBA)</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>Building Technical Capability &amp; Capacity (CBA)</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>Functional Measures</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Functional Solution Criteria (measure)</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Interdependency</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Interoperability</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Size</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Project Challenge</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Product Challenge</td>
<td></td>
<td>34</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td>17</td>
</tr>
</tbody>
</table>

Applied Machine Learning in Software Security  
August 10, 2017  
© 2016 Carnegie Mellon University  
Approved for Public Release; Distribution is Unlimited
Applied ML: Software cost estimation
Applied ML:
Incident report mapping
Applied ML: Incident report mapping

- Threat actors
- Agency infrastructure
- Compromised website
- International partners
- Agency response teams
- Malware class
- US-CERT infrastructure
- Common reference websites
- Phishing campaign

Landscape

Observation

Inference

Tickets
Applied ML: Incident report mapping

1. Unobserved real world indicator generating processes
2. Observed tickets
3. Estimated indicator groups

Agency response teams
US-CERT infrastructure
Common reference websites
Phishing campaign
Malware class
Compromised website
Indicators across tickets

Indicators occur with diverse patterns across tickets, reporters and time.  

*Time on x axis, count on y axis, color coded by reporter.*
Similarity of indicators

Beginning with a reference indicator, we find indicators similar to it. Example: a malicious IP

- Colored circles are tickets
- Grey circles are indicators
- Large indicators near center of circle have similar occurrence patterns to the reference indicator.
indicator communities

But what if we aren’t starting with a reference indicator? We assume that indicators generated by a coherent real world process will be more likely to co-occur in tickets than arbitrary pairs of indicators. Find groups of highly similar indicators in complete indicator-ticket graph.
Indicator-ticket graph

A subset of the ticket-indicator graph
(for a small set of selected indicators)
• Tickets are grey triangles
• Indicators are black circles
• Edges connect tickets to the indicators they contain
Contact Information

Eliezer Kanal
Technical Manager
Telephone:  +1 412.268.5204
Email:   ekanal@sei.cmu.edu