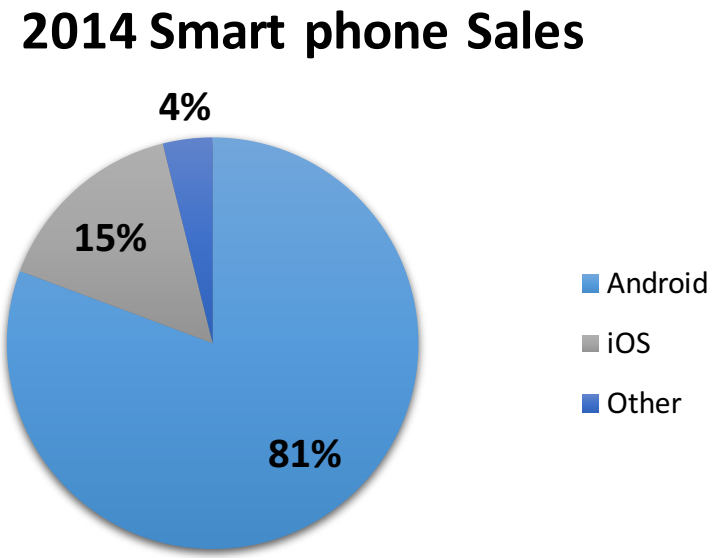
Experimental Study with Real-world Data for Android App Security Analysis using Machine Learning

Sankardas Roy, **Jordan DeLoach**, Yuping Li, Nic Herndon, Doina Caragea, Xinming Ou, Venkatesh Prasad Ranganath, Hongmin Li, Nicolais Guevara

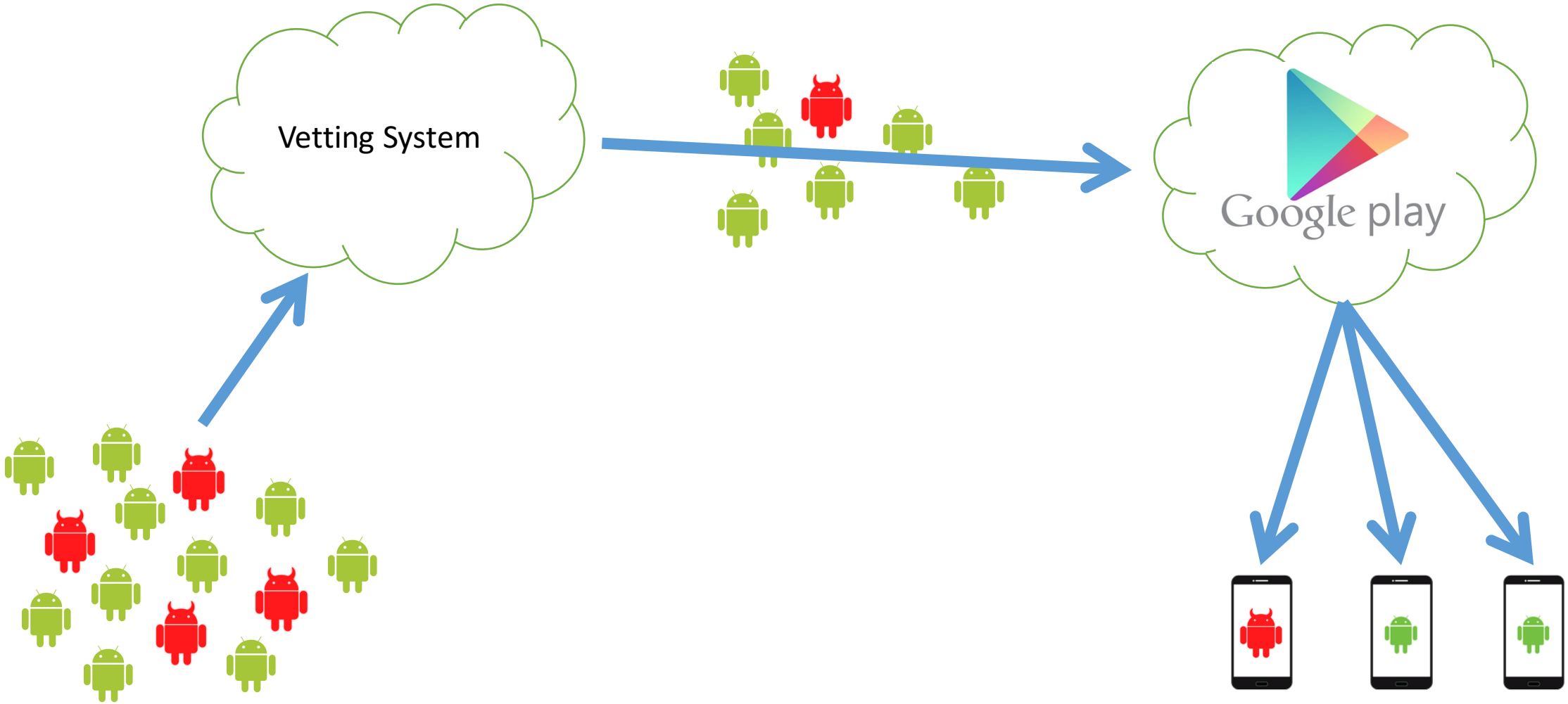
Motivations

- Android dominates market share world wide
- Smart phones wide variety of uses make them also vulnerable to a wide variety of malicious attacks
- Common Malware Behavior:
  - Leaking personal data
  - GPS tracking
  - SMS messages to premium numbers



Gartner 2015

Vetting Process



A Machine Learning Approach

- Machine learning could be an effective approach to help detect malware
- Doing machine learning is hard
- Doing *standardized* machine learning is even harder

Impacting Factors of an ML Approach

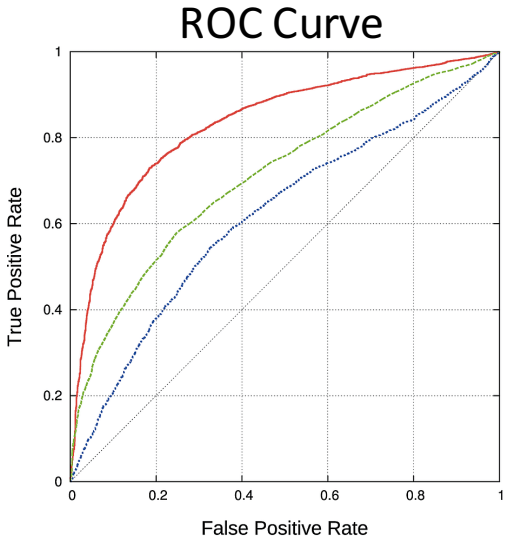
Research Questions

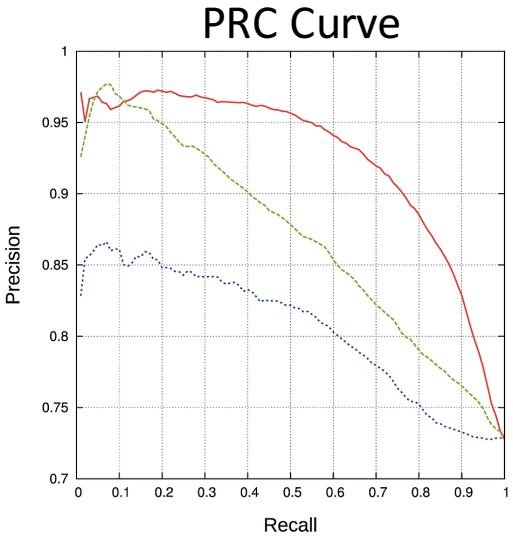
- Evaluation Strategy
  - What is the best performance metric?
- Input Data
  - Does the age of the malware in dataset mislead performance?
  - Does data imbalance affect the performance?
  - Does quality of ground truth affect the performance?
  - Does presence of adware in the dataset affect the performance?
- Algorithm Design
  - Are more features always better?

Experimental Framework

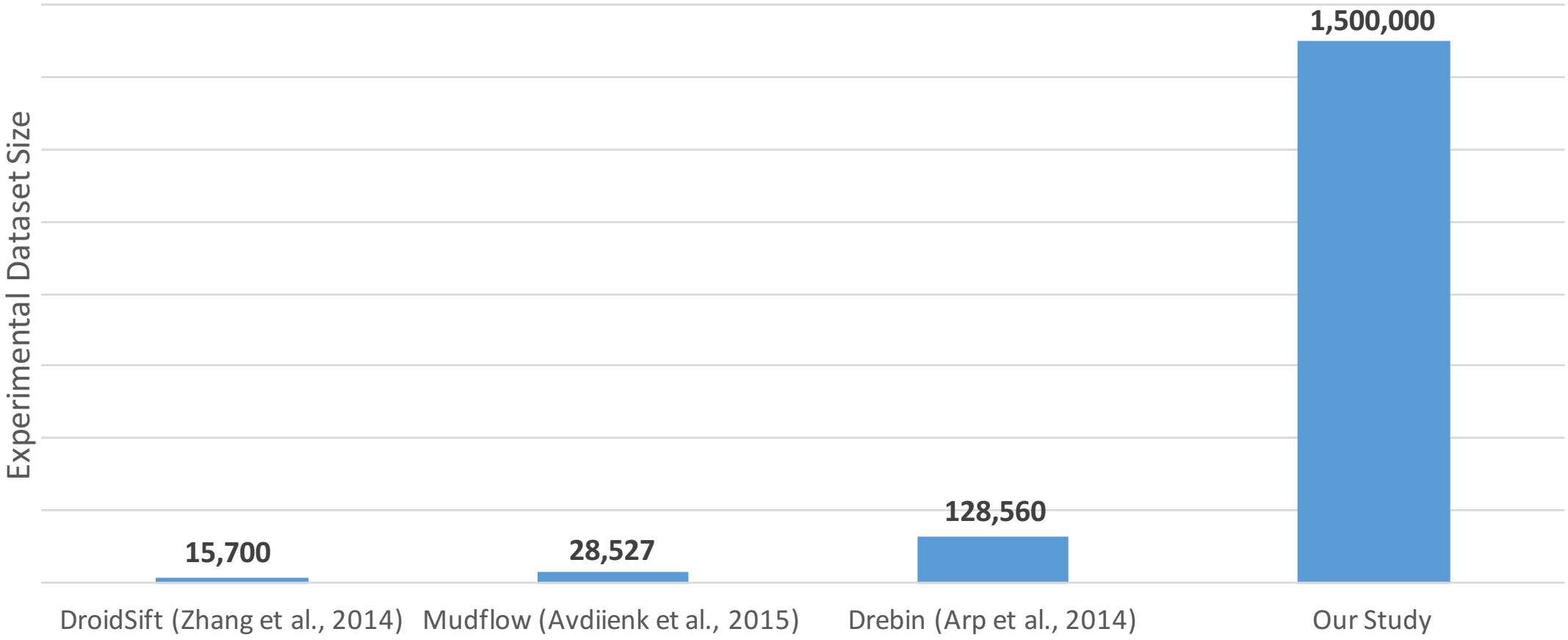
- Classifier: k Nearest Neighbor (k=1)
- Feature Set: 471 features
  - Contains permissions, intent strings, and critical APIs
- Train/Test Split: 5-fold Cross-validation
- Evaluation Metrics
  - True Positive Rate (TPR)
  - False Positive Rate (FPR)
  - Receiver Operator Characteristic Curve (auROC)
  - Precision Recall Curve (auPRC)

True Positive %	$\frac{TP}{TP+FN}$	Precision	$\frac{TP}{TP+FP}$
False Positive %	$\frac{FP}{FP+TN}$	Recall	$\frac{TP}{TP+FN}$

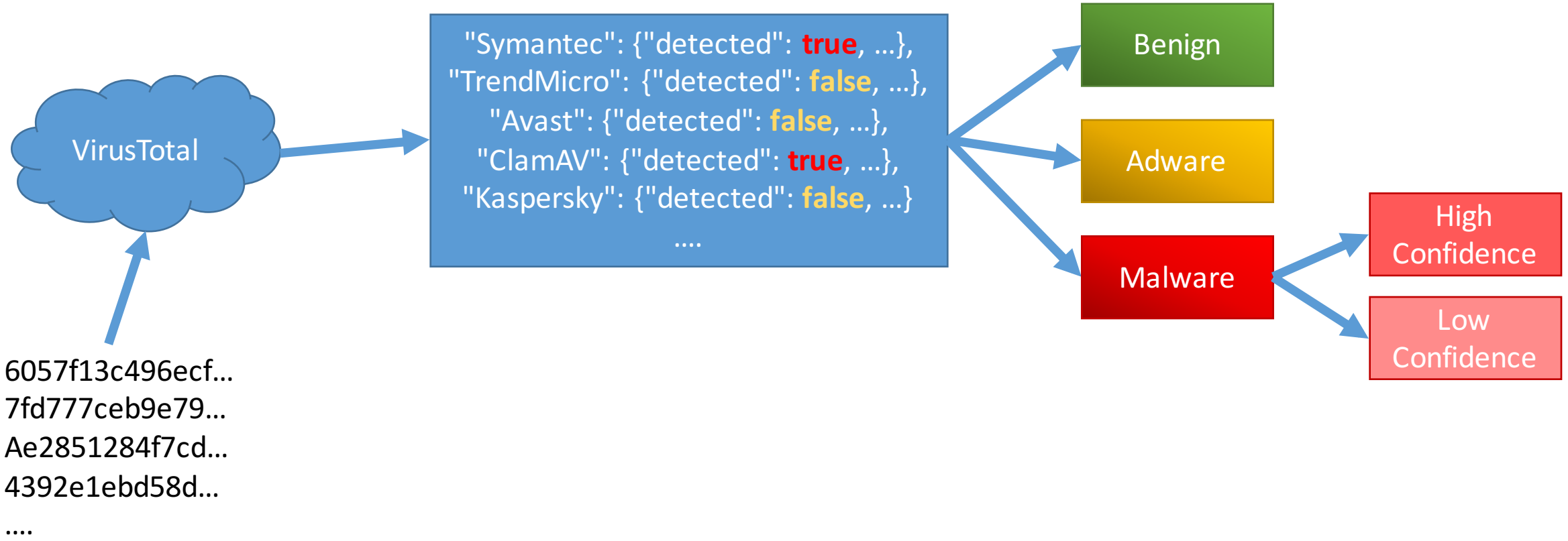




Experimental Datasets



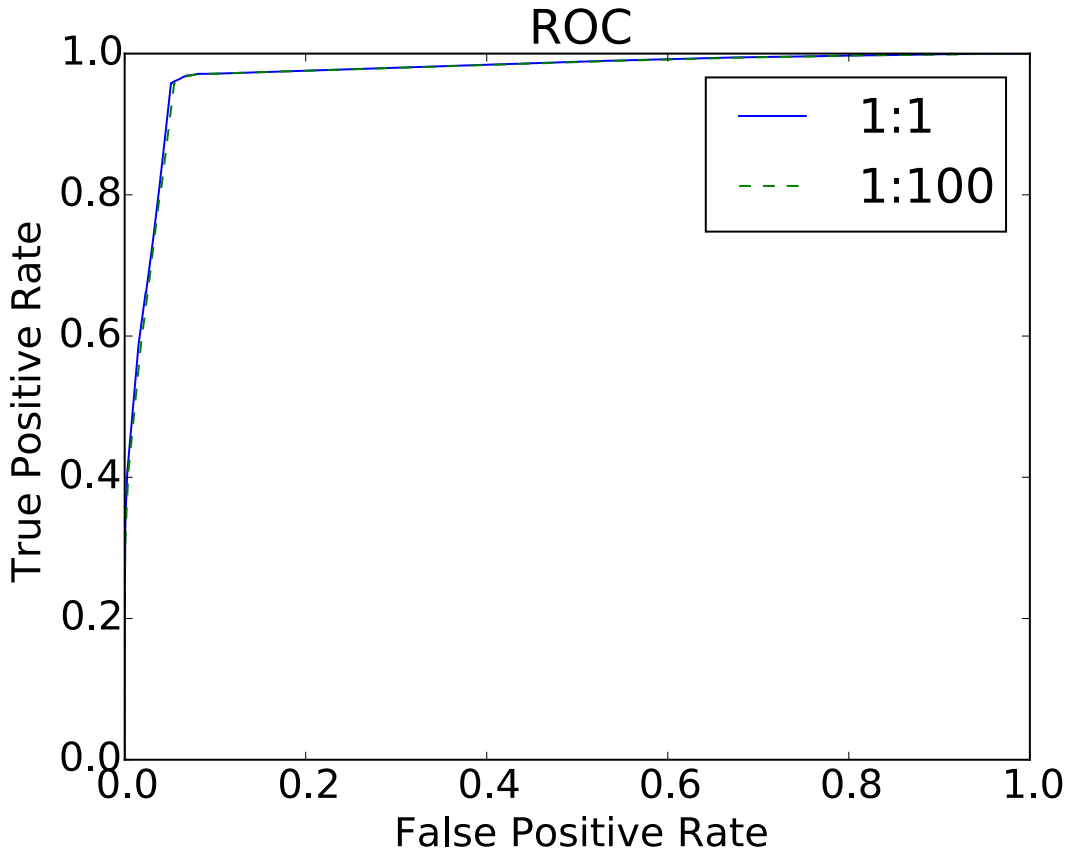
Experimental Ground Truth Preparation

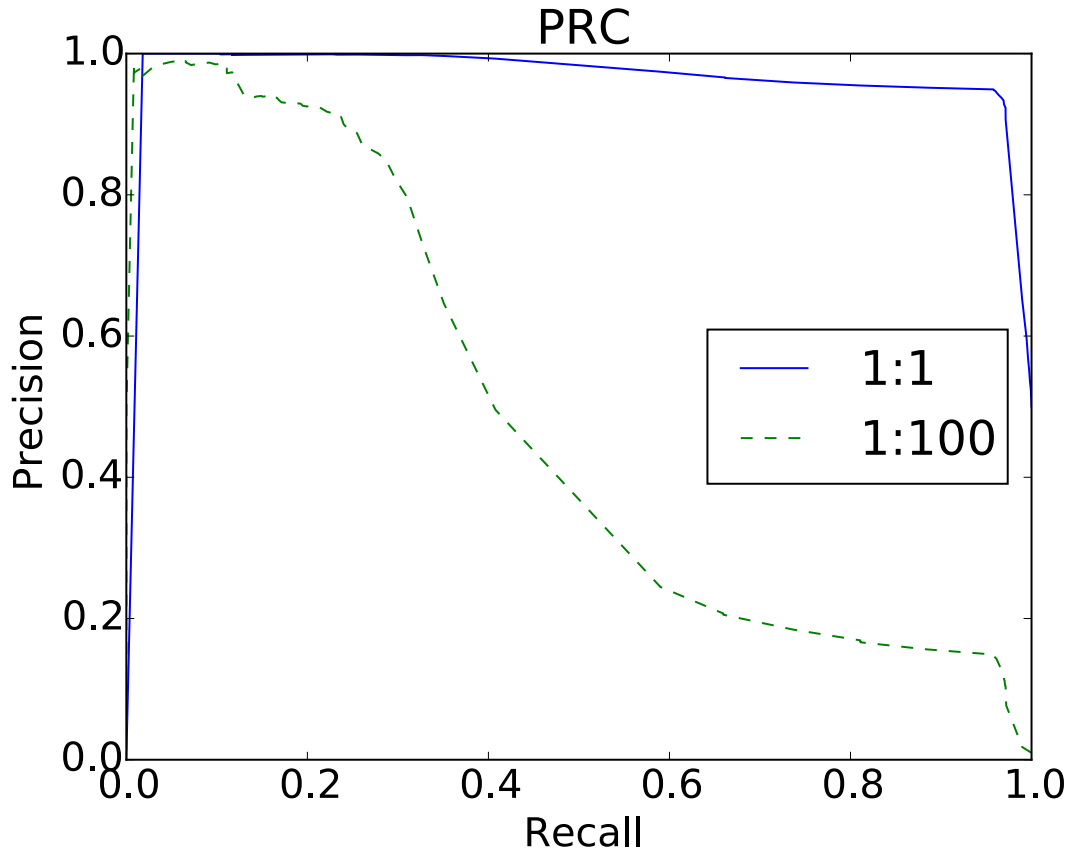


RQ1: Evaluation Strategy - ROC or PRC?

- General ML theory has found in highly unbalanced datasets, auPRC to be the best metric (*Davis et al., 2006*).
- Most common for Android ML malware solutions is auROC
- **Hypothesis**: For largely imbalanced datasets, auPRC provides better insight into classifier performance
- **Experiment:** Compare classifier performance on a 1:1 and 1:100 malware to benign ratio datasets

ROC vs PRC


ROC
True Positive Rate
False Positive Rate
1:1
1:100
0.0
0.2
0.4
0.6
0.8
1.0


PRC
Precision
Recall
1:1
1:100
0.0
0.2
0.4
0.6
0.8
1.0

RQ1: Evaluation Strategy - ROC or PRC?

- **Observations:** While TPR, FPR, and auROC remain nearly constant, we see a dramatic drop in auPRC.
- **Conclusions:** auPRC is a better metric for comparing largely imbalanced datasets in the Android space

RQ2: Dataset Age

- The Genome dataset was collected by NCSU with apps from 2010/2011.
- **Hypothesis:** Using Genome, or other dated datasets, leads to misleading results.
  - The classifier learns *old versus new* as opposed to *malicious versus benign*.

RQ2: Experimental Setup

Genome Trial

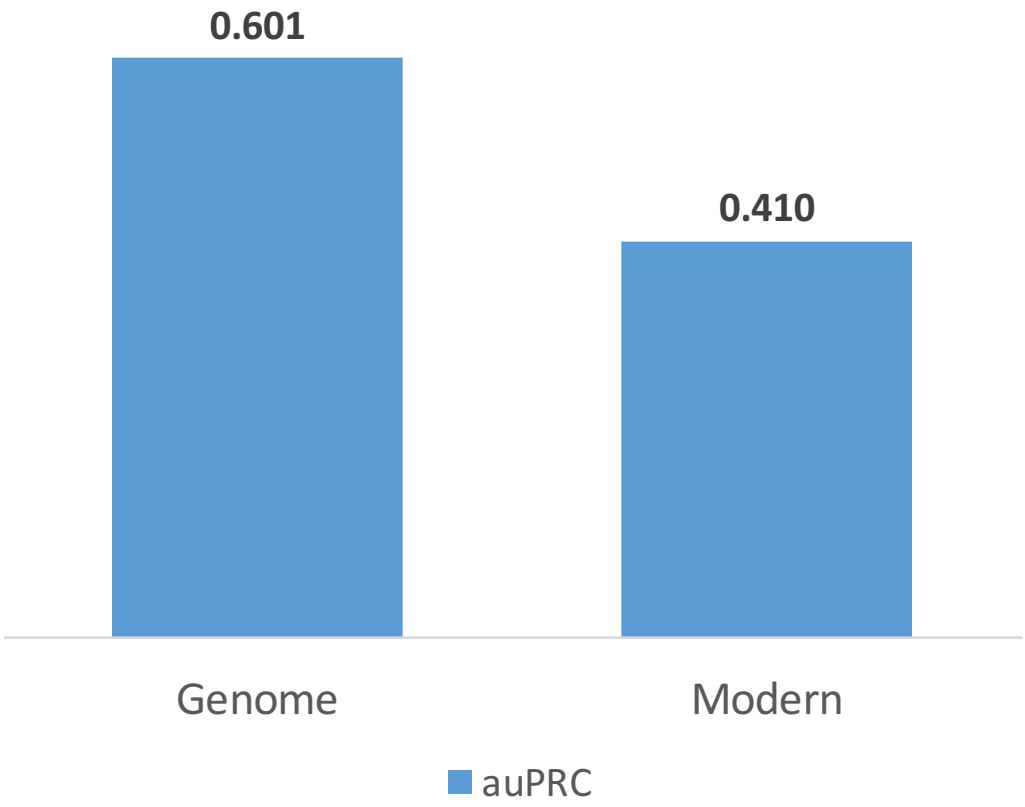
- **Malicious:** 1,260 Genome Apps
- **Benign:** 63K benign apps from the Play Store

Modern Malware Trial

- **Malicious:** 1,260 Modern Malware from Arbor/VirusShare
- **Benign:** Same 63K benign apps

RQ2: Dataset Age - Conclusions

- **Observations:** We see a dramatic drop in auPRC.
- **Conclusion**: In reality, modern malware is much more diverse and difficult to learn from.

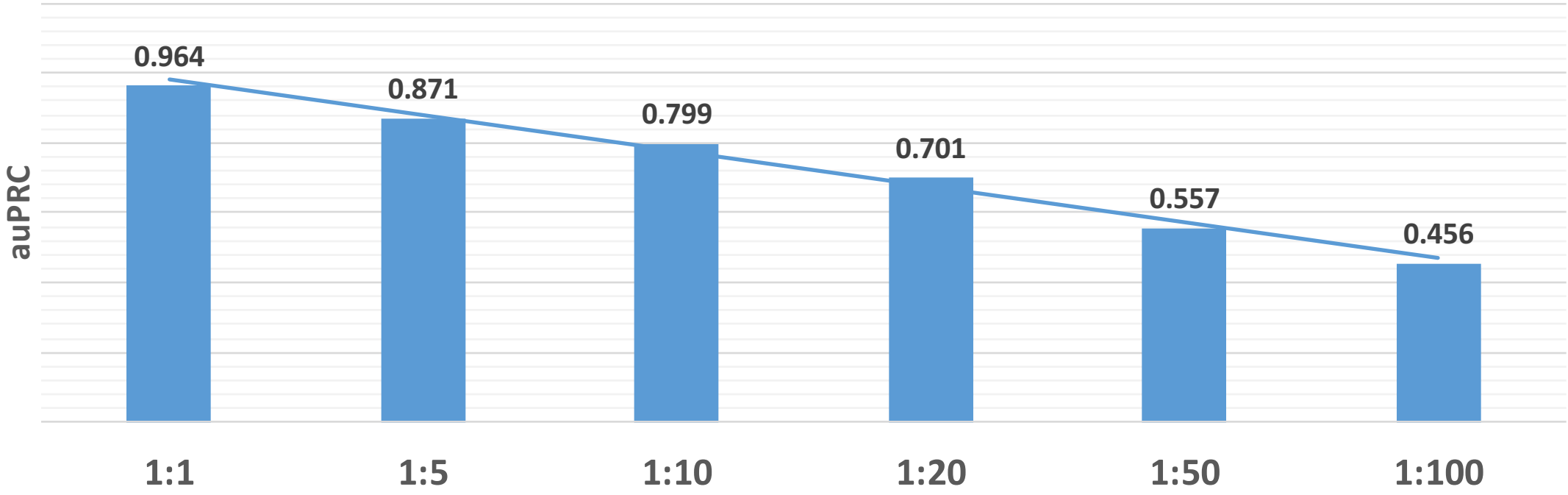


RQ3: Dataset Imbalance

- Most apps submitted to a vetting system are benign
  - Realistic ratio is around 1:100
  - Many peer works use ratios varying from 1:4 to 1:22.
- **Hypothesis:** As the ratio of imbalance between malicious and benign apps gets larger, the problem becomes more difficult.

RQ3: Dataset Imbalance

- **Observations:** auPRC substantially declines as realistic imbalances are approached.
- **Conclusion:** In the Android ML space, consideration of class imbalance is critical for crafting real-world solutions.

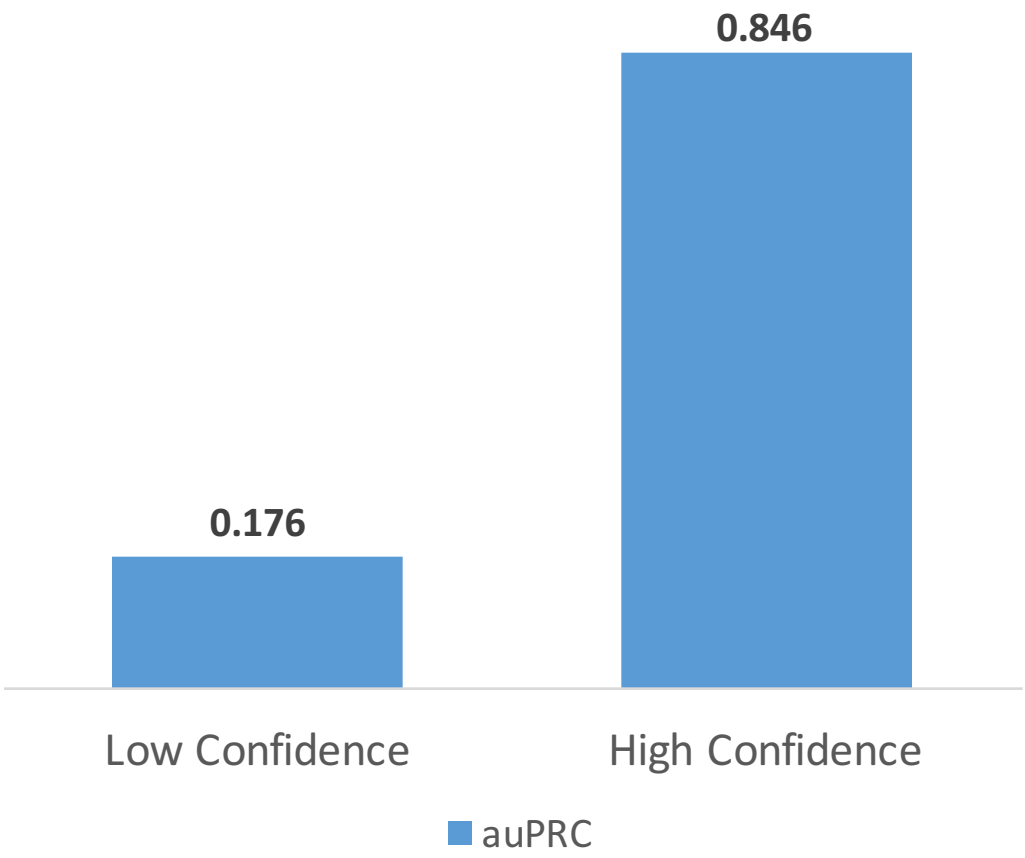


RQ4: Ground Truth Labeling Quality

- Classifier performance directly derives from accuracy of ground truth
- The more accurate the ground truth, the more accurate the classifier
- **Hypothesis**: If we train our classifier off of higher confidence malware, we will have higher accuracy.

RQ4: Ground Truth Labeling Quality

- **Observations:** When training over low confidence malware, true positives decreased, but false positives skyrocketed.
- **Conclusions:** Training with high confidence malware is critical to discerning malicious patterns.

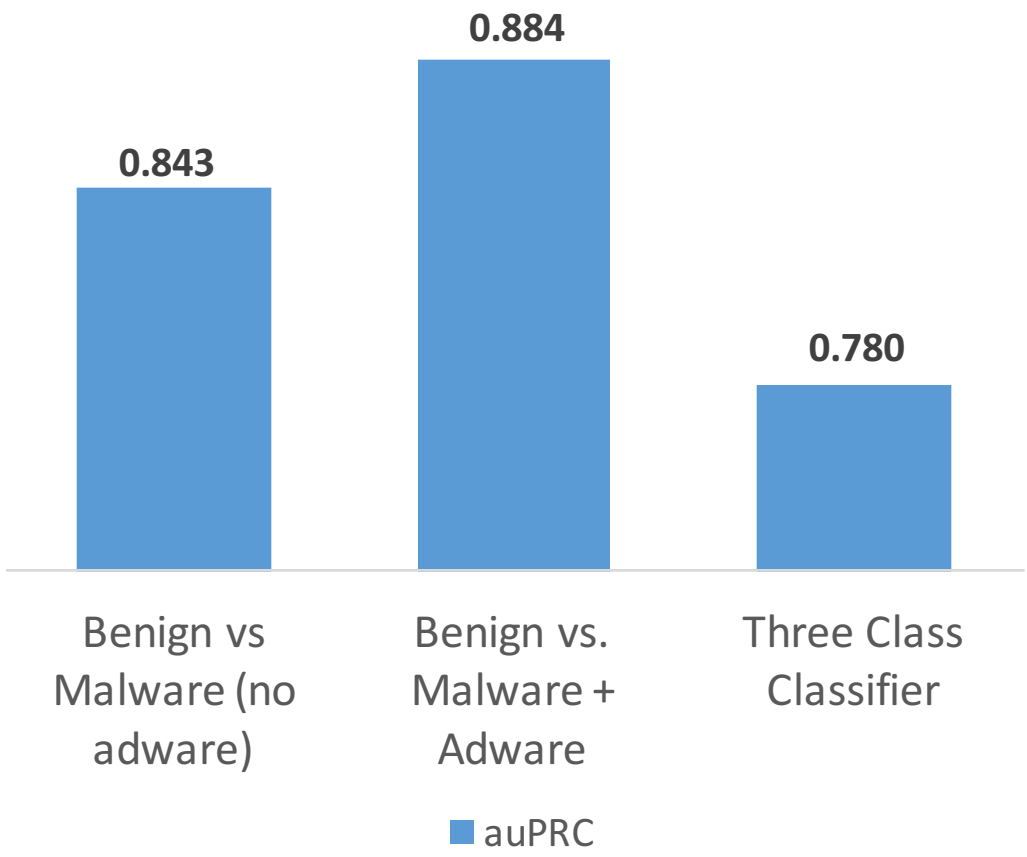


RQ5: Adware - Motivations

- Adware are a special category of apps that exist in some grey area between malicious and benign apps.
- We seek to understand where adware belong in the scheme when attempting to classify apps.
- **Hypothesis**: The inclusion of adware will decrease accuracy as the problem becomes more complicated with multiple classes.

RQ5: Adware - Conclusions

- **Observations:** We see that performance is worst when we attempt to distinguish malware from adware in the three class classifier.
- **Conclusions:** Adware noticeably impacts performance. As adware are prevalent in the real world, they cannot be combined nor neglected.

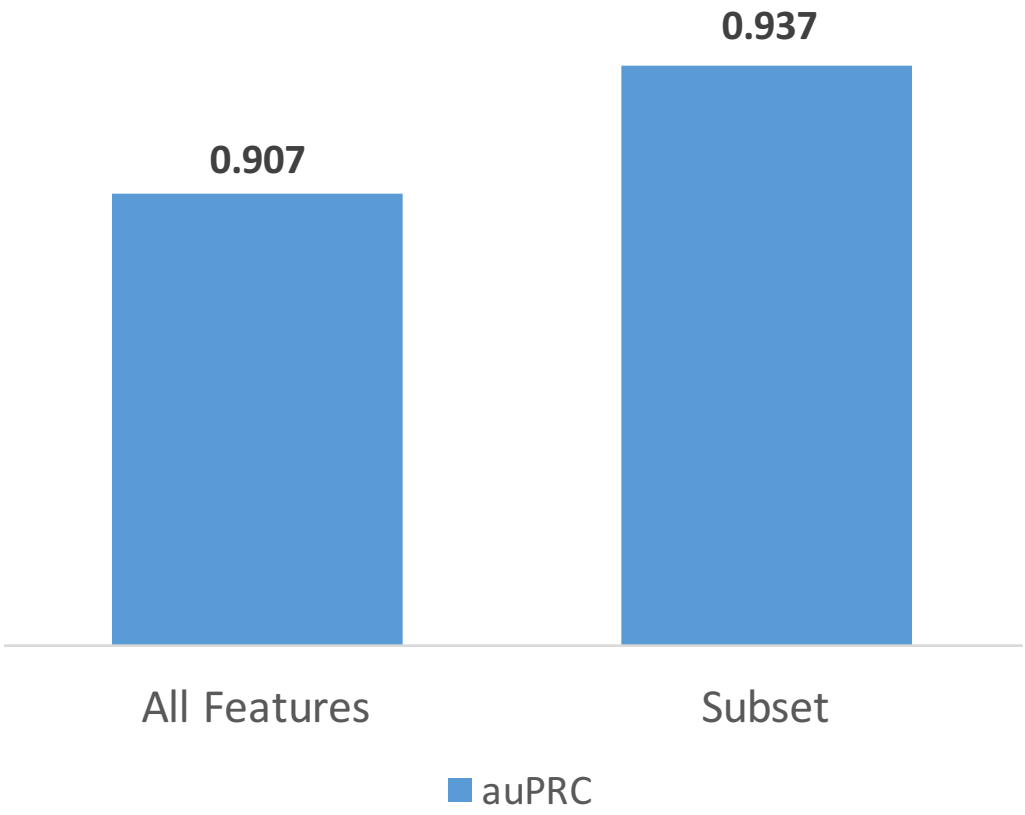


RQ6: Feature Selection - Motivations

- Many approaches use a large amount of features as part of their ML algorithm
  - Drebin uses a dataset-specific-sized feature set that numbered over a million from 180K apps
  - DroidSIFT uses 1,183 API dependency graph-based features
- **Hypothesis**: More features does not always mean better. Reducing can improve computational performance as well as increase classifier performance

RQ6: Feature Selection - DroidSIFT

- Used DroidSIFT's rich, graph-based feature vectors
- Selected a subset of 192 features using mutual information from the base 1,183 features



RQ6: Feature Selection - Drebin

	All Features (1.37 million)	Subset (2,246)
TPR	98.2%	98.2%
FPR	**1.5%**	**0.1%**
auROC	0.911	0.982
auPRC	**0.990**	**0.994**

RQ6: Feature Selection - Conclusions

- More doesn’t always mean better
  - Not all features are discriminative
- Computational performance gains are made by reducing amount of features extracted by orders of magnitude
  - Pays off both when extracting features and when training the model
- Classifier performance gains are made by reducing the noise and confounding features in the dataset

Contributions

- Identified 6 Research Questions and derived best practices from these
  - RQ1 - auPRC is a better measurement of classifier accuracy
  - RQ2 - Old data misleads performance. Genome should be used with care.
  - RQ3 - Data imbalance affects performance
  - RQ4 - Ground Truth is vital to accurate classifier performance
  - RQ5 - Adware decreases performance
  - RQ6 - More doesn't mean better for features
- Proposed an experimental framework for Android machine learning experimentation based upon our findings

Jordan DeLoach: jdeloach@ksu.edu

Backup Slides

RQ4: Full Stats

	High Confidence Malware	Low Confidence Malware
TPR	97.8%	65.0%
FPR	5.1%	28.7%
auPRC	0.846	0.176

RQ5: Full Stats

	Benign vs Malware (no adware)	Benign vs. Malware + Adware	Three Class Classifier
True Positive Rate	79.6%	80.6%	76.2%
False Positive Rate	18.8%	15.7%	11.9%
auPRC	0.843	0.884	0.780

RQ6: DroidSIFT

	All Features (1183)	Subset (192)
TPR	**90.6%**	**95.6%**
FPR	18.8%	22.1%
auROC	0.932	0.955
auPRC	**0.907**	**0.937**

RQ6: Drebin

	All Features (1.37 million)	Subset (2,246)
TPR	98.2%	98.2%
FPR	**1.5%**	**0.1%**
auROC	0.911	0.982
auPRC	0.990	0.994

PRC vs. ROC – FPR vs. Precision

- According to the metric definition of FPR, a large change in the number of false positives can only lead to a small change in FPR which is used in ROC analysis.
- However, since precision compares false positives against true positive, PRC will be able to capture the effect of the large number of negative examples on the algorithm's performance.