Sensor-Based Repackaged Malware Detection

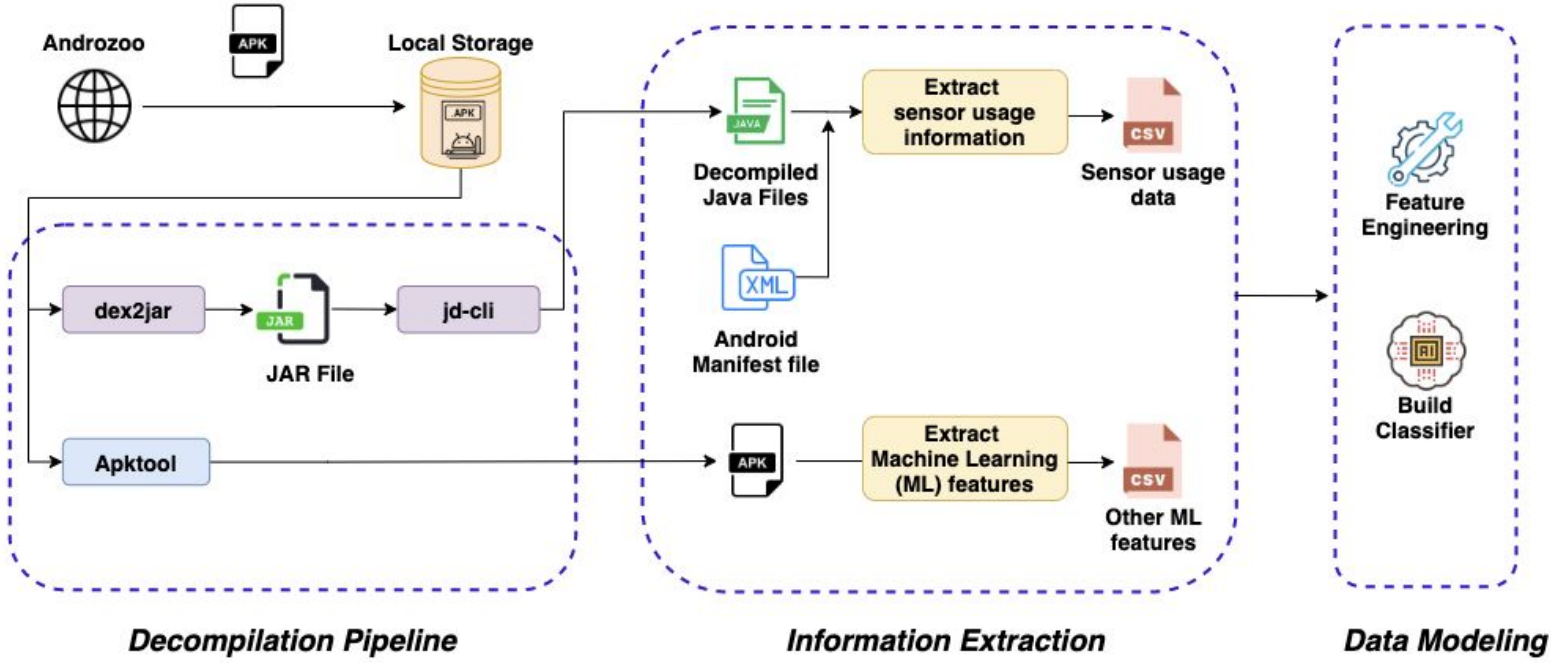
Authors: Boyu Liu, Duanyue Yun, Huiyu Song, Xin Guo, Xiao Ji
Mentors: Shirish Singh, Gail Kaiser

Data Science Institute

Introduction

- Repackaged malware poses a real threat to the health of the Android ecosystem.
- There is evidence that zero-permission sensors in phones are used by malware to perform various activities.
- We are interested in whether sensor usage information of an app will help with repackaged malware detection.

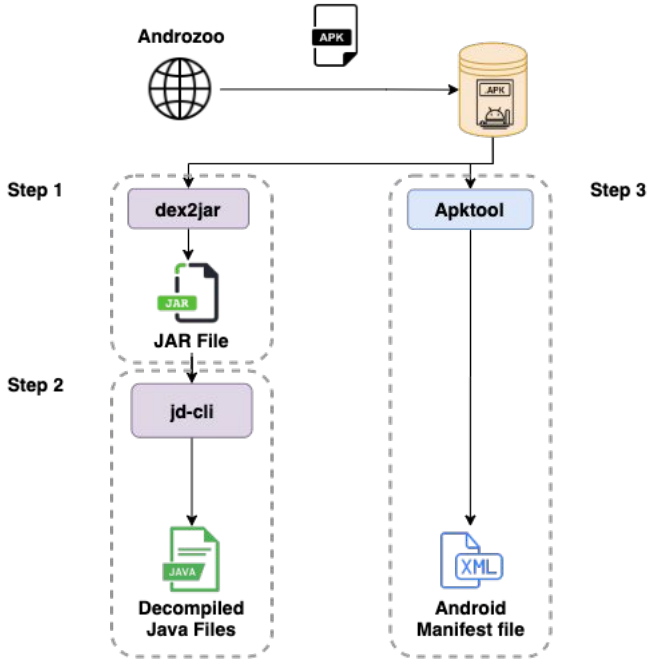
Modeling Pipeline



Decompilation

- Download Android Applications (APKs) from AndroZoo:
  - Dataset: 15,297 pairs
    (2,776 original & 15,297 repackaged apps)

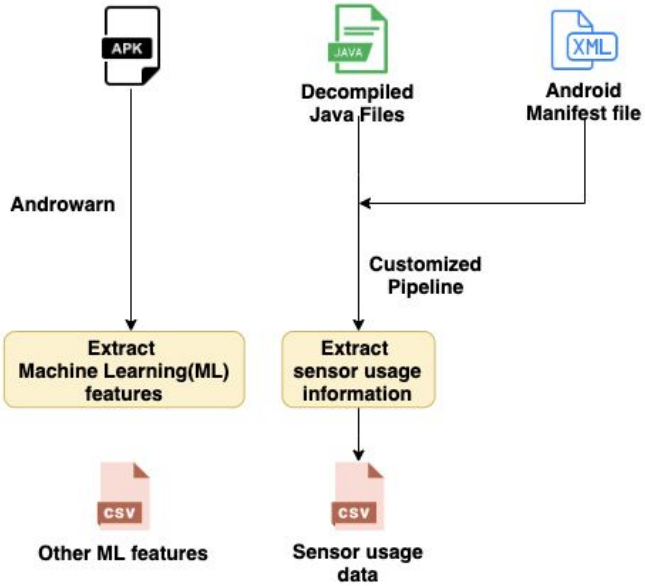
- Decompile APKs into:
  - Java source code
    - Use dex2jar & jd-cli to convert classes.dex
  - AndroidManifest.xml
    - Use Apktool to extract



Information Extraction

- Build search algorithm to find sensor usage information.
- Use **VirusTotal** to separate data .

- Drop ambiguous apps:
  - Original apps: VirusTotal report > 0
  - Repackaged apps: VirusTotal report < 5
- Keep benign apps and malwares:
  - Original apps: VirusTotal report =0
  - Repackaged apps: VirusTotal report >= 5



Statistical Testing

- From the graph, the percentage of malware using sensors is higher than benign apps, but is the difference *significant*?
- Yes! - the statistical test shows that the difference is significant.

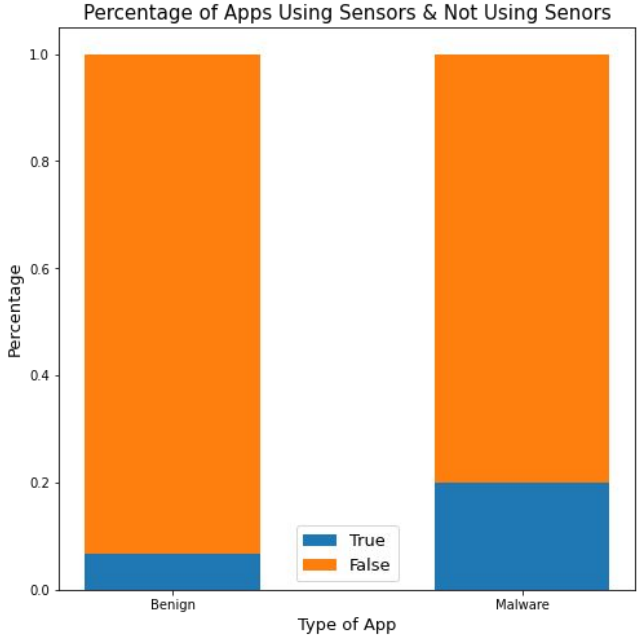
p1/p2: proportion of malicious/benign apps using at least one sensor

H0: p1=p2 H1: p1>p2

The test statistic is calculated as follows:

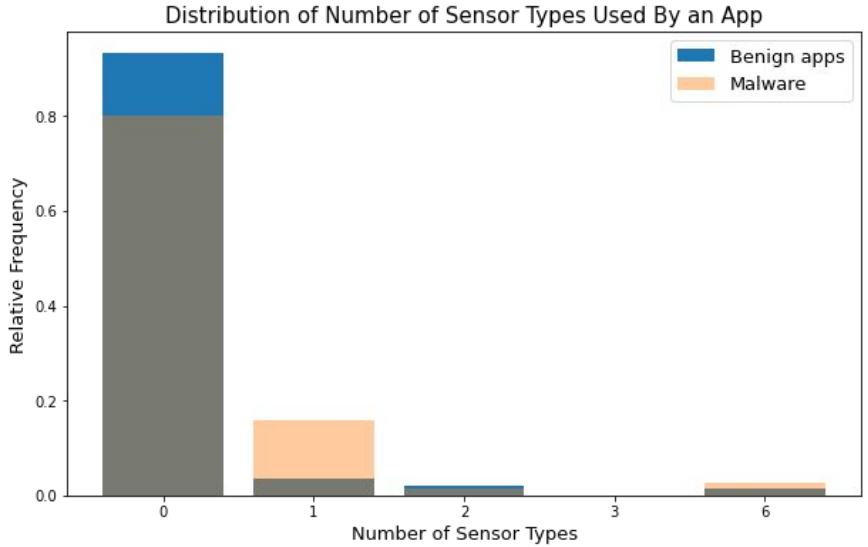
$$z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} = 13.947$$

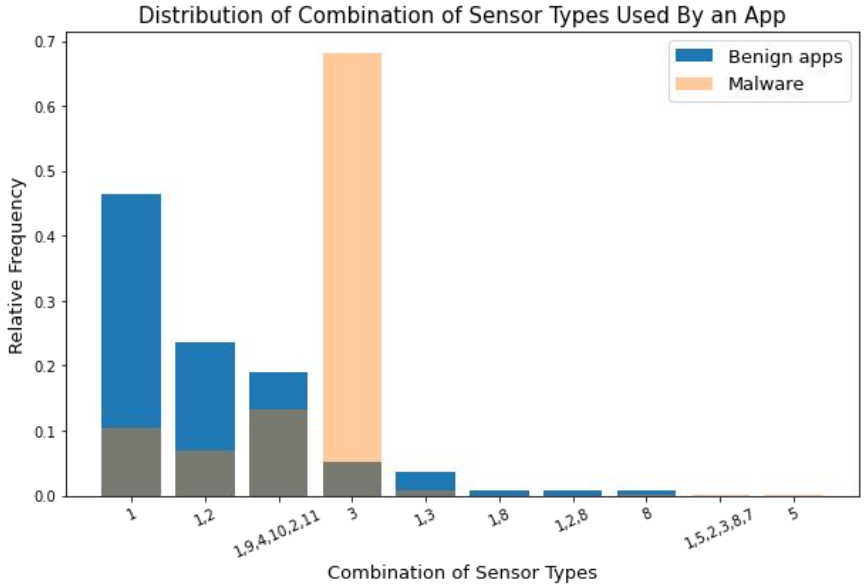
Reject H0!



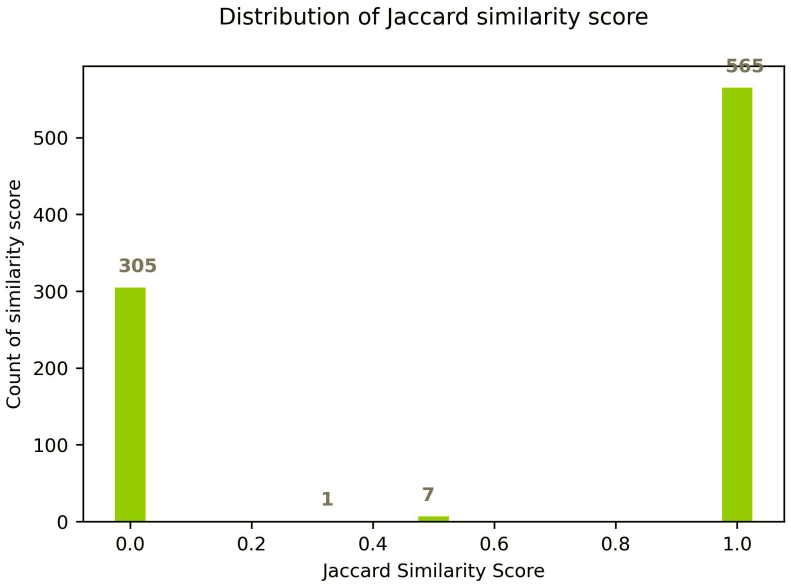
Data Visualization

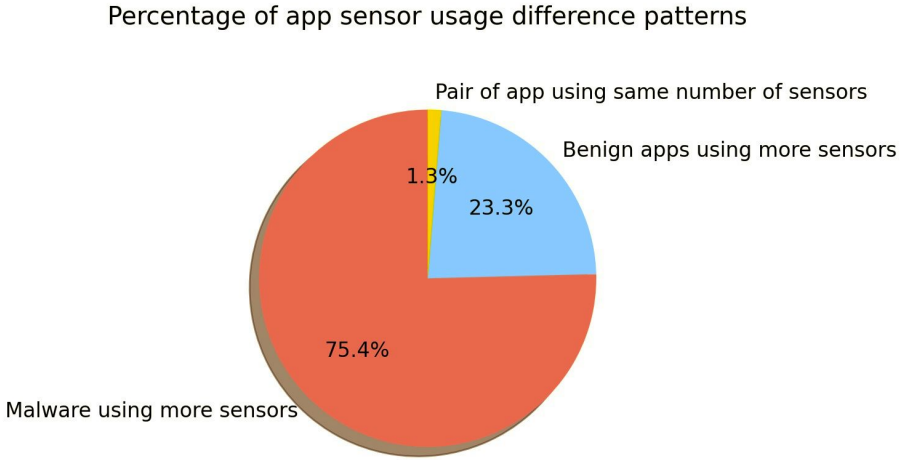
Malware and benign apps have different sensor usage patterns!





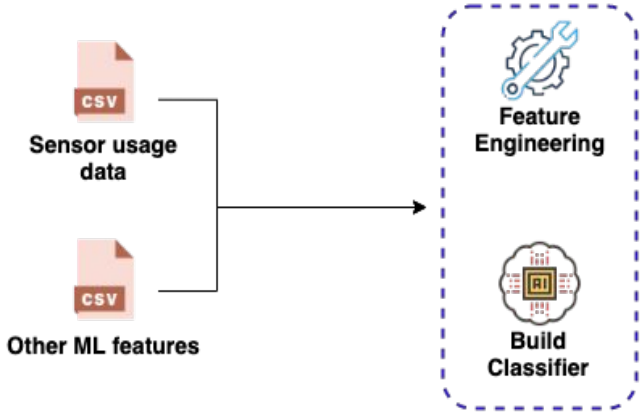
Data Visualization





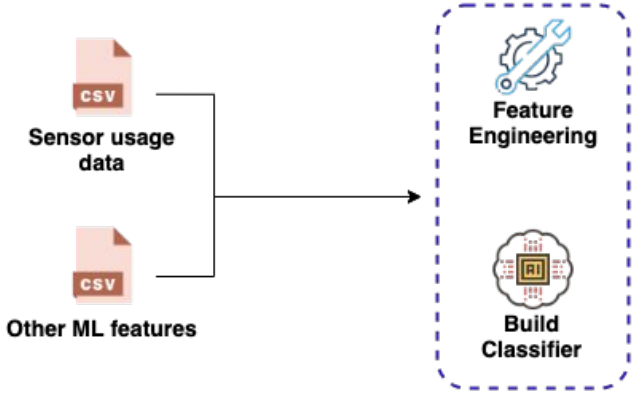
Data Modeling

- Features: drop zero variance features
  - 10 sensor-related features
  - 323 other model interpretable features
- Evaluation metric:
  - Imbalanced dataset (80% are malware)
  - Use balanced accuracy
- Models experimented includes *Logistic Regression* (LR), *K Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), *XGBoost* (XGB), *Deep Neural Network* (DNN)



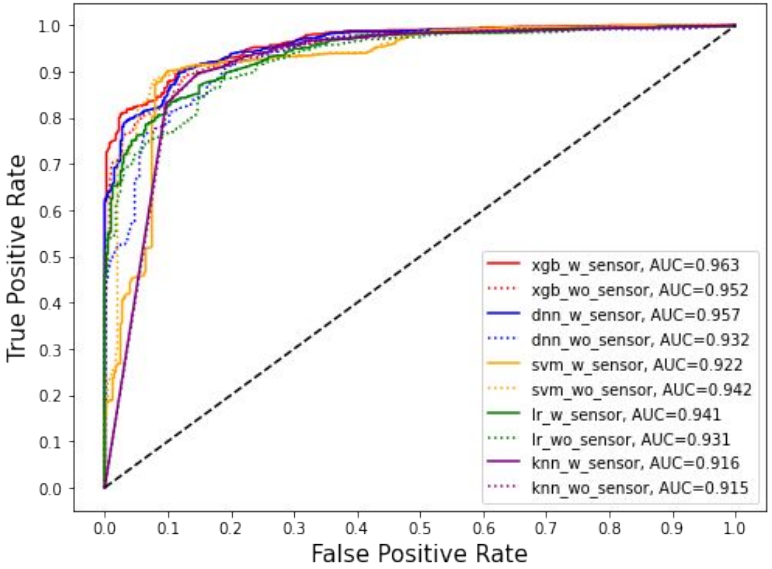
Results

Model	Balanced Accuracy (with sensor features)	Balanced Accuracy (without sensor features)
LR	0.855	0.842
KNN	0.851	0.846
SVM	**0.883**	**0.879**
XGB	0.870	0.863
DNN	0.870	0.872



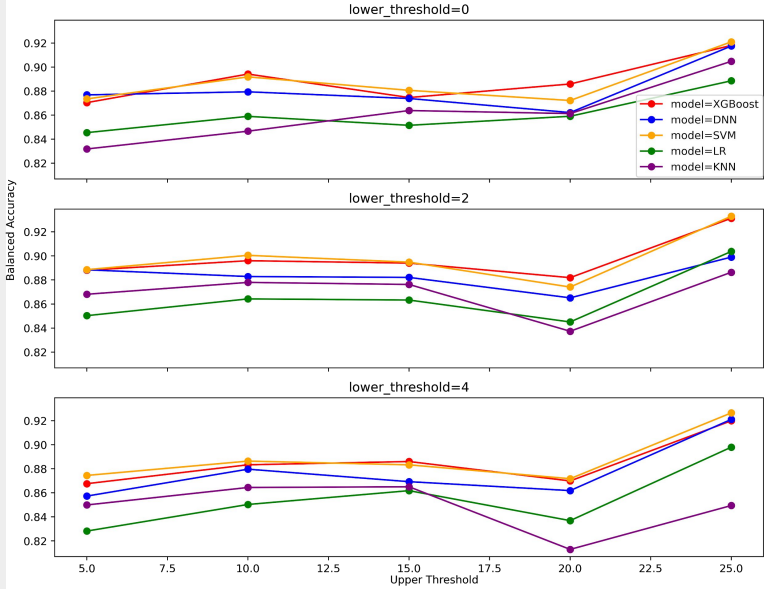
Findings

- Including sensor-related features leads to slight improvement across all models except for DNN.
- XGBoost has the highest AUC among all models.



Findings

- For our data modeling, we used a lower threshold of 0 and upper threshold of 5. Therefore, an app with a VirusTotal score of 0 is labelled as benign and an app with score >=5 is labelled as malware.
- The thresholds used to determine whether an app is malware or not might have an impact on performance.



Thank you!

Gather Location: 37