Applying Natural Language Processing to Software Engineering: the Case of Classifying Security Bug Reports

Mayana Pereira
Data Scientist
Microsoft

#GHC19
A Complex Engineering Environment

~47,000 Engineers all potentially contributing security bugs
Over 2,000 products in development

100+ different Bug Tracking Systems
Unique labels and queries for security bugs
Dependency on Engineers correctly labeling security bugs

Data Source
Data from engineering and compliance systems across Microsoft
What is a Security Bug?

Identifying security-related issues among reported bugs is a pressing need among software development teams.

Security issues call for more expedited fixes to meet compliance requirements and ensure the integrity of the software and customer data.
Impact of machine learning-based solution for security bug identification

- More complete identification of security bugs.
- More accurate identification of security bugs
- Expedite security bug solving.

#GHC19
Main Challenges

- Dataset consists of Bug titles only
  - Not all Bugs have descriptions in our database
  - Sensitive data such as credentials are commonly found in bug descriptions

- Manual identification of security issues error-prone. It is estimated that around 30% of bugs are mislabeled.
  - Development team’s lack of expertise in security
  - Fuzziness of certain problems

- Security space has a very high acceptance criteria
  - Precision & Recall > 90%
Data

- The data was collected from various teams across Microsoft in the years 2015, 2016, 2017 and 2018.
- All reports were closed and verified at some point by a Security Engineer.
- The training dataset consists of Bug titles and bug labels.

<table>
<thead>
<tr>
<th>XSS</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Overflow</td>
<td>Security</td>
</tr>
<tr>
<td>Button in wrong place</td>
<td>Non-Security</td>
</tr>
<tr>
<td>Wrong font in main page</td>
<td>Non-Security</td>
</tr>
<tr>
<td>Button does not work</td>
<td>Non-Security</td>
</tr>
</tbody>
</table>
Objectives

Goal 1
Obtain a Classifier that is “as close as possible” to a Security Expert for the task of classifying a bug report as security/non-security by using only bug titles as input data.

Goal 2
Understand whether noisy data sets can be used to train security bugs identification systems. How different classifiers behave in the presence of noise for identification of Security Bugs?
Goal 1

Machine learning classifiers for security bug identification
Is it possible to train a ML model for security bug identification?
More Challenges

How do we do it?

Unique vocabulary
specific terms that relates to software vulnerabilities and security flaws -- needed our own corpus

Interpretability is important!

• Represent titles as fixed sized vectors using text frequency – inverse document frequency (TF-IDF) technique.
  • For a given term $t$ in document $d$, TF-IDF will attribute a weight to $t$ by measuring the frequency of the term in $d$ and the number of documents that contains $t$ in the entire data set.

Interpretable machine learning techniques and more complex machine learning techniques
  • Logistic Regression
  • Naïve Bayes
  • Boosted Decision Trees

#GHC19
Test models trained using different techniques
Interpretable models perform well.

<table>
<thead>
<tr>
<th>ML Model</th>
<th>Metric</th>
<th>Acc</th>
<th>TPR</th>
<th>FPR</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td></td>
<td>0.9309</td>
<td>0.9353</td>
<td>0.0735</td>
<td>0.9831</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td></td>
<td>0.9210</td>
<td>0.9189</td>
<td>0.0769</td>
<td>0.9770</td>
</tr>
<tr>
<td>AdaBoost</td>
<td></td>
<td>0.8122</td>
<td>0.7018</td>
<td>0.0774</td>
<td>0.9143</td>
</tr>
</tbody>
</table>

(a) ROC curve for Logistic Regression Model
(b) ROC curve for Naive Bayes Model
(c) ROC curve for AdaBoost Model
Goal 2

Understand whether noisy data sets can be used to train security bugs identification systems
Can we train our model using noisy data?

We introduce noise in our dataset by flipping each entry in the data set that the true label is security with probability $p_{br}$. 
Can we train our model using noisy data?
Training data set does need to be perfect
Imperfect Data Can Work!

Class-independent noise

<table>
<thead>
<tr>
<th>$p_{sbr}$ value</th>
<th>Machine Learning Model</th>
<th>AUC Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>logistic regression</td>
<td>naive Bayes</td>
</tr>
<tr>
<td>0.0</td>
<td>0.9827</td>
<td>0.9739</td>
</tr>
<tr>
<td>0.05</td>
<td>0.9820</td>
<td>0.9716</td>
</tr>
<tr>
<td>0.10</td>
<td>0.9808</td>
<td>0.9703</td>
</tr>
<tr>
<td>0.15</td>
<td>0.9792</td>
<td>0.9692</td>
</tr>
<tr>
<td>0.20</td>
<td>0.9763</td>
<td>0.9676</td>
</tr>
<tr>
<td>0.25</td>
<td>0.9714</td>
<td>0.9658</td>
</tr>
<tr>
<td>0.30</td>
<td>0.9621</td>
<td>0.9626</td>
</tr>
<tr>
<td>0.35</td>
<td>0.9412</td>
<td>0.9566</td>
</tr>
<tr>
<td>0.40</td>
<td>0.8917</td>
<td>0.9425</td>
</tr>
<tr>
<td>0.45</td>
<td>0.7645</td>
<td>0.8939</td>
</tr>
<tr>
<td>0.50</td>
<td>0.4994</td>
<td>0.4806</td>
</tr>
</tbody>
</table>
Takeaways

Bug Titles contain a lot of information!
- We have shown the feasibility of security bug report classification based solely on the title of the bug report. This is particularly relevant in scenarios where the entire bug report is not available due to privacy constraints.

- Our classification model that utilizes a combination of TF-IDF and logistic regression performs at an AUC of 0.9831.

- Imperfect Data can work!
  - All three classifiers are robust to single-class noise.
  - The decrease in AUC is very small (0.01) for a level of noise of 50% (single-class noise).

- Finally, class-dependent noise significantly impacts the AUC only when there is more than 35% noise in both classes.
- The first systematic study on the effect of noisy data sets for security bug report identification.
Please remember to complete the session survey in the mobile app.

THANK YOU!