DeepIntent: Deep Icon-Behavior Learning for Detecting Intention-Behavior Discrepancy in Mobile Apps

Shengqu Xi¹*, Shao Yang²*, Xusheng Xiao², Yuan Yao¹, Yayuan Xiong¹, Fengyuan Xu¹, Haoyu Wang³, Peng Gao⁴, Zhuotao Liu⁵, Feng Xu¹, Jian Lu¹

¹ Nanjing University
² Case Western Reserve University
³ Beijing University of Posts and Telecommunications
⁴ University of California, Berkeley
⁵ University of Illinois at Urbana-Champaign

* The first two authors contributed equally to this research
Outline

• Background and Motivation
• DeepIntent Approach
  – Icon Widget Analysis
  – Deep Icon-Behavior Learning
  – Detecting Intention-Behavior Discrepancy
• Experiments
• Conclusions
Outline

• Background and Motivation
• DeepIntent Approach
  – Icon Widget Analysis
  – Deep Icon-Behavior Learning
  – Detecting Intention-Behavior Discrepancy
• Experiments
• Conclusions
Mobile Apps

- Mobile apps are playing an increasingly important role
  - E.g., travel, education, business
- Many apps access sensitive data to meet users' needs
  - E.g., camera, location, microphone
- However, malicious apps may also illegally collect sensitive data
  - E.g., exploiting users' private resources for advertising
Detecting Undesired Behaviors of Apps

• Industry: permission-based access control [statista. 2017]
  – Cons: Difficult to decide when to use the permission

• Research: undesired behavior patterns [Huang et al. USENIX Security’15, Nan et al. USENIX Security’15]
  – Cons: Only capture a fixed set of undesired behaviors

Our observation: the UI intentions perceived by users and the undesired behaviors of apps are usually incompatible
Intentions and Behaviors

• App's intentions to use sensitive data are often expressed via UI widgets
  – Mainly through icons and texts

• App’s behaviors are performed by program executions
  – Thousands of APIs, but mainly summarized using permissions
Detecting Intention-Behavior Discrepancy

<table>
<thead>
<tr>
<th>UI Widgets</th>
<th>Intention</th>
</tr>
</thead>
<tbody>
<tr>
<td>dial a number</td>
<td>CALL ✔</td>
</tr>
<tr>
<td>call</td>
<td>CALL ✔</td>
</tr>
<tr>
<td>timing filter</td>
<td>NONE ✗</td>
</tr>
</tbody>
</table>

- What are the **intentions** expressed from icons and contextual texts?
- What are the **behaviors** the Apps really perform?
- Are the behaviors **compatible** with the intentions?
Challenges

- **C1**: UI widgets’ intentions  
  - Difficult for computers to understand  
  - Lack of modeling joint semantics  

- **C2**: Program behaviors  
  - Difficult for precise analysis  
  - E.g., handlers, multi-threading, ICC  

- **C3**: Discrepancies  
  - Difficult to correlate intentions and behaviors
Insights

• **I₁**: Same type of sensitive behavior should have similar looks, e.g., to be evident to users
  – *Deep learning* to identify similar UI widgets

• **I₂**: Permission uses can be extracted by analyzing the source code of apps
  – *Static analysis* to map permissions to widgets

• **I₃**: Undesired behaviors usually contradict users expected specific looks
  – *Outlier analysis* to detect undesired behaviors
Outline

• Background and Motivation
• DeepIntent Approach
  – Icon Widget Analysis
  – Deep Icon-Behavior Learning
  – Detecting Intention-Behavior Discrepancy
• Experiments
• Conclusions
Overview of DeepIntent

Training APKs

Icon Widget Analysis

Icon-Behavior Association

Contextual Text Extraction

Icon-Permission Mappings

Contextual Texts for Icons

Deep Icon-Behavior Learning

Detecting Intention-Behavior Discrepancy

APK

Behavior Prediction

Outlier Detection

Icon-Behavior Model

Predicted Permission Use

Abnormal Permission Use

DeepIntent - CCS 2019
Overview of DeepIntent

• Phase 1: Icon Widget Analysis
  – Program analysis to extract features (i.e., icons and texts) and labels (i.e., permission uses) of icon widgets
Overview of DeepIntent

• Phase 2: Deep Icon-Behavior Learning
  – Training icon-behavior model based on both icons and their contextual texts, and the corresponding behaviors, i.e., permission uses
Overview of DeepIntent

- **Phase 3: Detecting Intention-Behavior Discrepancy**
  - Predicts permission uses for icon widgets, and detects abnormal permission uses
Phase 1: Icon-Behavior Analysis

- Icon-Widget Association
- Extended Call Graph Construction
- API Permission Checking
- Contextual Texts Extraction for Icons
Icon-Widget Association

• Associate the UI widgets with icons, i.e., drawable objects
  – Layout file: XML parsing
  – Source code: data flow analysis

• Adopt static analysis [Xiao et al. ICSE’19] to associate icons and UI widgets
Extended Call Graph Construction

• Associate the UI widgets with behaviors, i.e., API calls
  – Build call graph and patch missing links

<table>
<thead>
<tr>
<th>Caller</th>
<th>Callee</th>
</tr>
</thead>
<tbody>
<tr>
<td>setOnClickListener</td>
<td>onClick</td>
</tr>
<tr>
<td>Thread.start</td>
<td>Thread.run</td>
</tr>
<tr>
<td>AsyncTask.execute</td>
<td>doInBackground</td>
</tr>
<tr>
<td></td>
<td>onPreExecute</td>
</tr>
<tr>
<td></td>
<td>onPostExecute</td>
</tr>
<tr>
<td>sendMessage</td>
<td>handleMessage</td>
</tr>
</tbody>
</table>

Implicit caller and callee pairs captured, except for ICC methods
API Permission Checking

- Adopt PScout mapping [Kathy et al. CCS’12]
- Output the association between each icon and a set of permissions
- Allow one to many mapping
  - An icon can invoke one or more sensitive APIs
  - A sensitive API maps to multiple permissions
Contextual Texts Extraction for Icons

• Similar icons may reflect different intentions in different UI contexts

• Contextual texts
  – Layout texts that contained in the XML layout files
  – Icon-embedded texts
  – Resource names split by variable naming conventions
Phase 2: Deep Icon-Behavior Learning

- Icon Feature Extraction
- Text Feature Extraction
- Feature Combination
- Training Icon-Behavior Model
Icon Feature Extraction

- CNNs, e.g., DenseNet [Huang et al. CVPR’17], are successfully used in image recognition and model the icons.

- Adopt DenseNet with 4 channels (RGBA)
  - 4 dense blocks and 3 transition
  - Resize icons to 128 * 128
  - Output with 16 * 16 regions

\[ f_u = \text{DenseNet}(u) \]
Text Feature Extraction

• **RNNs** [Yang et al. NAACL’16] have been successfully applied in various natural language tasks, e.g., textual classification

• **Bidirectional** RNNs
  - Embed each word into vector with 100 dimension
  - Adopt GRU neurons
  - Max length is 20

\[
\begin{align*}
\vec{h}_i &= GRU(v_i, \vec{h}_{i-1}) \\
\hat{h}_i &= GRU(v_i, \hat{h}_{i-1}) \\
h_i &= [\vec{h}_i, \hat{h}_i] \\
f_v &= [h_1, h_2, ..., h_N]
\end{align*}
\]
Feature Combination

• Intuition
  – Icon and its text could be **semantically correlated**
  – **Simultaneously update the icon features and the text** features can capture the correlations

• **Co-Attention** [Lu et al. NeurIPS’16, Zhang et al. AAAI’19]
  – Compute correlation matrix
  – Transfer the features for each other

\[
C = \tanh(f_v^T W_c f_u) \quad H_u = \tanh(W_u f_u + (W_v f_v) C) \quad \tilde{f}_u = \sum_{i=0}^{M} a_u^i f_u^i \quad f = \tilde{f}_u + \tilde{f}_v
\]
Training Icon-Behavior Model

• Multi-label prediction problem
  – Predict each permission as a binary classification problem
  – Sigmoid function in logistic regression

\[ p = \text{sigmoid}(W_p f + b_p) \]

• Loss function
  – Binary cross entropy

\[ L = \frac{1}{D} \left( - \sum_{(p,z) \in D} \sum_i (p^i \ast \log(z^i) + (1 - p^i) \ast \log(1 - z^i)) \right) \]
Phase 3: Detecting Intention-Behavior Discrepancy

- Detecting group-wise outlier
- Computing final outlier score
Detecting Group-Wise Outlier

• Low-dimensional features
  – Tend to be more robust

• AutoEncoder: simple and effective
  – Reduce and reconstruct
    \[ g = \text{reduce}(f) \]
    \[ f' = \text{reconstruct}(g) \]
  – Minimize the reconstruction error
    \[ \min \sum_{j=1}^{d} (f^j - f'^j)^2 \]
Computing Final Outlier Score

• Aggregate
  – Combine prediction results

• Aggregation methods
  – Distance-based aggregation: local neighborhood density
    \[ s = \frac{s_1}{AvgDis_1} + \frac{s_2}{AvgDis_2} + \cdots + \frac{s_n}{AvgDis_n} \]
  – Prediction-based aggregation: predicted probabilities
    \[ s = s_1 \cdot (1 - p_1) + s_2 \cdot (1 - p_2) + \cdots + s_n \cdot (1 - p_n) \]
  – Combined aggregation
    \[ s = s_1 \cdot \left(1 - p_1 + \frac{1}{AvgDis_1}\right) + \cdots + s_n \cdot \left(1 - p_n + \frac{s_n}{AvgDis_n}\right) \]
Outline

• Background and Motivation
• DeepIntent Approach
  – Icon Widget Analysis
  – Deep Icon-Behavior Learning
  – Detecting Intention-Behavior Discrepancy
• Experiments
• Conclusions
Evaluation Setup: Implementation

- **Program analysis**
  - Gator [Rountev and Yan. CGO’14], Soot [Vallee-Rai et al. CC’00], ApkTool [Tumbleson et al. Github’17] and PScout [Au et al. CCS’12]

- **Icon processing**
  - Pillow [Clark. Github’10] and Google Tesseract Optical Character Recognition (OCR) [Smith et al. Github’06]

- **Deep learning**
  - Keras [Chollet et al. Keras’15] and PyOD [Zhao et al. JMLR’19]

Publicly available at https://github.com/deepintent-ccs/DeepIntent
Evaluation Setup: Subject

- **Benign apps**: 9,891
  - Google Play
  - No anti-virus engines flagged
- **Malicious apps**: 16,262
  - Resort to Wang et al. and RmvDroid
  - Flagged by at least 20 anti-virus engines

- **Total icons**: 7,691 (training) + 1,274 (benign testing) + 1,362 (malicious testing)
- **Manually labeled testing icons**: 1,274 + 1,362
Research Questions

• **RQ1**: How effective is the co-attention mechanism for icons and texts in improving icon-behavior learning?

• **RQ2**: How effective is icon-behavior association based on static analysis in improving icon-behavior learning?

• **RQ3**: How effective is DeepIntent in detecting intention-behavior discrepancies?
RQ1: Joint Feature Learning

<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>NETWORK</th>
<th>LOCATION</th>
<th>MICROPHONE</th>
<th>SMS</th>
<th>CAMERA</th>
<th>CALL</th>
<th>STORAGE</th>
<th>CONTACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICONINTENT</td>
<td>0.9054</td>
<td>0.3702</td>
<td>0.2334</td>
<td>0.3221</td>
<td>0.2917</td>
<td>0.2264</td>
<td>0.1286</td>
<td>0.2644</td>
</tr>
<tr>
<td></td>
<td>icon_only</td>
<td>0.9478</td>
<td>0.7006</td>
<td>0.7447</td>
<td>0.7817</td>
<td>0.7793</td>
<td>0.9138</td>
<td>0.8273</td>
<td>0.7725</td>
</tr>
<tr>
<td></td>
<td>text_only</td>
<td>0.9775</td>
<td>0.8505</td>
<td>0.8515</td>
<td>0.8651</td>
<td>0.8763</td>
<td>0.8028</td>
<td>0.7971</td>
<td>0.8489</td>
</tr>
<tr>
<td></td>
<td>concatenate</td>
<td>0.9665</td>
<td>0.8684</td>
<td>0.7827</td>
<td>0.8747</td>
<td>0.8935</td>
<td>1.000</td>
<td>0.8526</td>
<td>0.8809</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>0.9617</td>
<td>0.8588</td>
<td>0.8117</td>
<td>0.9076</td>
<td>0.9378</td>
<td>1.000</td>
<td>0.8915</td>
<td>0.8528</td>
</tr>
<tr>
<td></td>
<td>co-attention</td>
<td>0.9674</td>
<td>0.8675</td>
<td>0.8531</td>
<td>0.9389</td>
<td>0.9322</td>
<td>1.000</td>
<td>0.9137</td>
<td>0.9073</td>
</tr>
<tr>
<td></td>
<td>ICONINTENT</td>
<td>0.7488</td>
<td>0.7897</td>
<td>0.6985</td>
<td>0.6825</td>
<td>0.6885</td>
<td>0.7059</td>
<td>0.5454</td>
<td>0.5576</td>
</tr>
<tr>
<td></td>
<td>icon_only</td>
<td>0.987</td>
<td>0.6021</td>
<td>0.4355</td>
<td>0.5209</td>
<td>0.5558</td>
<td>0.6306</td>
<td>0.4548</td>
<td>0.5326</td>
</tr>
<tr>
<td></td>
<td>text_only</td>
<td>0.9807</td>
<td>0.8784</td>
<td>0.7473</td>
<td>0.7874</td>
<td>0.7512</td>
<td>0.7315</td>
<td>0.6802</td>
<td>0.7926</td>
</tr>
<tr>
<td></td>
<td>concatenate</td>
<td>0.9841</td>
<td>0.8426</td>
<td>0.5839</td>
<td>0.72</td>
<td>0.7238</td>
<td>0.6134</td>
<td>0.4054</td>
<td>0.7177</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>0.9858</td>
<td>0.8549</td>
<td>0.5606</td>
<td>0.6605</td>
<td>0.7214</td>
<td>0.65</td>
<td>0.468</td>
<td>0.7532</td>
</tr>
<tr>
<td></td>
<td>co-attention</td>
<td>0.9883</td>
<td>0.8325</td>
<td>0.6487</td>
<td>0.7763</td>
<td>0.7274</td>
<td>0.6111</td>
<td>0.5554</td>
<td>0.7705</td>
</tr>
</tbody>
</table>

- DeepIntent significantly outperforms IconIntent
RQ1: Joint Feature Learning

<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>NETWORK</th>
<th>LOCATION</th>
<th>MICROPHONE</th>
<th>SMS</th>
<th>CAMERA</th>
<th>CALL</th>
<th>STORAGE</th>
<th>CONTACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICONIntent</td>
<td>0.9054</td>
<td>0.3702</td>
<td>0.2334</td>
<td>0.3221</td>
<td>0.2917</td>
<td>0.2264</td>
<td>0.1286</td>
<td>0.2644</td>
</tr>
<tr>
<td></td>
<td>icon_only</td>
<td>0.9478</td>
<td>0.7066</td>
<td>0.7447</td>
<td>0.7817</td>
<td>0.7793</td>
<td>0.9138</td>
<td>0.8273</td>
<td>0.7725</td>
</tr>
<tr>
<td></td>
<td>text_only</td>
<td>0.9775</td>
<td>0.8505</td>
<td>0.8515</td>
<td>0.8651</td>
<td>0.8763</td>
<td>0.8028</td>
<td>0.7971</td>
<td>0.8489</td>
</tr>
<tr>
<td></td>
<td>concatenate</td>
<td>0.9665</td>
<td>0.8684</td>
<td>0.7827</td>
<td>0.8747</td>
<td>0.8935</td>
<td>1.000</td>
<td>0.8526</td>
<td>0.8809</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>0.9617</td>
<td>0.8588</td>
<td>0.8117</td>
<td>0.9076</td>
<td>0.9378</td>
<td>1.000</td>
<td>0.8915</td>
<td>0.8528</td>
</tr>
<tr>
<td></td>
<td>co-attention</td>
<td>0.9674</td>
<td>0.8675</td>
<td>0.8531</td>
<td>0.9389</td>
<td>0.9322</td>
<td>1.000</td>
<td>0.9137</td>
<td>0.9073</td>
</tr>
<tr>
<td></td>
<td>ICONIntent</td>
<td>0.7488</td>
<td>0.7897</td>
<td>0.6985</td>
<td>0.6825</td>
<td>0.6885</td>
<td>0.7059</td>
<td>0.5454</td>
<td>0.5576</td>
</tr>
<tr>
<td></td>
<td>icon_only</td>
<td>0.987</td>
<td>0.6021</td>
<td>0.4355</td>
<td>0.5209</td>
<td>0.5558</td>
<td>0.6306</td>
<td>0.4548</td>
<td>0.5326</td>
</tr>
<tr>
<td></td>
<td>text_only</td>
<td>0.9807</td>
<td>0.8784</td>
<td>0.7473</td>
<td>0.7874</td>
<td>0.7512</td>
<td>0.7315</td>
<td>0.6802</td>
<td>0.7926</td>
</tr>
<tr>
<td></td>
<td>concatenate</td>
<td>0.9841</td>
<td>0.8426</td>
<td>0.5839</td>
<td>0.72</td>
<td>0.7238</td>
<td>0.6134</td>
<td>0.4054</td>
<td>0.7177</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>0.9858</td>
<td>0.8549</td>
<td>0.5606</td>
<td>0.6605</td>
<td>0.7214</td>
<td>0.65</td>
<td>0.468</td>
<td>0.7532</td>
</tr>
<tr>
<td></td>
<td>co-attention</td>
<td>0.9883</td>
<td>0.8325</td>
<td>0.6487</td>
<td>0.7763</td>
<td>0.7274</td>
<td>0.6111</td>
<td>0.5554</td>
<td>0.7705</td>
</tr>
<tr>
<td></td>
<td>ICONIntent</td>
<td>0.8197</td>
<td>0.5041</td>
<td>0.3499</td>
<td>0.4327</td>
<td>0.4098</td>
<td>0.3429</td>
<td>0.2081</td>
<td>0.3587</td>
</tr>
<tr>
<td></td>
<td>icon_only</td>
<td>0.967</td>
<td>0.6476</td>
<td>0.5496</td>
<td>0.6252</td>
<td>0.6488</td>
<td>0.7462</td>
<td>0.5869</td>
<td>0.6305</td>
</tr>
<tr>
<td></td>
<td>text_only</td>
<td>0.9791</td>
<td>0.8642</td>
<td>0.796</td>
<td>0.8244</td>
<td>0.8089</td>
<td>0.7655</td>
<td>0.734</td>
<td>0.8196</td>
</tr>
<tr>
<td></td>
<td>concatenate</td>
<td>0.9732</td>
<td>0.8553</td>
<td>0.6688</td>
<td>0.7898</td>
<td>0.8016</td>
<td>0.7604</td>
<td>0.5495</td>
<td>0.791</td>
</tr>
<tr>
<td></td>
<td>add</td>
<td>0.9736</td>
<td>0.8568</td>
<td>0.6632</td>
<td>0.7646</td>
<td>0.8155</td>
<td>0.7829</td>
<td>0.6138</td>
<td>0.7999</td>
</tr>
<tr>
<td></td>
<td>co-attention</td>
<td>0.9777</td>
<td>0.8496</td>
<td>0.737</td>
<td>0.8499</td>
<td>0.8172</td>
<td>0.7586</td>
<td>0.6909</td>
<td>0.8333</td>
</tr>
</tbody>
</table>

- DeepIntent significantly outperforms IconIntent
- DeepIntent performs best compared to ‘text_only’ and ‘icon_only’ variants
### RQ1: Joint Feature Learning

<table>
<thead>
<tr>
<th>Metric</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>NETWORK</th>
<th>LOCATION</th>
<th>MICROPHONE</th>
<th>SMS</th>
<th>CAMERA</th>
<th>CALL</th>
<th>STORAGE</th>
<th>CONTACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICONIntent</td>
<td>0.9054</td>
<td>0.3702</td>
<td>0.2334</td>
<td>0.3221</td>
<td>0.2917</td>
<td>0.2264</td>
<td>0.1286</td>
<td>0.2644</td>
</tr>
<tr>
<td>icon_only</td>
<td>0.9478</td>
<td>0.7006</td>
<td>0.7447</td>
<td>0.7817</td>
<td>0.7793</td>
<td>0.9138</td>
<td>0.8273</td>
<td>0.7725</td>
</tr>
<tr>
<td>text_only</td>
<td>0.9775</td>
<td>0.8505</td>
<td>0.8515</td>
<td>0.8651</td>
<td>0.8763</td>
<td>0.8028</td>
<td>0.7971</td>
<td>0.8489</td>
</tr>
<tr>
<td>concatenate</td>
<td>0.9665</td>
<td>0.8684</td>
<td>0.7827</td>
<td>0.8747</td>
<td>0.8935</td>
<td>1.000</td>
<td>0.8526</td>
<td>0.8809</td>
</tr>
<tr>
<td>add</td>
<td>0.9617</td>
<td>0.8588</td>
<td>0.8117</td>
<td>0.9076</td>
<td>0.9378</td>
<td>1.000</td>
<td>0.8915</td>
<td>0.8528</td>
</tr>
<tr>
<td>co-attention</td>
<td>0.9674</td>
<td>0.8675</td>
<td>0.8531</td>
<td>0.9389</td>
<td>0.9322</td>
<td>1.000</td>
<td>0.9137</td>
<td>0.9073</td>
</tr>
<tr>
<td>ICONIntent</td>
<td>0.7488</td>
<td>0.7897</td>
<td>0.6985</td>
<td>0.6825</td>
<td>0.6885</td>
<td>0.7059</td>
<td>0.5454</td>
<td>0.5576</td>
</tr>
<tr>
<td>icon_only</td>
<td>0.987</td>
<td>0.6021</td>
<td>0.4355</td>
<td>0.5209</td>
<td>0.5558</td>
<td>0.6306</td>
<td>0.4548</td>
<td>0.5326</td>
</tr>
<tr>
<td>text_only</td>
<td>0.9807</td>
<td>0.8784</td>
<td>0.7473</td>
<td>0.7874</td>
<td>0.7512</td>
<td>0.7315</td>
<td>0.6802</td>
<td>0.7926</td>
</tr>
<tr>
<td>concatenate</td>
<td>0.9841</td>
<td>0.8426</td>
<td>0.5839</td>
<td>0.72</td>
<td>0.7238</td>
<td>0.6134</td>
<td>0.4054</td>
<td>0.7177</td>
</tr>
<tr>
<td>add</td>
<td>0.9858</td>
<td>0.8549</td>
<td>0.5606</td>
<td>0.6605</td>
<td>0.7214</td>
<td>0.65</td>
<td>0.468</td>
<td>0.7532</td>
</tr>
<tr>
<td>co-attention</td>
<td>0.9883</td>
<td>0.8325</td>
<td>0.6487</td>
<td>0.7763</td>
<td>0.7274</td>
<td>0.6111</td>
<td>0.5554</td>
<td>0.7705</td>
</tr>
<tr>
<td>ICONIntent</td>
<td>0.8197</td>
<td>0.5041</td>
<td>0.3499</td>
<td>0.4377</td>
<td>0.4098</td>
<td>0.3429</td>
<td>0.2081</td>
<td>0.3587</td>
</tr>
<tr>
<td>icon_only</td>
<td>0.967</td>
<td>0.6476</td>
<td>0.5496</td>
<td>0.6252</td>
<td>0.6488</td>
<td>0.7462</td>
<td>0.5869</td>
<td>0.6305</td>
</tr>
<tr>
<td>text_only</td>
<td>0.9791</td>
<td>0.8642</td>
<td>0.796</td>
<td>0.8244</td>
<td>0.8089</td>
<td>0.7655</td>
<td>0.734</td>
<td>0.8198</td>
</tr>
<tr>
<td>concatenate</td>
<td>0.9732</td>
<td>0.8553</td>
<td>0.6688</td>
<td>0.7898</td>
<td>0.8018</td>
<td>0.7604</td>
<td>0.5495</td>
<td>0.791</td>
</tr>
<tr>
<td>add</td>
<td>0.9736</td>
<td>0.8568</td>
<td>0.6662</td>
<td>0.7646</td>
<td>0.8155</td>
<td>0.7879</td>
<td>0.6138</td>
<td>0.7990</td>
</tr>
<tr>
<td>co-attention</td>
<td>0.9777</td>
<td>0.8496</td>
<td>0.737</td>
<td>0.8499</td>
<td>0.8172</td>
<td>0.7586</td>
<td>0.6909</td>
<td>0.8333</td>
</tr>
</tbody>
</table>

- DeepIntent significantly outperforms IconIntent
- DeepIntent performs best compared to ‘text_only’ and ‘icon_only’ variants
- Compared to others, co-attention performs especially well in 4 out of 8 permission groups
RQ2: Icon-Behavior Analysis

• Without program analysis -> Manifest file
  – Unused permissions and error Prone

• Re-trained with permissions from manifest files
  – Precision decrease dramatically
  – Essential to accurately extract icon-permission mappings
RQ3: Intention-Behavior Discrepancies

- Identifying intention-behavior discrepancies
  - Achieving 39.9% and 26.1% relative improvements on the benign apps and the malicious apps compared with IconIntent
  - Combining icon and text features are useful for discrepancy detection
  - Outperforms ‘prediction’ -> Essential to evolve outlier detection

- Precision and recall curves of DeepIntent
  - Precision results are high when K < #outliers
Outline

• Background and Motivation
• DeepIntent Approach
  – Icon Widget Analysis
  – Deep Icon-Behavior Learning
  – Detecting Intention-Behavior Discrepancy
• Experiments
• Conclusions
Conclusion

• DeepIntent
  – *Program analysis* techniques to associate the widgets to permission uses
  – *Deep learning* techniques to *jointly model* icons and their contextual texts of the icon widgets
  – Detecting the *intention-behavior discrepancies* by computing and aggregating the outlier scores

• Evaluation on 9,891 benign and 16,262 malicious apps
  – Achieves at least $19.3\%$ relative improvement in *predicting permission uses* compared with computer vision techniques
  – Program analysis is essential and achieves $70.8\%$ relative improvement on average compared to the learning approach *without program analysis*
  – Detect discrepancies with AUC value $0.8656$ and $0.8839$ for benign and malicious apps ($39.9\%$ and $26.1\%$ relative improvements over *IconIntent*)

**program analysis**
- Traceability & Label Inference
  - UI widgets <-> permissions
- Constructing a large-scale high-quality training dataset

**deep learning**
- Modeling unstructured artifacts
  - E.g., icons and texts
- Predicting expected intentions based on icons and texts