SmartSPEC: Customizable Smart Space Datasets via Event-Driven Simulations

Andrew Chio¹, Daokun Jiang¹, Peeyush Gupta¹, Georgios Bouloukakis², Roberto Yus³, Sharad Mehrotra¹, Nalini Venkatasubramanian¹

Virtual, March 23, 2022
IEEE PerCom 2022

¹ University of California, Irvine
² Télécom SudParis, IP Paris
³ University of Maryland, Baltimore County
IoT-Enabled Smart Spaces

Internet-of-Things (IoT)

Healthcare
- Hospitals
- Senior Homes

Facility Management
- Consumable Monitoring
- Elevators

Safety
- Stores
- Residential Homes

Benefits:
- Energy Efficiency, Sustainability
- Building Resilience, Reliability
- Adaptability to Dynamic Conditions
Towards Smarter Buildings: The Need for Realistic Data

- Realistic data is necessary to test and validate smart space approaches in heterogeneous human environments
  - Evaluating robustness of algorithms
  - Failure testing
  - Scalability testing
  - Operating in extreme scenarios

Fire Evacuation in a High-Rise Building
Challenge: Obtaining Real Data

Deployment of Sensors
• Cost & sensor placement

Recruitment of Participants
• Reluctance to share data
• Time-consuming
• Limited in scale

Preservation of Participant Privacy
• Data regulations
• Leakage of sensitive data

FERPA
Family Educational Rights and Privacy Act

HIPAA

GDPR
Generating Realistic Synthetic Data with Simulators

Challenge: Modeling smart spaces accurately
- Variability/dynamicity of activities
- Faithfulness to reality

Approach 1:
Extend previously captured dataset\(^1\)
- Issue: violates causality, limited to initial space

Approach 2:
Generate data randomly based on sensor models\(^2\)
- Issue: random ≠ realistic

Approach 3:
Create dataset based on interactions of people and their activities\(^3\)
- Issue: Semantic Explainability - Why people visit the spaces that they do?

Activities of Daily Living:
- Brushing
- Toileting
- Walking

\(^1\) Replication, Modification, Sampling: Tay et al., UpSizeR (Information Systems ‘13)
\(^2\) Random Data Generation: Mockaroo, Hoag and Thompson, PSDG (ACM SIGMOD Record ‘07)
\(^3\) Activities of Daily Living: Alshammari et al., OpenSHS, Sensors ‘17
Mobility Models and Trajectory Models: Rhee et al., IEEE/ACM TON ‘12; Alessandretti et al., Nature ‘20
Trajectory Models: Brinkoff, Geoinformatica ‘02; Pelekis et al., ACM Sigspatial ‘15
Generative Models: Gupta et al., CVPR ‘18; Rossi et al., Pattern Recognition ‘21
The SmartSPEC Approach

Exploit semantics to generate realistic synthetic smart space datasets

Applications
Event Analysis
Sustainable Buildings
Safety/Surveillance
Covid Tracking

Seed Dataset (Observed) → SmartSPEC → Synthetic Smart Space Data

Sensors
People
Spaces
The Contributions of this Paper

**Scenario:** a digital depiction of the activities and operations in the smart space

**SmartSPEC Platform**

- **Seed Dataset (Observed)**
- **Scenario Learning**
- **Semantic Model**
- **Scenario Generation**
- **Synthetic Smart Space Data**
- **Assessing Realism**
SmartSPEC : Semantic Model

Seed Dataset (Observed) → Scenario Learning → Semantic Model → Scenario Generation → Synthetic Smart Space Data → Assessing Realism

SmartSPEC Platform

Scenario Generation

Semantic Model

Spaces, Sensors, Events, People

Assessing Realism

Synthetic Smart Space Data

Seed Dataset

Observed
Smart Space: A Semantic Characterization

**Sensors S**
- id
- coverage
- interval
- observes
- in
- occupies
- carried_by
- id
- coordinates
- capacity
- adjacency
- attends
- in

**Spaces C**
- id
- coverage
- interval
- observes
- in
- occupies
- carried_by
- id
- capacity
- adjacency
- attends
- in

**People P**
- id
- affinities
- TimeProfile
- carried_by
- observes
- attends
- in

**Events E**
- id
- attendance
- time
- periodicity
- carried_by
- observes
- attends
- in
SmartSPEC : Scenario Learning

**SmartSPEC Platform**

Seed Dataset (Observed) → Scenario Learning → Semantic Model → Scenario Generation → Synthetic Smart Space Data  
Assessing Reality
SmartSPEC : Scenario Learning

- SmartSPEC Platform
  - Scenario Learning
  - Event Learner
  - People-Event Interaction Learner

- Semantic Model
  - Spaces
  - Sensors
  - Events
  - People

- Scenario Generation

- Seed Dataset (Observed)

- Synthetic Smart Space Data

- Assessing Realism
Learning Events through Occupancy

For each space $C$

Occupancy $\lambda^{c,t_s,t_e}_D$

- Number of unique people from dataset $D$ that are in space $C$ during time period $(t_s, t_e)$.

Presence $\rightarrow$ Occupancy $\rightarrow$ Events
Learning Events

Intuition:

<table>
<thead>
<tr>
<th>Algorithm 1: Extracting Events, Learning MetaEvents.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Dataset $D$, Spaces $C$, Date $start$, Date $end$, int $b$</td>
</tr>
<tr>
<td><strong>Output:</strong> Events $\mathcal{E}$, MetaEvents $\mathcal{M}_\mathcal{E}$</td>
</tr>
<tr>
<td>1 $\mathcal{E} \leftarrow \emptyset$</td>
</tr>
<tr>
<td>2 for $d \leftarrow start$ . . . $end$ do</td>
</tr>
<tr>
<td>3 for $c \leftarrow C$ do</td>
</tr>
<tr>
<td>4 $data \leftarrow D.query(space = c, day = d)$</td>
</tr>
<tr>
<td>5 $ts \leftarrow computeOccupancy(data, minutes = b)$</td>
</tr>
<tr>
<td>6 $bkpts \leftarrow changePointDetection(ts)$</td>
</tr>
<tr>
<td>7 $\mathcal{E} \leftarrow \mathcal{E} \cup createEvents(c, bkpts)$</td>
</tr>
<tr>
<td>8 $distMat \leftarrow computeDistanceMatrix(\mathcal{E})$</td>
</tr>
<tr>
<td>9 $clusters \leftarrow doAgglomerativeClustering(distMat)$</td>
</tr>
<tr>
<td>10 $\mathcal{M}_\mathcal{E} \leftarrow makeMetaEvents(clusters)$</td>
</tr>
<tr>
<td>11 return $\mathcal{E}, \mathcal{M}_\mathcal{E}$</td>
</tr>
</tbody>
</table>

Intuition:

*Change Point Detection*

Create time-series of occupancy in space $C$ on date $d$

Use *Change Point Detection* to learn when one event ends, and another starts

Presence $\rightarrow$ Occupancy $\rightarrow$ Events

Breakpoints occur when there are large changes in occupancy

Occupancy stays roughly consistent during an event
Learning Events

### Algorithm 1: Extracting Events, Learning MetaEvents.

**Input:** Dataset $D$, Spaces $C$. Date $start$, Date $end$, int $b$

**Output:** Events $\mathcal{E}$, MetaEvents $\mathcal{ME}$

1. $\mathcal{E} \leftarrow \emptyset$
2. for $d \leftarrow start \ldots end$ do
3.   for $c \leftarrow C$ do
4.     data $\leftarrow D.query(space = c, day = d)$
5.     ts $\leftarrow$ computeOccupancy(data, minutes = $b$)
6.     bkpts $\leftarrow$ changePointDetection(ts)
7.     $\mathcal{E} \leftarrow \mathcal{E} \cup$ createEvents($c, bkpts$)
8. distMat $\leftarrow$ computeDistanceMatrix($\mathcal{E}$)
9. clusters $\leftarrow$ doAgglomerativeClustering(distMat)
10. $\mathcal{ME} \leftarrow$ makeMetaEvents(clusters)
11. return $\mathcal{E}, \mathcal{ME}$

---

**Intuition:**

**Agglomerative Clustering**

- Each event starts in its own cluster, and is merged with other “nearby” clusters
- Terminates once distance between clusters $\geq$ threshold $\epsilon$
- Cluster distance based on set of attendees and time of event

**Jaccard Index**

- Given two sets $A$ and $B$, define similarity ratio $r = \frac{\text{card}(A \cap B)}{\text{card}(A \cup B)}$.
- Interpretation: $r = 1$ only if $A = B$.

**Intuition:**

Create time-series of occupancy in space $C$ on date $d$

Use Change Point Detection to learn when one event ends, and another starts

Use Agglomerative Clustering to learn types of events

Presence $\rightarrow$ Occupancy $\rightarrow$ Events
Learning People-Event Interactions

Learned Events:

• Event $e_1$: attendees = \{p_1, p_2, p_3\}

• Event $e_2$: attendees = \{p_2, p_3\}

• Event $e_3$: attendees = \{p_1\}

• Event $e_4$: attendees = \{p_3\}

• Event $e_5$: attendees = \{p_1, p_2\}

Characterize people based on attended events

Person $P_1$

attended: \{e_1, e_3, e_5\}

Person $P_2$

attended: \{e_1, e_5\}

Person $P_3$

attended: \{e_1, e_2, e_4\}

Apply Agglomerative Clustering to group people by similarity of attended events (until a threshold $\epsilon$)
SmartSPEC : Scenario Generation

**SmartSPEC Platform**

- **Scenario Learning**
  - Event Learner
  - People-Event Interaction Learner

- **Semantic Model**
  - Spaces
  - Sensors
  - Events
  - People

- **Scenario Generation**

- **Assessing Realism**
  - Synthetic Smart Space Data

**Seed Dataset (Observed)**

- **Spaces**
- **People**
- **Sensors**

Image represents a diagram showing the flow of data from observed datasets through the SmartSPEC platform, involving event and interaction learning, leading to scenario generation and assessing realism.
SmartSPEC : Scenario Generation

SmartSPEC Platform

Scenario Learning
- Event Learner
- People-Event Interaction Learner

Semantic Model
- Spaces
- Sensors
- Events
- People

Scenario Generation
- Entity Generator
- Synthetic Data Generator

Seed Dataset (Observed)

Synthetic Smart Space Data
(Trajectory, Sensor Data Observations, Occupancy)

Assessing Realism
Given types of events and profiles of people, how can we create a new set of events and people for our synthetic dataset?

**Generating a new Event**
- **Type?**
- **When?**
- **Where?**
- **How many people?**

**Generating a new Person**
- **Profile?**
- **Affinity?**
- **Enter/Exit Times?**
### Algorithm 3: Synthetic data generation.

**Input:** Date $d_s$, Date $d_e$, People $P$, Events $E$, Spaces $C$  
**Output:** LogFile $log$

```plaintext
1 log ← ∅  
2 for $P$ ← $P$ do  
3   for $d, t_s, t_e$ ← $P$.queryActiveDateTime($d_s$, $d_e$) do  
4     $t$ ← $t_s$  
5     while $t ≤ t_e$ do  
6       if $E$ is null then  
7         path ← getPath($P$.space, $E$.space)  
8       else  
9         attd ← ∅  
10        for $E$ ← $E$ do  
11          if $E$.hasSpaceCapacity($t$)  
12             or $E$.hasPeopleCapacity($P$)  
13             or $E$.conflictsWith($P$.prevEvents) then  
14                continue  
15            $P_e$ ← getPath($P$.space, $E$.space)  
16            arrival ← $t + P_e$.estTravelTime()  
17            if $|arrival - E$.startTime| ≥ $\epsilon$ then  
18                continue  
19          $E$, path ← select(attd, $P$.eventAffinity)  
20        end  
21     end  
22     for $c$ ← path do  
23       Block until $C_c$.cap($d, t$) ≤ $C_c$.maxCap  
24       Move $P$ to $c$, updating $t$  
25       log.record($P$, $E$.space, $E$.time)  
26       $P$.recordAttendance($E$)  
27     end  
28 return log
```

### Intuition:

- Get date/time that person is in the smart space
- Choose an event to attend, preferably a previously attended periodic event
- Semantic Constraints on spaces, people, events
- Estimate travel time; estimated arrival must be within a threshold $\epsilon$
- Move to an event space
- Record data in log file
SmartSPEC: Assessing Realism

**SmartSPEC Platform**

**Scenario Learning**
- Event Learner
- People-Event Interaction Learner

**Semantic Model**
- Spaces
- Sensors
- Events
- People

**Scenario Generation**
- Entity Generator
- Synthetic Data Generator

Seed Dataset (Observed) → Scenario Learning → Semantic Model → Scenario Generation → Synthetic Smart Space Data (Trajectory, Sensor Data Observations, Occupancy) → Assessing Realism
SmartSPEC : Assessing Realism

SmartSPEC Platform

Scenario Learning

- Event Learner
- People-Event Interaction Learner

Scenario Generation

- Entity Generator
- Synthetic Data Generator

Semantic Model

Seed Dataset (Observed)

Synthetic Smart Space Data (Trajectory, Sensor Data Observations, Occupancy)

Assessing Realism
Assessing Realism of Smart Space Datasets

Dataset $D$ → Generator $G$ → Dataset $D'$

- **Sensors $S$**
- **People $P$**
- **Events $E$**
- **Spaces $C$**

How to quantify the realism of $D$, $D'$?

- **Occupancy**: a space's perspective of the dataset
- **Trajectory**: a person’s perspective of the dataset

<table>
<thead>
<tr>
<th>Person P</th>
<th>Space C</th>
<th>DateTime t</th>
</tr>
</thead>
<tbody>
<tr>
<td>f28c94f</td>
<td>1412</td>
<td>2017-09-01 08:19:00</td>
</tr>
<tr>
<td>f20a461</td>
<td>6029</td>
<td>2017-09-01 08:19:00</td>
</tr>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:07</td>
</tr>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:26</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Similarity of Space’s Occupancy

- **Occupancy** of space $C$: number of unique people in space $C$ during time period $(t_s, t_e)$.
- **Occupancy Distance** is the mean squared error in occupancy over time.

\[
\frac{1}{|C| |t_s - t_e|} \sum_{c \in C} \sum_{t_s, t_e \in T} |\lambda_{D,c,t_s,t_e} - \lambda_{D',c,t_s,t_e}|^2
\]
Consider the following:

**Dataset** $D$  
**Dataset** $D'$

- Extract Trajectory
- Extract Trajectory

$\delta_{D}^{(i)} \in \Delta_{D}$

$\delta_{D'}^{(j)} \in \Delta_{D'}$

**Person P**  
**Space C**  
**DateTime t**

<table>
<thead>
<tr>
<th>Person P</th>
<th>Space C</th>
<th>DateTime t</th>
</tr>
</thead>
<tbody>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:07</td>
</tr>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:26</td>
</tr>
<tr>
<td>238be6</td>
<td>3254</td>
<td>2017-09-01 08:20:50</td>
</tr>
<tr>
<td>238be6</td>
<td>3256</td>
<td>2017-09-01 08:21:13</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

• **Trajectory** of person $P$: sequence of spaces $C$ visited by $P$ over datetime $t$
  
  • Should we naively compare all trajectories against each other?
Similarity of People’s Trajectory

- Control Variables are applied to partition trajectories into comparable bins. 
  e.g., $V = (t_s, t_e) = (1:00, 1:30)$ contains trajectories with $t_s \approx 1:00$, $t_e \approx 1:30$. 

Dataset $D$

Extract Trajectory

$\delta^{(i)}_D \in \Delta_D$

Apply Control Variable $V$

$\Delta^V_D$

Dataset $D'$

Extract Trajectory

$\delta^{(j)}_{D'} \in \Delta_{D'}$

Apply Control Variable $V$

$\Delta^V_{D'}$

<table>
<thead>
<tr>
<th>Person P</th>
<th>Space C</th>
<th>DateTime t</th>
</tr>
</thead>
<tbody>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:07</td>
</tr>
<tr>
<td>238be6</td>
<td>3231</td>
<td>2017-09-01 08:19:26</td>
</tr>
<tr>
<td>238be6</td>
<td>3254</td>
<td>2017-09-01 08:20:50</td>
</tr>
<tr>
<td>238be6</td>
<td>3256</td>
<td>2017-09-01 08:21:13</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Similarity of People’s Trajectories

**Distance Function $\Phi$**

<table>
<thead>
<tr>
<th>$t_s$</th>
<th>$t_e$</th>
<th>1:00</th>
<th>1:30</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Let $\Phi(\delta^1, \delta^2)$ be a function that computes the distance between two trajectories.

- e.g., Fréchet Distance Metric

$\Delta^V_D$

$\Delta^V_{D'}$

<table>
<thead>
<tr>
<th>$t_s$</th>
<th>$t_e$</th>
<th>1:00</th>
<th>1:30</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How do we compare multiple trajectories against one another?**
Similarity of People’s Trajectories

Distance Function $\Phi$

$\Delta_D^v$

<table>
<thead>
<tr>
<th>$t_s$</th>
<th>$t_e$</th>
<th>1:00</th>
<th>1:30</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

$\Delta_D^{v'}$

<table>
<thead>
<tr>
<th>$t_s$</th>
<th>$t_e$</th>
<th>1:00</th>
<th>1:30</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00</td>
<td></td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1:30</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

$t_s, t_e = (1:00, 1:00)$

$\Phi (\n \n, \n \n) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$t_s, t_e = (1:00, 1:30)$

$\Phi (\n \n, \n \n) = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

$t_s, t_e = (1:30, 1:30)$

$\Phi (\n \n, \n \n) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

Trajectory Distance

$$\frac{1}{|V|} \sum_{(v^{(i)}, v^{(j)}) \in M} \Phi (\delta^{(i)}, \delta^{(j)}) + \alpha (|\Delta_D^v| - |\Delta_D^{v'}|)$$

Penalty Term for difference in trajectory set sizes

• **Match** trajectories between corresponding bins
• Matching matrix $M$ does not need to be injective
Interpreting Dataset Similarity

How to determine if generator $G$ produces realistic datasets?
Interpreting Dataset Similarity

Compare distances between pairs of real datasets

How do \textit{real} datasets vary against other \textit{real} datasets?

Compare distances between pairs of real and simulated datasets

How do \textit{real} datasets differ from \textit{synthetic} datasets?

How well does synthetic data mimic the seed from which it was produced?
Interpreting Dataset Similarity

Compare distances between pairs of real datasets

*How do real datasets vary against other real datasets?*

Simulated $\approx$ Real?

Compare distances between pairs of real and simulated datasets

*How do real datasets differ from synthetic datasets?*

$D_i$

$D_{i,k}$

$D'_i$

How well have we extracted patterns from one dataset and applied them to the next?
Experiment: 2 Distinct Scenarios

Scenario 1: Campus
• 6 floor campus building: 125+ faculty offices, 10 classrooms, 4 lecture halls
• 64 WiFi Access Points (WiFi APs)
• 5 weeks of WiFi connectivity events, ~300K connections/week, partitioned into 5 periods of 1 week each

Scenario 2: City – GeoLife GPS Trajectories¹
• GPS trajectories in city of Beijing, China
• 1150 points of interest to cluster GPS data
• 63 people over 28 months, ~36K GPS data/month, partitioned into 1-month periods

Learned types of events / profiles of people from both scenarios

Learned Events in Campus Scenario

Events

• 510 “ground truth” events
• Best-effort mapping of events to WiFi APs

• Average paired difference between:
  • Event Start Time: $15 \pm 18$ mins
  • Event End Time: $21 \pm 27$ mins
Baselines and Metrics

Mobility Model Baselines

- **Random Waypoint (RAND)**: Next visited space is random
- **Brownian Motion (BROW)**: Next visited space is adjacent
- **Lévy Flight (LÉVY)**: Next visited space is chosen by following a power law distribution on distance
- **Exponential Preferential Return (EPR)**: Same as Lévy Flight but selects previously visited spaces with higher probability

Comparison Metrics

- **Trajectory Distance**: Average paired Fréchet distance controlled over start/end times
  - Start/End Times on 30-minute blocks
- **Occupancy Distance**: Average difference in occupancy
  - Over 5-minute intervals
- Averaged results from 3 simulations, comparing against next week (campus scenario) or month (city scenario)
Evaluating Realism in Campus Scenario

On average, there was a 35% difference in trajectory distances between SmartSPEC and the campus dataset.

On average, there was a 36% difference in occupancy counts per space between SmartSPEC and the campus dataset.

Most mobility models do significantly worse.

SmartSPEC produces trajectories and occupancy counts that are close to real data on the scope of a campus building.

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartSPEC</td>
<td>263.92</td>
<td>252.09</td>
<td>272.43</td>
<td>240.99</td>
</tr>
<tr>
<td>RAND</td>
<td>789.8</td>
<td>754.07</td>
<td>740.23</td>
<td>606.74</td>
</tr>
<tr>
<td>BROW</td>
<td>533.27</td>
<td>479.68</td>
<td>501.39</td>
<td>407.32</td>
</tr>
<tr>
<td>LÉVY</td>
<td>760.3</td>
<td>713.53</td>
<td>713.18</td>
<td>583.97</td>
</tr>
<tr>
<td>EPR</td>
<td>693.38</td>
<td>554.26</td>
<td>635.81</td>
<td>459.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Week 1</th>
<th>Week 2</th>
<th>Week 3</th>
<th>Week 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>185.65</td>
<td>188.67</td>
<td>191.31</td>
<td>194.60</td>
</tr>
<tr>
<td>SmartSPEC</td>
<td>8.63</td>
<td>10.0</td>
<td>7.16</td>
<td>8.61</td>
</tr>
<tr>
<td>RAND</td>
<td>14.20</td>
<td>13.92</td>
<td>14.01</td>
<td>13.65</td>
</tr>
<tr>
<td>BROW</td>
<td>12.29</td>
<td>12.37</td>
<td>12.75</td>
<td>12.34</td>
</tr>
<tr>
<td>LÉVY</td>
<td>13.83</td>
<td>13.49</td>
<td>13.64</td>
<td>13.23</td>
</tr>
<tr>
<td>EPR</td>
<td>14.75</td>
<td>12.86</td>
<td>14.83</td>
<td>10.05</td>
</tr>
</tbody>
</table>
Evaluating Realism in City Scenario

• On average, there was a **13% difference in trajectory distances** between SmartSPEC and the GeoLife dataset.

• On average, there was a **37% difference in occupancy counts per space** between SmartSPEC and the GeoLife dataset.

• Brownian motion baseline creates similar trajectories to real data, but have very different occupancy.

SmartSPEC produces trajectories and occupancy counts that are close to real data on the scope of a city.
Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC
SmartSPEC: Workflow

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

Sample Seed Data

```plaintext
[learners]
start = 2017-04-01
end = 2017-05-01
unit = 5
validity = 10
smooth = EMA
window = 10
time-thresh = 30
occ-thresh = 1

[filepath]
spaces = data/demo/Spaces.json
sensors = data/demo/Sensors.json
metaevents = data/demo/MetaEvents.json
metapeople = data/demo/MetaPeople.json
...```
SmartSPEC: Workflow

Our code is publicly available on GitHub: https://github.com/andrewgchio/SmartSPEC

Sample Configuration File for Scenario Generation

```json
[people]
number = 500

generation = all

[events]
number = 5000

generation = diff

[synthetic-data-generator]
start = 2018-01-08
day = 2018-01-29

[filepath]
metapeople = data/demo/MetaPeople.json
metaevents = data/demo/MetaEvents.json
spaces = data/demo/Spaces.json

sensors = data/demo/Sensors.json
people = data/demo/People.json

events = data/demo/Events.json
output = data/demo/output/
```
SmartSPEC: Workflow

Our code is publicly available on GitHub: [https://github.com/andrewgchio/SmartSPEC](https://github.com/andrewgchio/SmartSPEC)

Sample Configuration File for Scenario Generation

```plaintext
[people]
number = 500
generation = all

[events]
number = 5000
generation = diff

[synthetic-data-generator]
start = 2018-01-08
end = 2018-01-29

[filepath]
metapeople = data/demo/MetaPeople.json
metaevents = data/demo/MetaEvents.json
spaces = data/demo/Spaces.json
sensors = data/demo/Sensors.json
people = data/demo/People.json
events = data/demo/Events.json
output = data/demo/output/
...
SmartSPEC: Applicability and Utility

Sample of Synthetic Data Output

```
PersonID, EventID, SpaceID, StartDatetime, EndDatetime
60, 1660, 1422, 2018-01-15 09:59:19, 2018-01-15 10:37:00
...```

Sample Generated Dataset
SmartSPEC: Applicability and Utility

TIPPERS: Testbed for IoT-based Privacy-Preserving PERvasive Spaces
• Design robust, experimental testbed
• Explore privacy technologies
• Real-world deployments

NAVWAR Trident Warrior:
• Explore potential benefits of IoT technologies for naval use cases
• Day in the life of a sailor in mission-critical scenarios and non-mission-critical scenarios
  • Simulated activities on a Navy Ship

Credit: Navy Media Content Services
Key Takeaways

- **Realistic and Semantically Explainable data** are required to test and validate smart space approaches.

- We developed SmartSPEC: an **event-driven** smart space simulator
  - Customizable smart space datasets using models of entities in smart space ecosystems.
  - ML techniques to learn profiles of people and types of events from seed data.

- We presented a **structured methodology to evaluate the realism of synthetic data**.

- Our experiments show that SmartSPEC produces data that is **1.4x - 4.4x** more realistic than baselines.

- The SmartSPEC approach can also be employed to generate synthetic sensor data.

- **Our code is publicly available on GitHub**: [https://github.com/andrewgchio/SmartSPEC](https://github.com/andrewgchio/SmartSPEC)