Budapest University of Technology and Economics
Department of Measurement and Information Systems

SUMO User Conference 2024.

On Vehicular Data Aggregation in Federated Learning – Parking Simulation and Privacy

Levente Alekszejenkó – alelevente@mit.bme.hu
Tadeusz Dobrowiecki – dobrowiecki@mit.bme.hu

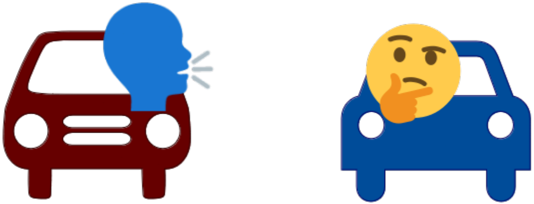
May 15. 2024

Introduction

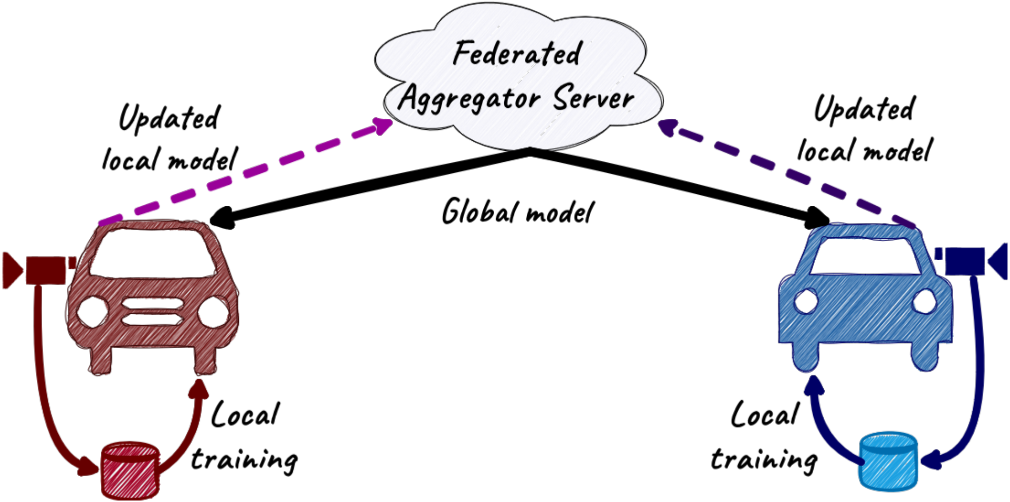
Vehicular Crowdsensing

Challenges of V2V crowdsensing:

- Communication channel usage?
- Heterogeneous sensors?
- Interoperability?
- Privacy?



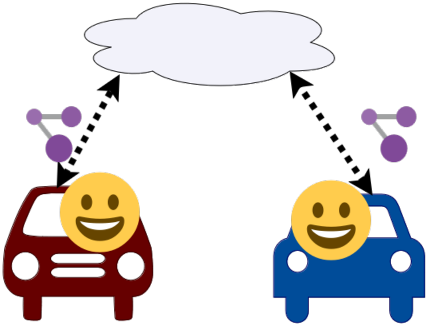
Vehicular Federated Learning



Vehicular Federated Learning

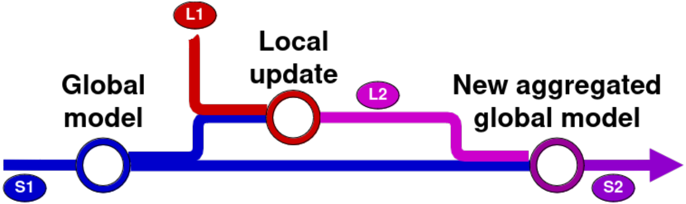
Solved challenges of V2V crowdsensing:

- Communication channel usage ✓
- Heterogeneous sensors ✓
- Interoperability ✓
- Privacy ✓ ?

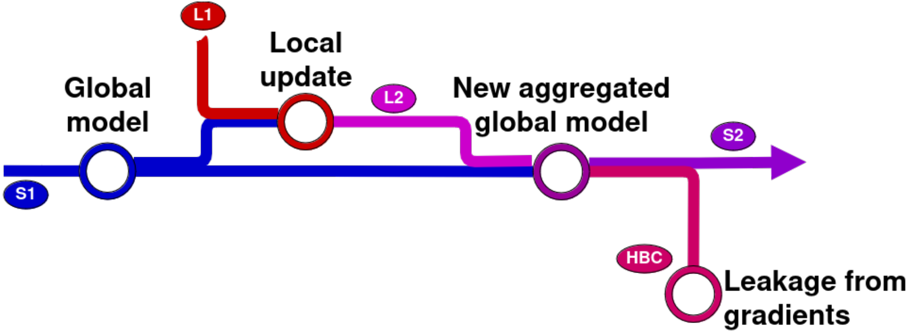


What about Privacy?

Federated Learning – Ideal Case



Federated Learning – Leakage from Gradients



A Case Study with SUMO

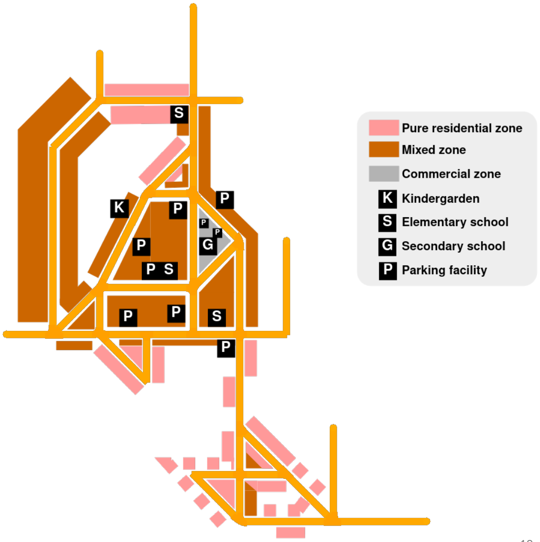
A Parking Monitoring System – Scenario

A generated small town:

- road network: `netgenerate`
- activities:
  - 10,000 inhabitants
  - 3,500 households
  - -200, +1000 person as commuter traffic
  - 10% uniform random background traffic
  - EU rural age distribution
  - `activitygen`

Parking lots:

- on-street parking on every street
- off-street parking facilities/garages (capacity: 1500 vehicles)



A Parking Monitoring System – Simulation in SUMO

`parking_activities`:

- a new Python tool (available on GitHub)
- enhances the output of `activitygen`
- **input**: `#days`, trip output file of `activitygen`
- **output**: a trip file

A Parking Monitoring System – Simulation in SUMO

The added `parking_activities`:

- **Random traffic:** no modification
- **Commuter traffic:**
  - enters and leaves the simulated network each day
  - repeats the original movements each day
  - adds a parking stop at the destination edge

A Parking Monitoring System – Simulation in SUMO

The added `parking_activities`:

- **Household traffic:**
  - does not leave the road network
  - concatenates the separate movements of a vehicle generated by `activitygen`
  - adds stops in parking lots between the original movements at the destination edge
  - in the morning 95% of the vehicles depart within a $\pm$15 minutes window (following a $\mathcal{N}(0,7)$ [minutes] distribution) compared to the original activity-chains

A Parking Monitoring System – Running the Simulation

Simulation scenarios:

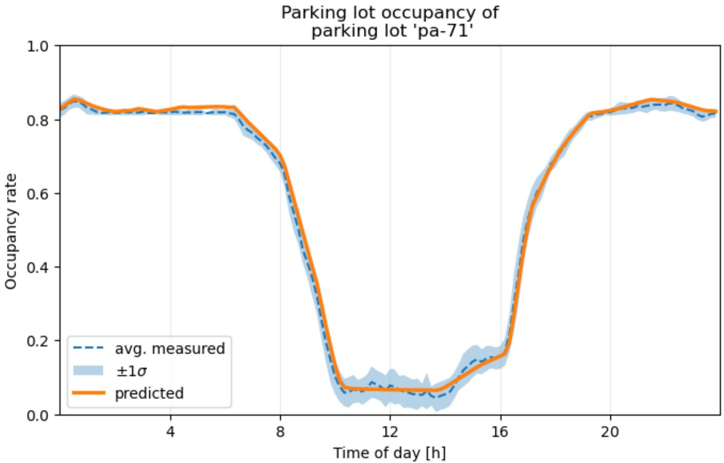
- **Burn-in:** 4 simulated days
- **Stable:** 5 simulated days

During simulations:

- vehicles measure the parking lot occupancy with a 50 m measurement range (via TraCI)
- `ParkingAreaRerouters` handles the potential overdemand for parking lots

A Parking Monitoring System – An Example

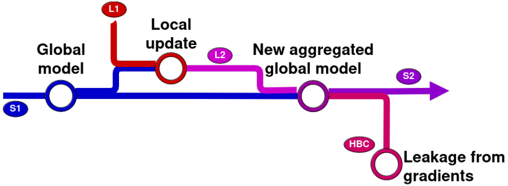
$\widehat{\mathcal{O}}(p, t)$: occupancy estimate for p parking lot at t time



Privacy Leakage

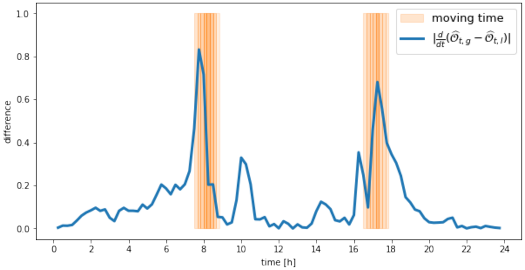
A Simple Tracking Attack

1. **Compute:**
   $$\Delta = (\widehat{\mathcal{O}}_{\text{global}}(p,t) - \widehat{\mathcal{O}}_{\text{local}}(p,t))^2$$
2. **Location inference:**
   2.1 Average Δ over time
   2.2 Select the locations of the resulted top 10 differences

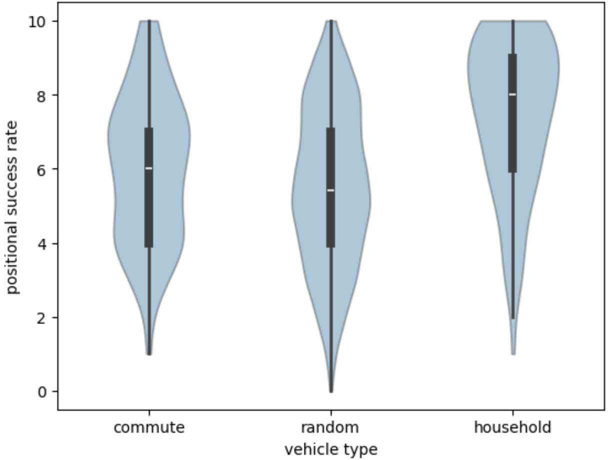


A Simple Tracking Attack

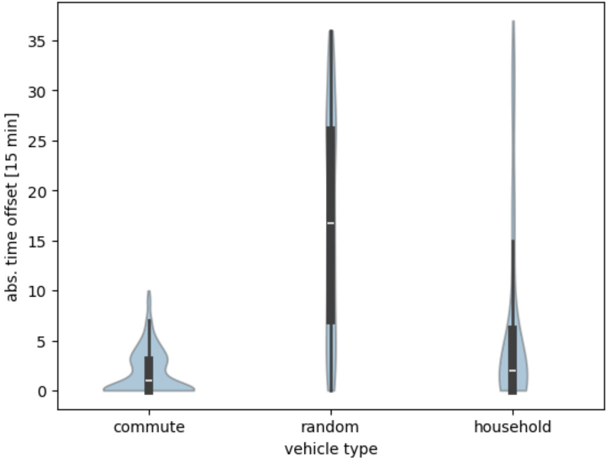
1. **Compute:**
   $\Delta = (\widehat{\mathcal{O}}_{\text{global}}(p,t) - \widehat{\mathcal{O}}_{\text{local}}(p,t))^2$
2. **Moving time inference:**
   2.1 Resample Δ with a $w = 15$ minutes window and average over parking lots $\rightarrow \widehat{\mathcal{O}_t}$
   2.2 Compute $|\frac{d}{dt}\widehat{\mathcal{O}_t}|$
   2.3 Select $\arg\max |\frac{d}{dt}\widehat{\mathcal{O}_t}|$



Results – Location

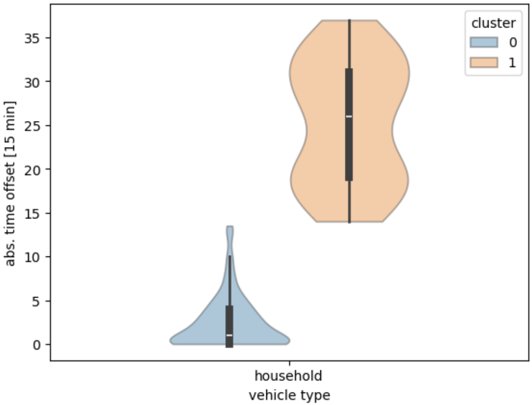


Results – Moving Time



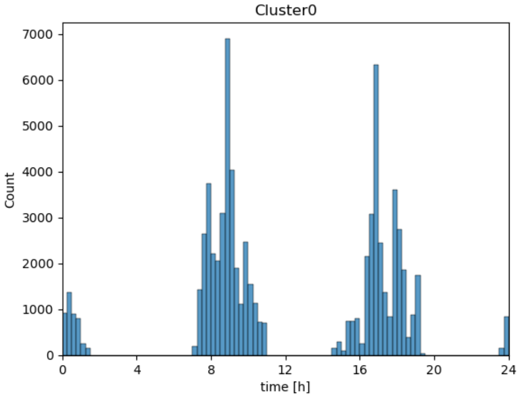
Results – Household Moving Time

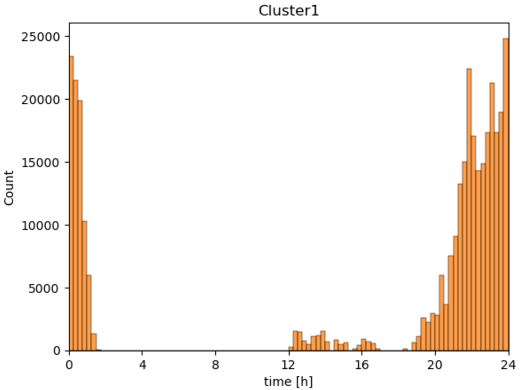
Household type vehicles separated into clusters:



Results – Household Moving Time

Household type vehicles separated into clusters:





Conclusion

Conclusion

1. SUMO parking activity simulation:
   - `parking_activities` tool
   - burn-in and stable phases
2. Vanilla vehicular federated learning leaks private information:
   - Household vehicles: 80% accuracy in position, $\pm$ 30 minutes in time!
   - Commuter vehicles: 60% accuracy in position, $\pm$ 30 minutes in time!
   - Random vehicles: 50% accuracy in position, inaccurate in time ✓

Questions?