Repackaged
Android App
Detection

Columbia University
Data Science Capstone Project
Team 1

Presented by
Jianfeng Zhuang, Huazhang Liu, Chengyou Ju, Shaofeng Wu, Weitao Chen

Faculty Mentor: Professor Gail Kaiser
PhD Mentor: Shirish Singh
Overview

Harms of Android apps repackaging

- Deprive benefits
- Spread malwares
- Increase workload
How the Reverse Engineering Happened
Dataset: RePack

- Collected From AndroZoo
- 2776 original apps
- 15,297 repackaged apps
Pipeline to Extract Java Code

Fig. Java Code Extraction Pipeline
Exploratory Analysis and Observations

- 14827 pairs of repackaged and original apps extracted successfully
  - 7178 pairs have same sensors
  - 6789 pairs do not have any sensors
  - 804 repackaged apps have additional sensors than original apps
  - 61 original apps have additional sensors than repackaged apps

- 29 used sensors
- 32 used hardware and software features
- 114 used permissions
- Most used:
  - Sensor: Accelerometer
  - Feature: Touchscreen multitouch
  - Permission: Internet
Exploratory Analysis and Observations

Fig. Top Used Features and Permissions

Fig. Top Used Sensors
Exploratory Analysis and Observations

Additional Used Sensors in Malware Compared with Benign

- TYPE_PRESSURE: 1
- TYPE_LIGHT: 1
- TYPE_TEMPERATURE: 1
- TYPE_PROXIMITY: 7
- TYPE_ACCELEROMETER: 66
- TYPE_ROTATION_VECTOR: 75
- TYPE_GRAVITY: 75
- TYPE_LINEAR_ACCELERATION: 78
- TYPE_GYROSCOPE: 85
- TYPE_MAGNETIC_FIELD: 174
- TYPE_ALL: 228
- TYPE_ORIENTATION: 364

Fig. Additional Used Sensors in Repackaged than Original
Works with Imbalanced Dataset

Original dataset contains 15297 pairs of repackaged-to-original apps

Solve the imbalance problem by:

- Down Sampling
- Random Oversampling
- Duplicating original apps
- SMOTE

Major Evaluation Metrics:

- Balanced accuracy scores
Modeling on Sensor, Feature, Permission Data

The models we used:

- Baseline Model (predict everything as malware)
- Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbor (KNN)
- Random Forest
- XGBoost
- Multilayer Perceptron (MLP)

Two methods on oversampling:

- Oversample minor class
- SMOTE (Synthetic Minority Oversampling Technique)
Modeling on Sensor, Feature, Permission Data

Model results in balanced accuracy:

- **Oversampling Model - Top 3 Models:**
  - Random Forest: 0.70
  - Logistic Regression: 0.69
  - SVM: 0.69

- **SMOTE Model - Top 3 Models:**
  - Logistic Regression: 0.68
  - Random Forest: 0.68
  - XGBoosting: 0.67
Modeling on Sensor, Feature, Permission Data

Fig. Feature Importance on Random Forest
Flow Data

I. Control Flow Graph (CFG)

- Static analysis and compiler application
- Tool used: Androguard
- Pipeline

Fig. Control Flow Graph Pipeline
II. Data Flow Graph (DFG)

- Flow Droid: a generic, platform-independent data flow tracker and platform-specific extensions
- Pipeline

Fig. Data Flow Graph Pipeline
Flow Data

DFG - Taint Analysis

- Find untrustworthy sources and mark them as tainted
- Follow the “tags” to trace the flow of tainted objects

Fig. Taint Analysis
Modeling on Flow Data

- Mainly focus on 804 pairs
  - Repackaged apps have extra sensors
  - Analyze in CFG, DFG and Both

- Additional CFG Analysis on 7997 pairs:
  - Adding 7178 pairs with same sensors

- Strategies on Imbalance:
  - Random Oversampling
  - Duplicating originals by pairs

- Model Selection:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - K-Nearest Neighbors (KNN)
  - Random Forest
  - Gradient Boosting
  - XGBoost
  - Multi-layer Perceptron (MLP)

- Evaluation Metrics:
  - Balanced Accuracy (Main)
  - ROC-AUC Score
  - F1-Score
Modeling on Flow Data

Model results in balanced accuracy:

- **804 pairs on CFG - Top 3 Models:**
  - MLP: 0.96
  - Gradient Boosting: 0.94
  - Random Forest: 0.89

- **804 pairs on DFG - Top 3 Models:**
  - SVM: 0.77
  - Logistic Regression: 0.74
  - XGBoost Classifier: 0.71
Modeling on Flow Data

Model results in balanced accuracy:

- **804 pairs on both CFG and DFG - Top 3 Models:**
  - Logistic Regression: 0.96
  - SVM: 0.94
  - Gradient Boosting: 0.94

- **7997 pairs on CFG - Top 3 Models:**
  - Logistic Regression: 0.64
  - Random Forest: 0.62
  - XGBoost: 0.62
Modeling on Flow Data

Feature importance in best Gradient Boosting model of 804 pairs:

- **Top 3 Feature in CFG**
  - Mobclix Browser Activity to Sensor Manager
  - Full Screen Activity to Sensor Manager
  - Unit Player to Sensor Manager

- **Top 3 Feature in DFG**
  - Call Graph Construction Time
  - Number of Sinks
  - Maximum of Memory Consumption

- **Top 3 Feature in both CFG and DFG**
  - Full Screen Activity to Sensor
  - Call Graph Construction Time
  - Full Screen Activity to Sensor Manager
Conclusion

Repackaged Apps thread Android ecosystem

Sensor based detection on repackaged apps helps

- Java code information based on sensor classifies some original-repackaged pairs
- Flow paths (CFG and DFG) through sensor also are able to classify pairs

Potential future works:

- Java-language-based detection
- Other methods to extract sensor information
Thank You For Listening
Light Talk on Youtube: https://youtu.be/ECMqzvwGnes

Androzoo: https://androzoo.uni.lu

Javalang: https://github.com/c2nes/javalang

AndroGuard: https://androguard.readthedocs.io/en/latest/

FlowDroid: https://github.com/secure-software-engineering/FlowDroid
## Appendix - Model Results

### Fig. Model Results on Sensor, Feature, and Permission Dataset using Oversampling

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>roc_auc_score</th>
<th>precision_score</th>
<th>recall_score</th>
<th>balanced_accuracy_score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RF</strong></td>
<td>0.644388</td>
<td>0.752475</td>
<td>0.951397</td>
<td>0.624267</td>
<td>0.703111</td>
</tr>
<tr>
<td><strong>LR</strong></td>
<td>0.708347</td>
<td>0.745679</td>
<td>0.936959</td>
<td>0.713710</td>
<td>0.692694</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td>0.711545</td>
<td>0.745366</td>
<td>0.936424</td>
<td>0.718109</td>
<td>0.692388</td>
</tr>
<tr>
<td><strong>xbg</strong></td>
<td>0.603454</td>
<td>0.710569</td>
<td>0.939716</td>
<td>0.582845</td>
<td>0.663603</td>
</tr>
<tr>
<td><strong>mlp</strong></td>
<td>0.597378</td>
<td>0.697650</td>
<td>0.921884</td>
<td>0.588343</td>
<td>0.623745</td>
</tr>
<tr>
<td><strong>KNN</strong></td>
<td>0.480652</td>
<td>0.535450</td>
<td>0.889831</td>
<td>0.461877</td>
<td>0.535450</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>0.872402</td>
<td>0.500000</td>
<td>0.872402</td>
<td>1.000000</td>
<td>0.500000</td>
</tr>
</tbody>
</table>

### Fig. Model Results on Sensor, Feature, and Permission Dataset using SMOTE

<table>
<thead>
<tr>
<th></th>
<th>accuracy</th>
<th>roc_auc_score</th>
<th>precision_score</th>
<th>recall_score</th>
<th>balanced_accuracy_score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LR</strong></td>
<td>0.715382</td>
<td>0.712449</td>
<td>0.930647</td>
<td>0.728006</td>
<td>0.678539</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td>0.710265</td>
<td>0.710404</td>
<td>0.930529</td>
<td>0.721774</td>
<td>0.676677</td>
</tr>
<tr>
<td><strong>xbg</strong></td>
<td>0.628718</td>
<td>0.712880</td>
<td>0.940912</td>
<td>0.612903</td>
<td>0.674873</td>
</tr>
<tr>
<td><strong>RF</strong></td>
<td>0.579149</td>
<td>0.732736</td>
<td>0.939601</td>
<td>0.553152</td>
<td>0.655022</td>
</tr>
<tr>
<td><strong>mlp</strong></td>
<td>0.563799</td>
<td>0.673253</td>
<td>0.916870</td>
<td>0.549853</td>
<td>0.604501</td>
</tr>
<tr>
<td><strong>KNN</strong></td>
<td>0.457307</td>
<td>0.614943</td>
<td>0.890826</td>
<td>0.430718</td>
<td>0.534908</td>
</tr>
<tr>
<td><strong>Baseline</strong></td>
<td>0.872402</td>
<td>0.500000</td>
<td>0.872402</td>
<td>1.000000</td>
<td>0.500000</td>
</tr>
</tbody>
</table>
Appendix - Model Results

### Fig. Model Results on CFG of 804 pairs

<table>
<thead>
<tr>
<th>Model</th>
<th>Balanced Accuracy</th>
<th>ROC AUC Score</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Oversampling</td>
<td>Duplicate Oversampling</td>
<td>Random Oversampling</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.89</td>
<td>0.89</td>
<td>0.92</td>
</tr>
<tr>
<td>Support Vector Classifier</td>
<td>0.89</td>
<td>0.89</td>
<td>0.92</td>
</tr>
<tr>
<td>KNN</td>
<td>0.84</td>
<td>0.85</td>
<td>0.88</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.89</td>
<td>0.89</td>
<td>0.97</td>
</tr>
<tr>
<td>Gradient Boosting Classifier</td>
<td>0.92</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>XGBoost Classifier</td>
<td>0.85</td>
<td>0.85</td>
<td>0.89</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.96</td>
<td>0.95</td>
<td>0.99</td>
</tr>
</tbody>
</table>

### Fig. Model Results on DFG of 804 pairs

<table>
<thead>
<tr>
<th>Model</th>
<th>Balanced Accuracy</th>
<th>ROC AUC Score</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Oversampling</td>
<td>Duplicate Oversampling</td>
<td>Random Oversampling</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.79</td>
<td>0.74</td>
<td>0.77</td>
</tr>
<tr>
<td>Support Vector Classifier</td>
<td>0.76</td>
<td>0.77</td>
<td>0.85</td>
</tr>
<tr>
<td>KNN</td>
<td>0.64</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.70</td>
<td>0.84</td>
<td>0.87</td>
</tr>
<tr>
<td>Gradient Boosting Classifier</td>
<td>0.62</td>
<td>0.67</td>
<td>0.79</td>
</tr>
<tr>
<td>XGBoost Classifier</td>
<td>0.67</td>
<td>0.71</td>
<td>0.86</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.55</td>
<td>0.56</td>
<td>0.74</td>
</tr>
</tbody>
</table>
### Appendix - Model Results

#### Fig. Model Results on both CFG and DFG of 804 pairs

<table>
<thead>
<tr>
<th>Model</th>
<th>Balanced Accuracy</th>
<th>ROC AUC Score</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Oversampling</td>
<td>Duplicate Oversampling</td>
<td>Random Oversampling</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.86</td>
<td>0.96</td>
<td>0.94</td>
</tr>
<tr>
<td>Support Vector Classifier</td>
<td>0.94</td>
<td>0.93</td>
<td>0.98</td>
</tr>
<tr>
<td>KNN</td>
<td>0.65</td>
<td>0.63</td>
<td>0.64</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.94</td>
<td>0.92</td>
<td>0.96</td>
</tr>
<tr>
<td>Gradient Boosting Classifier</td>
<td>0.94</td>
<td>0.89</td>
<td>0.97</td>
</tr>
<tr>
<td>XGBoost Classifier</td>
<td>0.91</td>
<td>0.86</td>
<td>0.97</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.78</td>
<td>0.61</td>
<td>0.91</td>
</tr>
</tbody>
</table>

#### Fig. Model Results on CFG of 7997 pairs

<table>
<thead>
<tr>
<th>Model</th>
<th>Balanced Accuracy</th>
<th>ROC AUC Score</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random Oversampling</td>
<td>Duplicate Oversampling</td>
<td>Random Oversampling</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.61</td>
<td>0.63</td>
<td>0.61</td>
</tr>
<tr>
<td>Support Vector Classifier</td>
<td>0.49</td>
<td>0.61</td>
<td>0.46</td>
</tr>
<tr>
<td>KNN</td>
<td>0.55</td>
<td>0.53</td>
<td>0.57</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.50</td>
<td>0.62</td>
<td>0.57</td>
</tr>
<tr>
<td>Gradient Boosting Classifier</td>
<td>0.81</td>
<td>0.52</td>
<td>0.64</td>
</tr>
<tr>
<td>XGBoost Classifier</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
</tr>
<tr>
<td>Multilayer Perceptron</td>
<td>0.51</td>
<td>0.53</td>
<td>0.51</td>
</tr>
</tbody>
</table>