Project #1: Easily extending smartphone apps to wearable devices

**Problem:** Writing wearable apps is tedious

**Opportunity:** Most wearable apps simply mirror the smartphone app

**UIWear:** Write once, extend to many

**Idea:** Decouple app design from app management.

**Architecture:** Developer writes simple metaprogram specifying app design

**UIWear creates wearable app**

**Implementation:** On Android OS, AndroidWear, Sony SmartGlass

**Evaluation:** Created 20 wearable apps with fraction of dev effort (some examples Lines-of-Code (LoC))

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Lines of Code (LoC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AndroidWear</td>
<td>716</td>
</tr>
<tr>
<td>Wearable App</td>
<td>2,163</td>
</tr>
<tr>
<td>Bookstore</td>
<td>1,802</td>
</tr>
<tr>
<td>Facebook</td>
<td>6,522</td>
</tr>
</tbody>
</table>

**UIWear published at ACM MobiCom 2017**

---

**Wearable Sensor Virtualization (Ongoing)**

**Problem:** Wearable sensors are not fully utilized by phone apps.

**Idea:** Design a sensor virtualization platform for phone apps to seamlessly access wearable sensors

---

**Project #2: Enabling Deep Learning on Phones for privacy and performance**

**Background:** Recurrent Neural Networks (RNNs) improve activity recognition, machine translation, and other tasks

**Problem:** Optimizations to run deep learning on phones focus on Convolutional Neural Networks (CNNs); Don’t work for RNNs.

**E.g.** GPU offloading harder on RNNs due to dependencies

**MobiRNN:** Offloading RNN models to GPU

**MobiRNN idea:** a mobile specific optimization for RNNs that parallelize within a cell rather than across the cells.

**MobiRNN performs coarse-grained parallelization where a row of computation is offloaded to a GPU core**

**Implementation:** On TensorFlow, using RenderScript framework

**Evaluation:** RNN models run 3-9 times faster on GPU, using MobiRNN

---

**Project #3: Private Intelligent Assistant (PrIA)**

**Problem:** Personalized intelligent assistance costs privacy

**PrIA Idea:** Enable intelligent assistance services completely locally from users personal devices

**PrIA News Recommendation**

**Key idea:** Decouple news aggregation from personalization

**Architecture:** PrIA downloads all news articles from an aggregation service, builds a local user profile, and provides recommendations, all on the users personal device

**PrIA App**

**User Study:** 6 users for an average of 10 days. Users rate recommendations from PrIA and Google News

PrIA performs poorer than Google News, but not significantly so

**PrIA News Recommendation**

<table>
<thead>
<tr>
<th>Recommendation system</th>
<th>Precision @10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>0.45</td>
</tr>
<tr>
<td>PrIA</td>
<td>0.38</td>
</tr>
</tbody>
</table>

**PrIA published at HotMobile 2017**

**Ongoing work**

- In-depth study of privacy leakage in intelligent home assistants such as Alexa and Google Home
- Designing techniques to reduce these privacy leakages by leveraging ideas from NLP and privacy research.

**Project Website:** pria.cs.stonybrook.edu

---

**Overview of the Mobile Systems research at NetSys Lab**

Jian Xu, Qingqing Cao, Aruna Balasubramanian (In collaboration with HexLab, LUNR lab, OSCAR lab, HI Lab, and Nokia)