Prioritizing Alerts from Static Analysis with Classification Models

PI: Lori Flynn, PhD

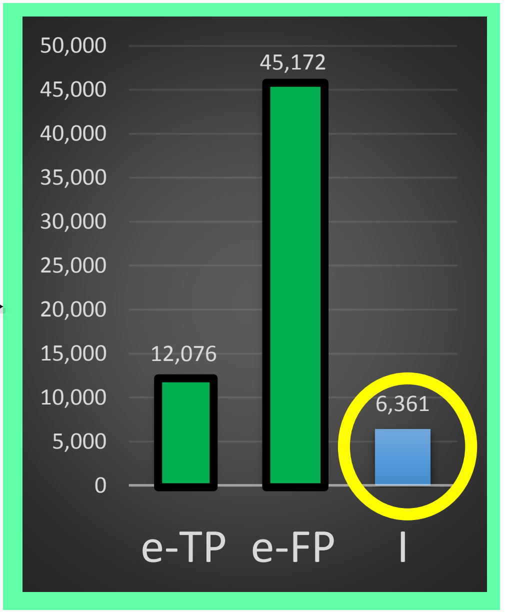
Team: Will Snavely, David Svoboda, Dr. David Zubrow, Bob Stoddard, Dr. Nathan VanHoudnos, Dr. Elli Kanal, Richard Qin, Jennifer Burns, Guillermo Marce-Santurio, and 3 DoD Organizations

Overview

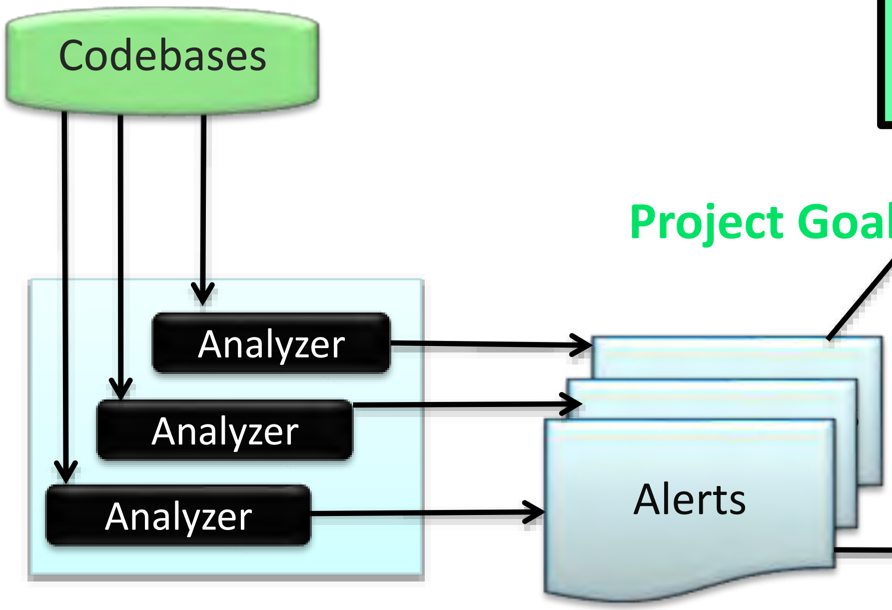
Long-term goal: Automated and accurate statistical classifier, intended to efficiently use analyst effort and to remove code flaws

Classification algorithm development using CERT- and collaborator-audited data, that accurately classifies most of the diagnostics as:

Expected True Positive (e-TP) or Expected False Positive (e-FP),
and
the rest as Indeterminate (I)

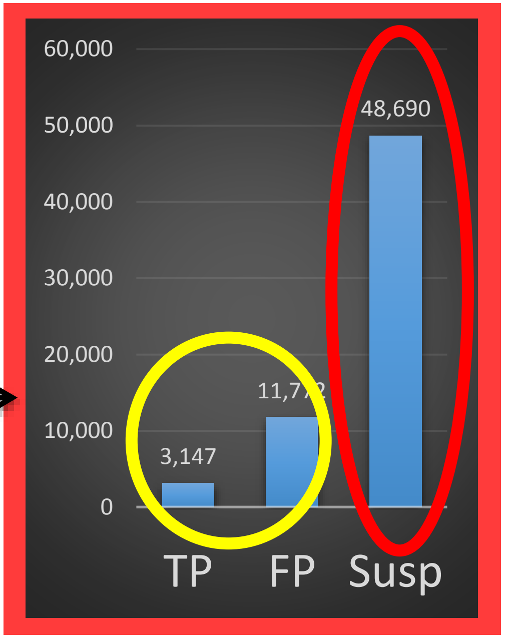


Prioritized, small number of alerts for manual audit



Project Goal

Today



Many alerts left un-audited!

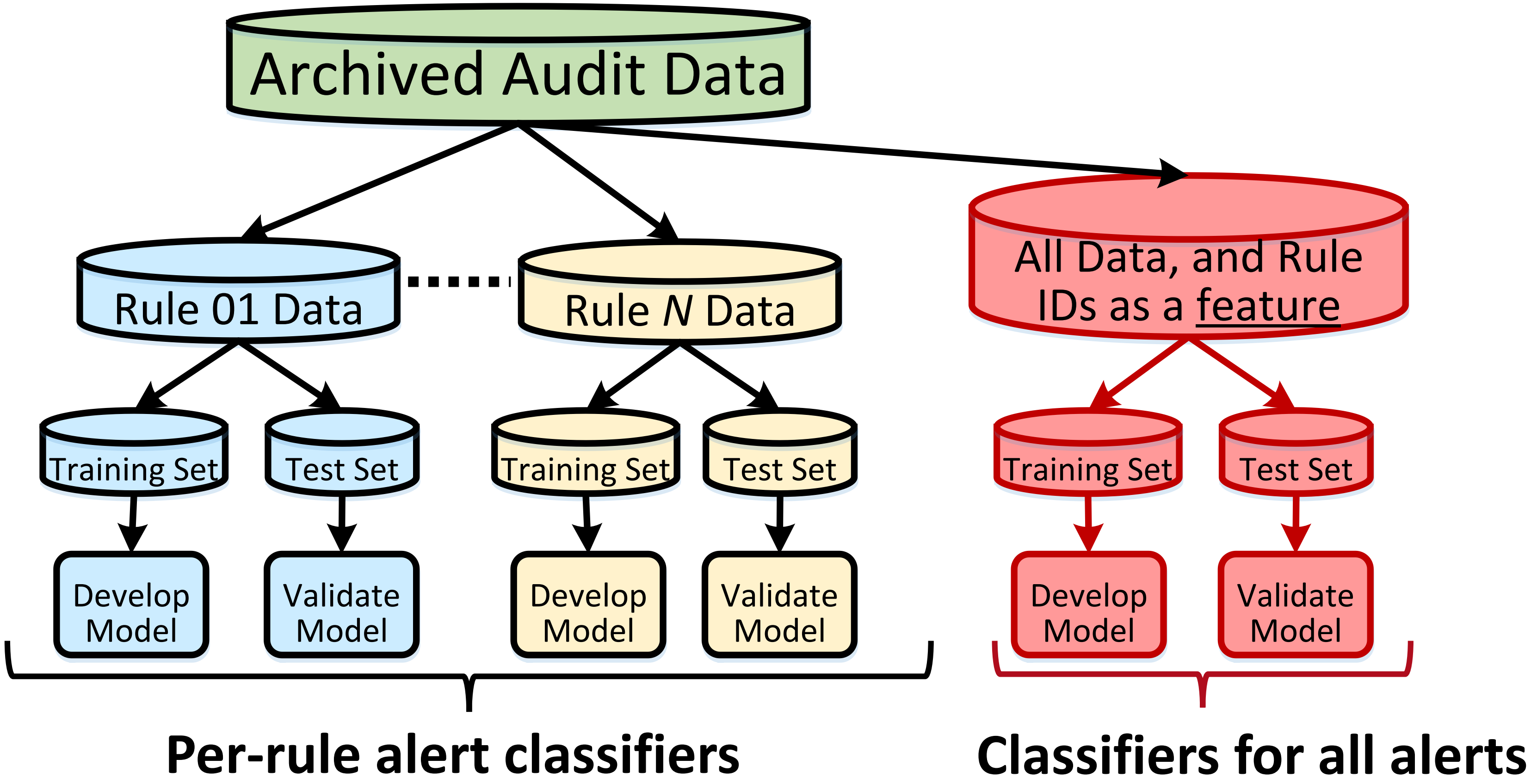
Image of woman and laptop from http://www.publicdomainpictures.net/view-image.php?image=47526&picture=woman-and-laptop “Woman And Laptop”

Scientific Approach

<u>Novel combined use of</u>:

1) multiple analyzers, 2) variety of features,

3) competing classification techniques!



Competing Classifiers to Test
Lasso Logistic Regression
CART (Classification and Regression Trees)
Random Forest
Extreme Gradient Boosting (XGBoost)

Some of the features used (many more)
Analysis tools used
Significant LOC
Complexity
Coupling
Cohesion
SEI coding rule

Data Used for Classifiers

Data used to create and validate classifiers:

- CERT-audited alerts:
  - ~7,500 audited alerts
- 3 DoD collaborators audit their own codebases with enhanced-SCALe

We pooled data (CERT + collaborators) and segmented it:

- Segment 1 (70% of data): train model
- Segment 2 (30% of data): testing

Added classifier variations on dataset:

- Per-rule
- Per-language
- With/without tools
- Others

288 classifiers developed and tested

Classifier Test Highlights

Classifiers made from all data, pooled:

All-rules (158) classifier accuracy:

- Lasso Logistic Regression: 88%
- Random Forest: 91%
- CART: 89%
- XGBoost: 91%

Single-rule classifier accuracy:

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	98%	97%	98%	97%
EXP01-J	74%	74%	81%	74%
OBJ03-J	73%	86%	86%	83%
FIO04-J*	80%	80%	90%	80%
EXP33-C*	83%	87%	83%	83%
EXP34-C*	67%	72%	79%	72%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	99%	100%	100%	100%
IDS00-J*	96%	96%	96%	96%
ERR01-J*	100%	100%	100%	100%
ERR09-J*	100%	88%	88%	88%

General results (not true for every test)

- Classifier accuracy rankings for all-pooled test data:
  XGBoost ≈ RF > CART ≈ LR
- Classifier accuracy rankings for collaborator test data:
  LR ≈ RF > XGBoost > CART
- Per-rule classifiers generally not useful (lack data), but 3 rules (INT31-C best) are exceptions.
- With-tools-as-feature classifiers better than without.
- Accuracy of single language vs. all-languages data:
  C > all-combined > Java

* Small quantity of data, results suspect

Results with DoD Transition Value

Software and paper: Classifier-development
- Code for developing classifiers in the R environment
- Paper: classifier development, analysis, & use [1]

Software: Enhanced-SCALe Tool (auditing framework)
- Added data collection
- Archive sanitizer
- Alert fusion
- Offline installs and virtual machine

Training to ensure high-quality data
- SEI CERT coding rules
- Auditing rules [2]
- Enhanced-SCALe use

Auditor quality test
- Test audit skill: mentor-expert designation

Conference/workshop papers:

[1] Flynn, Snavely, Svoboda, Qin, Burns, VanHoudnos, Zubrow, Stoddard, and Marce-Santurio. “Prioritizing Alerts from Multiple Static Analysis Tools, using Classification Models”, work in progress.

[2] Svoboda, Flynn, and Snavely. “Static Analysis Alert Audits: Lexicon & Rules”, IEEE Cybersecurity Development (SecDev), November 2016.

Future Work

Goal: improve accuracy

- Try different classification techniques
- Different mix of features:
  - Semantic features (ICSE 2016 paper)
  - Dynamic analysis tool results as features
- More audit archive data needed
  - Additional data welcome! Potential collaborators, please contact me
  - **FY17 project focuses on rapid expansion of per-rule classifiers**

Contact Information

Lori Flynn, PhD

Software Security Researcher

Telephone: +1 412.268.7886

Email: lflynn@sei.cmu.edu

Discussion

Results with DoD Transition Value: Sanitizer

New data sanitizer

- Anonymizes sensitive fields
- SHA-256 hash with salt
- Enables analysis of features correlated with alert confidence

SCALe project is in a SCALe database

- DB fields may contain sensitive information
- Sanitizing script anonymizes or discards fields
  - Diagnostic message
  - Path, including directories and filename
  - Function name
  - Class name
  - Namespace/package
  - Project filename

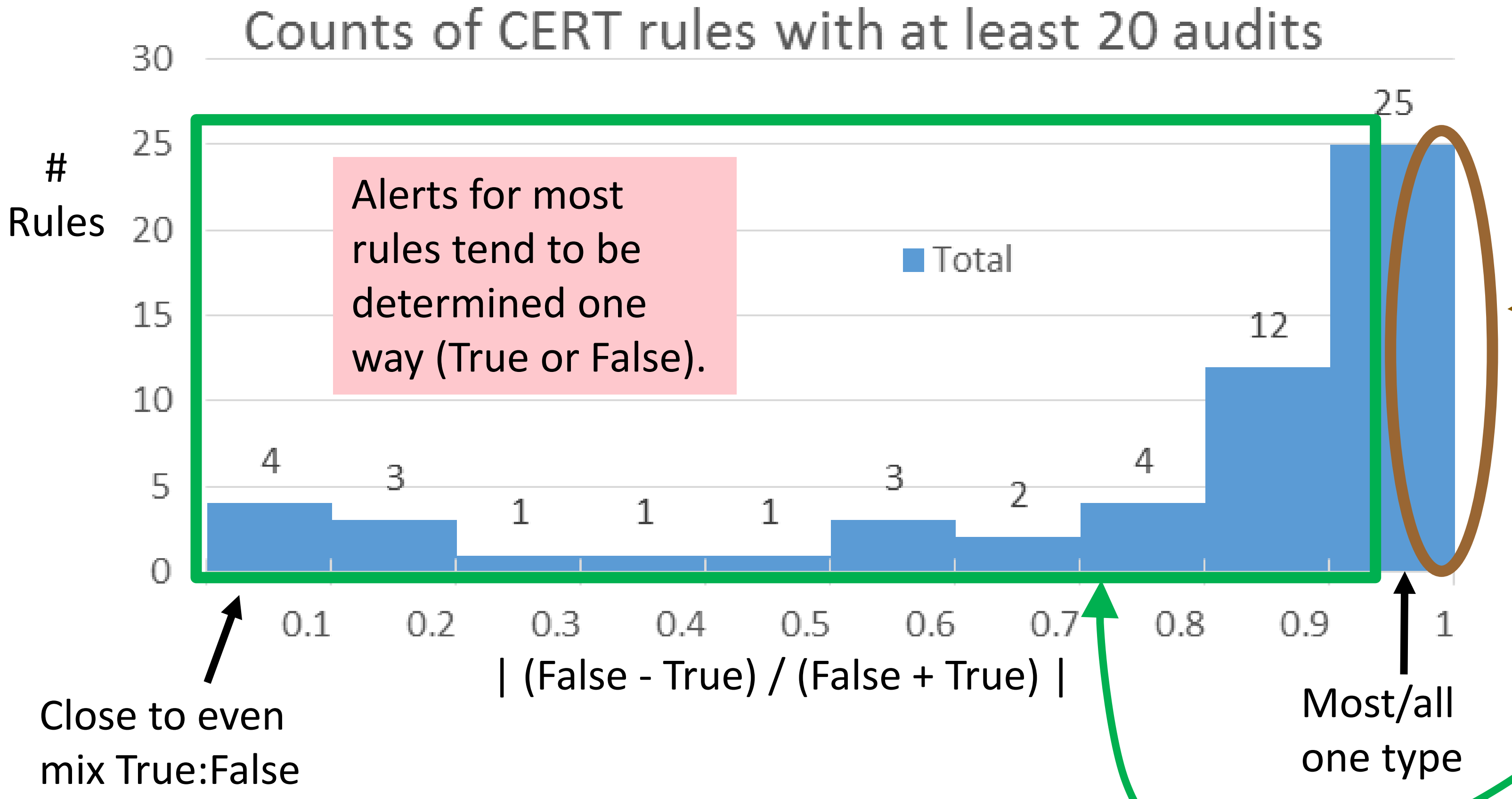
Transitionable Results: Fusion and Analysis

Fuse alerts and added analysis to prep data for classifiers

- SQLite multi-table file converted to flat .csv file
  - Flat file useful for classifier tools
- Alerts fused for same [rule, line number, file] tuple
- Add features:
  - Alerts per file
  - Alerts per function
  - Depth of file in project
  - Split filepath, so partially-shared filepaths can be used as feature
- Scripts that do this can be transitioned to DoD and others
  - Use directly on enhanced-SCALe databases
  - Modifiable for other database formats

CERT-Audited Data

56 CERT coding rules with 20 or more audits



<u>288 Classifiers Developed</u>

- 15 featureless classifiers (20 or more audits, 100% True or False)
- 201 classifiers for 11 CERT rules with mixed True/False
  - o True/False ratio & count combination insufficient for classifiers, for some rules
- 72 all-rules classifiers (rule name used as <u>feature</u>)
  - • 44 per-language classifiers

Classifier Results on CERT-Audited Data

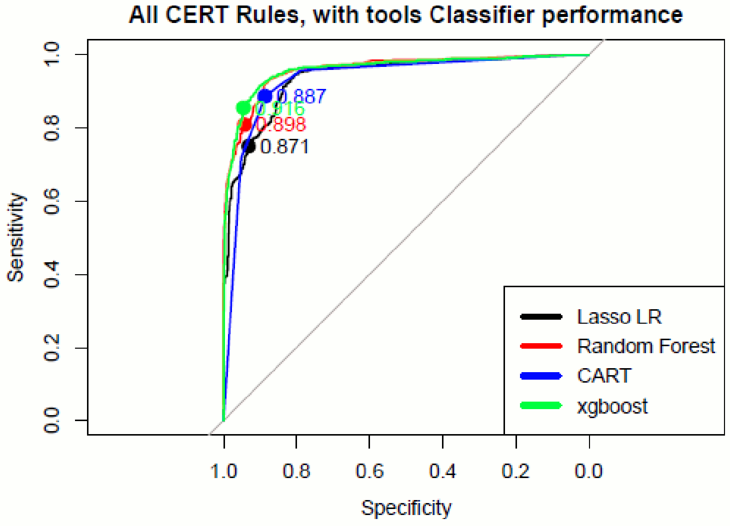
Built 2 types of classifiers using 70% CERT-audited data

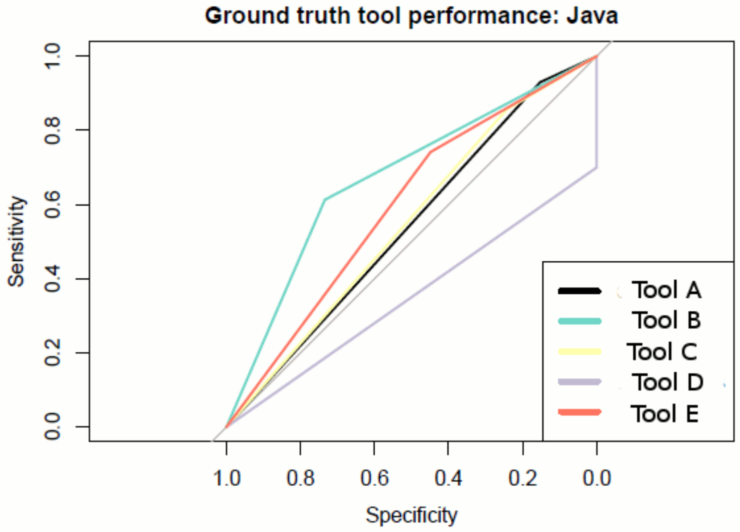
- For 10 rules (small dataset using only one rule's data)
- All-rules (large dataset with 382 rules)

Tested classifiers on remaining 30% data

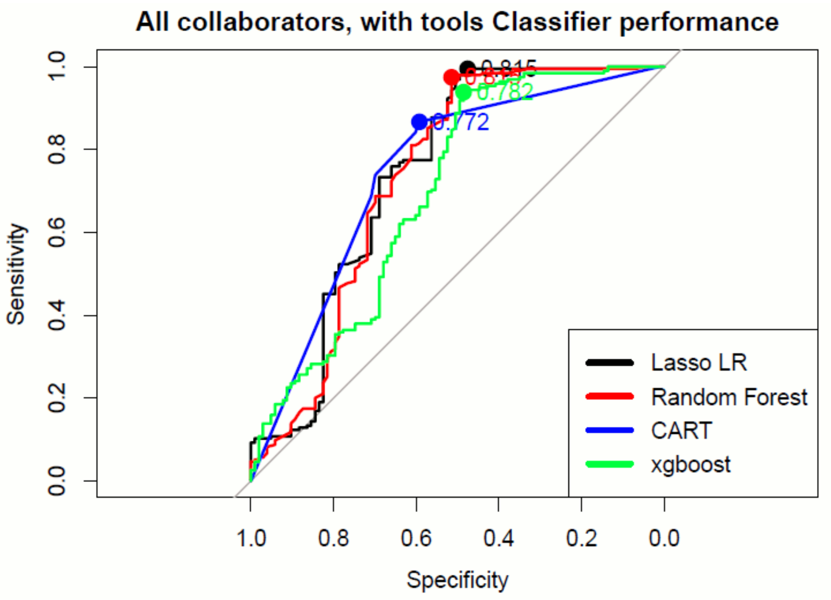
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 87%
  - Random Forest: 90%
  - CART: 89%
  - XGBoost: 92%

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	95%	96%	96%	95%
EXP01-J	68%	83%	89%	87%
OBJ03-J	73%	86%	86%	83%
FIO04-J*	75%	75%	83%	71%
EXP33-C*	90%	100%	90%	90%
EXP34-C*	74%	84%	87%	81%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	99%	99%	97%	97%
IDS00-J*	97%	94%	94%	88%
ERR01-J*	100%	100%	100%	100%





Classifier Results on Pooled Collaborator Data



All-rules

Classifier accuracy at best cut point, with tools:

- Lasso Logistic Regression: 82%
- Random Forest: 82%
- CART: 77%
- XGBoost: 78%

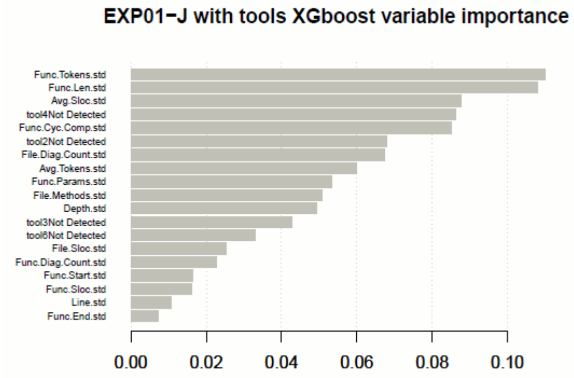
Per-rule:

Rule ID	Lasso LR	Random Forest	CART	XGBoost
EXP33-C*	71%	71%	71%	80%
EXP34-C*	87%	87%	82%	90%
FIO04-J*	85%	85%	85%	90%
IDS00-J*	97%	97%	97%	97%
INT31-C*	100%	100%	63%	100%

- Build classifiers using 100% of CERT-audited data for that rule
- Test on pooled collaborator data for that rule

No audited alerts map to 'featureless classifier' rules

* Small quantity of data, results suspect



Variable importance analysis done for each classifier

Classifier Results: Pooled Data Including CERT-Audited

Classifier made from all data, pooled:

- All-rules classifier accuracy:
  - Lasso Logistic Regression: 88%
  - Random Forest: 91%
  - CART: 89%
  - XGBoost: 91%

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	98%	97%	98%	97%
EXP01-J	74%	74%	81%	74%
OBJ03-J	73%	86%	86%	83%
FIO04-J*	80%	80%	90%	80%
EXP33-C*	83%	87%	83%	83%
EXP34-C*	67%	72%	79%	72%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	99%	100%	100%	100%
IDS00-J*	96%	96%	96%	96%
ERR01-J*	100%	100%	100%	100%
ERR09-J*	100%	88%	88%	88%

Built classifiers using 70% data

- All-rules (rules as feature)
- For 11 rules

Tested classifiers on remaining 30% data

* Small quantity of data, results suspect

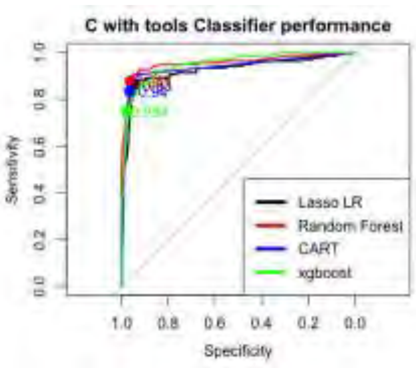
Classifier made only from CERT-audited data:

- All-rules classifier accuracy:
  - Lasso Logistic Regression: 87%
  - Random Forest: 90%
  - CART: 89%
  - XGBoost: 92%

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	95%	96%	96%	95%
EXP01-J	68%	83%	89%	87%
OBJ03-J	73%	86%	86%	83%
FIO04-J*	75%	75%	83%	71%
EXP33-C*	90%	100%	90%	90%
EXP34-C*	74%	84%	87%	81%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	99%	99%	97%	97%
IDS00-J*	97%	94%	94%	88%
ERR01-J*	100%	100%	100%	100%

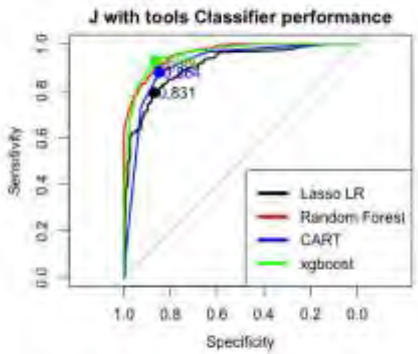
Classifier Results: Per-Language, Fully Pooled Data

All Java data, pooled:



- All-Java-rules classifier accuracy:
  - Lasso Logistic Regression: 83%
  - Random Forest: 88%
  - CART: 86%
  - XGBoost: 90%

All C data, pooled:



- All-C-rules classifier accuracy:
  - Lasso Logistic Regression: 93%
  - Random Forest: 95%
  - CART: 94%
  - XGBoost: 93%

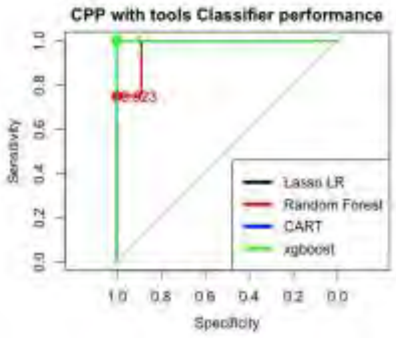
Built classifiers using 70% of data for single language

- All-rules (rules as feature)

Tested classifiers on remaining 30% data

Too little Perl data to create classifiers

All C++ data, pooled:



- All-C++-rules classifier accuracy:
  - Lasso Logistic Regression: 92%*
  - Random Forest: 92%*
  - CART: 100%*
  - XGBoost: 100%*

* C++ classifiers suspect (little data, ROC graph)

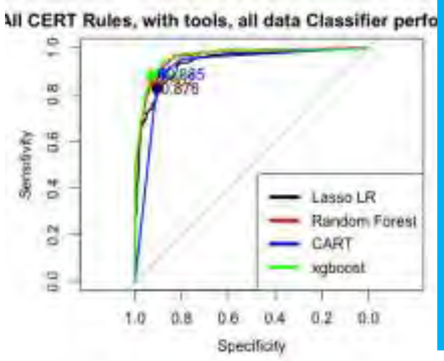
Classifier Results: No Function-Features

12% more data, not requiring function features

- Built classifiers using 70% of data with no function features
- Tested on remaining 30% data

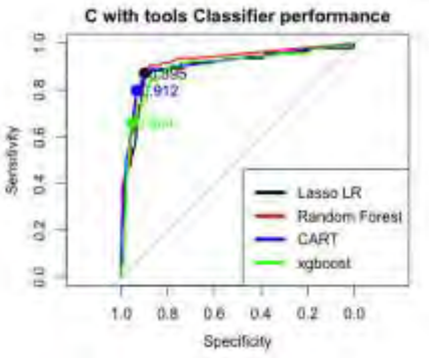
Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	97%	97%	97%	97%
EXP01-J	71%	75%	81%	77%
OBJ03-J	65%	86%	84%	84%
FIO04-J*	80%	80%	83%	80%
EXP33-C*	66%	80%	84%	80%
EXP34-C*	70%	72%	77%	72%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	98%	100%	100%	100%
IDS00-J*	96%	98%	96%	93%
ERR01-J*	100%	100%	100%	100%
STR31-C	93%	97%	93%	93%

All data pooled:



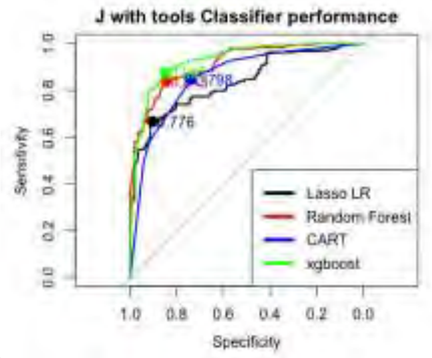
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 88%
  - Random Forest: 90%
  - CART: 88%
  - XGBoost: 91%

C language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 90%
  - Random Forest: 91%
  - CART: 91%
  - XGBoost: 90%

Java language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 78%
  - Random Forest: 84%
  - CART: 80%
  - XGBoost: 86%

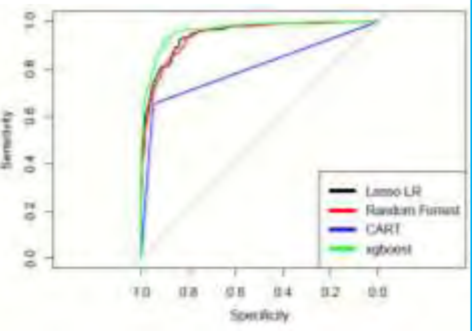
Classifier Results: Drop-Columns

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	97%	97%	98%	97%
EXP01-J	71%	75%	76%	76%
OBJ03-J	65%	86%	84%	84%
FIO04-J*	76%	86%	76%	83%
EXP33-C*	70%	68%	84%	65%
EXP34-C*	74%	84%	87%	81%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	99%	99%	97%	96%
IDS00-J*	98%	90%	95%	95%
ERR01-J*	100%	98%	98%	98%
STR31-C	100%	98%	98%	98%
ERR09-J*	100%	100%	100%	100%

All-CERT data, dropped features missing data

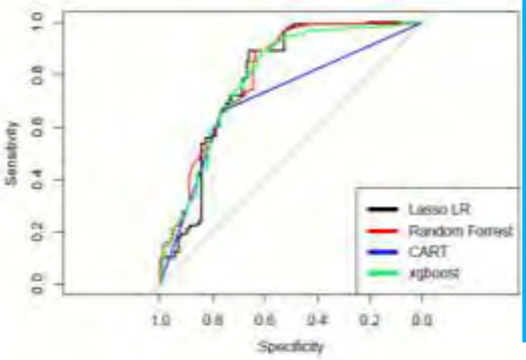
- Built classifiers using 70% of data and tested on other 30%
- Built classifiers using 100% of data and tested on pooled collaborator data

30% CERT data tested:



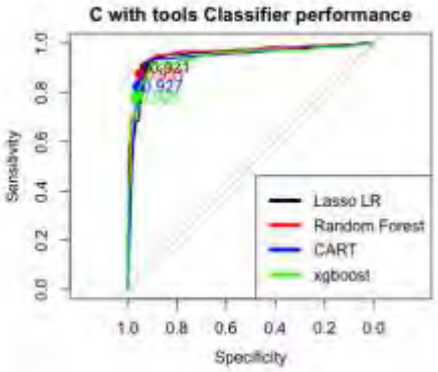
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 88%
  - Random Forest: 87%
  - CART: 85%
  - XGBoost: 91%

Pooled collaborator data tested:



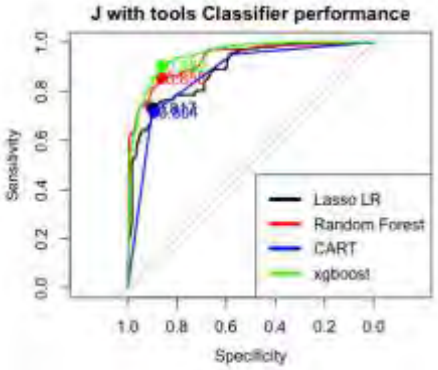
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 80%
  - Random Forest: 80%
  - CART: 70%
  - XGBoost: 79%

C language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 92%
  - Random Forest: 93%
  - CART: 93%
  - XGBoost: 92%

Java language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 82%
  - Random Forest: 86%
  - CART: 80%
  - XGBoost: 88%

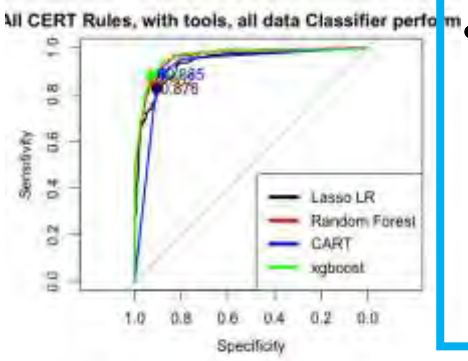
Classifier Results: Drop-Columns

Rule ID	Lasso LR	Random Forest	CART	XGBoost
INT31-C	97%	97%	97%	97%
EXP01-J	73%	69%	79%	83%
OBJ03-J	65%	86%	84%	84%
FIO04-J*	71%	77%	77%	74%
EXP33-C*	66%	80%	84%	80%
EXP34-C*	67%	72%	79%	72%
DCL36-C*	100%	100%	100%	100%
ERR08-J*	97%	98%	100%	100%
IDS00-J*	100%	98%	95%	93%
ERR01-J*	100%	100%	100%	100%
STR31-C	97%	97%	93%	93%
ERR09-J*	100%	100%	93%	100%

52% more pooled data (now with Perl), vs. function-features-required

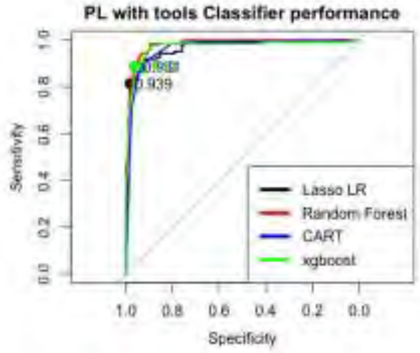
- Built classifiers using 70% of data (dropped columns if miss data)
- Tested on remaining 30% data

All data pooled:



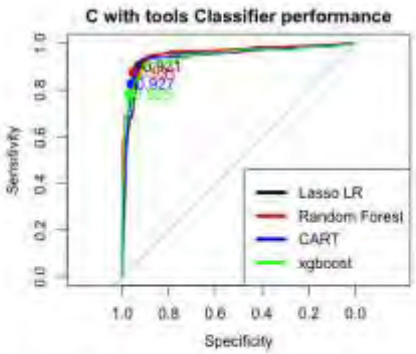
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 89%
  - Random Forest: 88%
  - CART: 86%
  - XGBoost: 90%

Perl language data pooled:



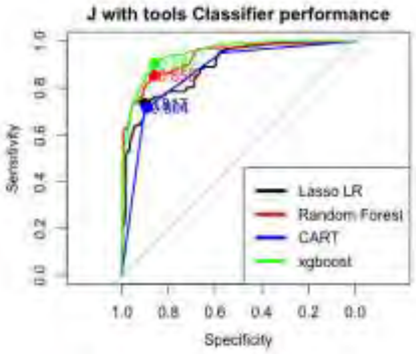
- All-rules classifier accuracy:
  - Lasso Logistic Regression: 94%
  - Random Forest: 94%
  - CART: 94%
  - XGBoost: 93%

C language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 92%
  - Random Forest: 93%
  - CART: 93%
  - XGBoost: 92%

Java language data pooled:



- All-rules classifier accuracy:
  - Lasso Logistic Regression: 82%
  - Random Forest: 86%
  - CART: 80%
  - XGBoost: 88%

Overview

Problem: The number of security-related code flaws detected by static analysis requires too much effort to triage.

Significance:

1) Code flaws and vulnerabilities remain.
2) Scarce resources are used inefficiently.

Project goal: Classification algorithm development using CERT- and collaborator-audited data to accurately estimate the probability of true and false positives, to efficiently use analyst effort and remove code flaws.