Enhancing Trajectory Recovery from Gradients via Mobility Prior

Kaiyue Zhang[1,2], Zipei Fan[3,4], Xuan Song[2,4], Shui Yu[1]

[1] University of Technology Sydney, Sydney, Australia

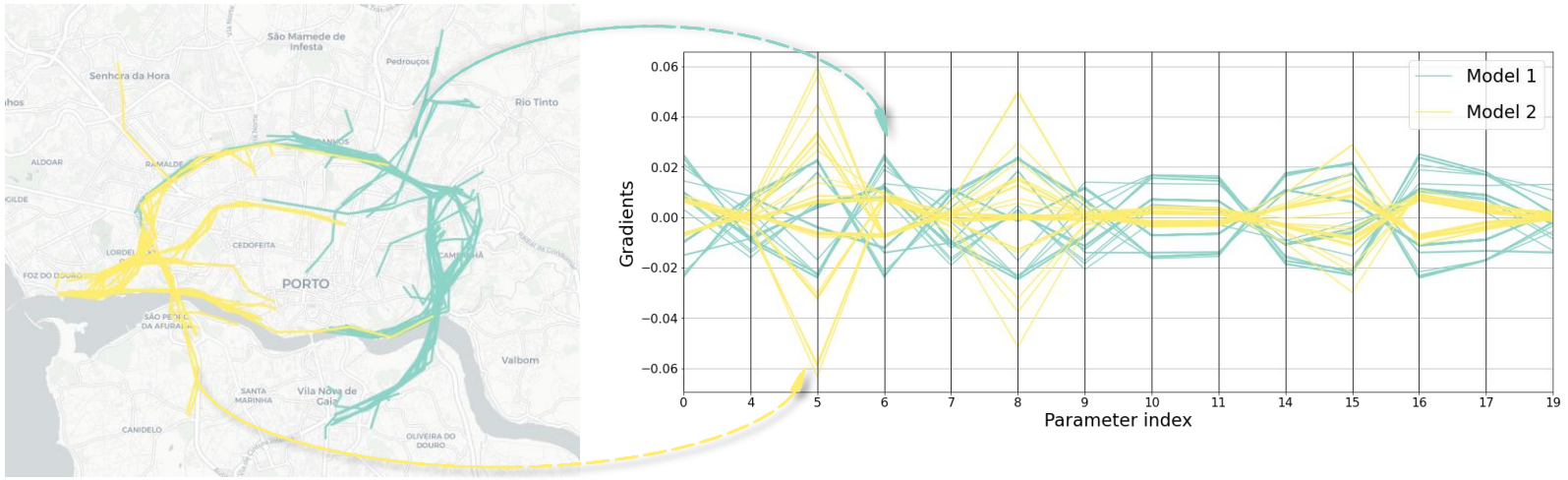
[2] Southern University of Science and Technology, Shenzhen, China

[3] The University of Tokyo, Center for Spatial Information Science, Kashiwa, Japan

[4] SUSTech-UTokyo Joint Research Center on Super Smart City, Southern University of Science and Technology, Shenzhen, China

Background

- The gradients can reveal privacy information in federated learning systems.
- For more complex recurrent neural network model, the existing attack algorithms *Deep Leakage from Gradients* (DLG)[1] are limited.



The parallel coordinate plot of weights from two different models, distinguished in two colors.

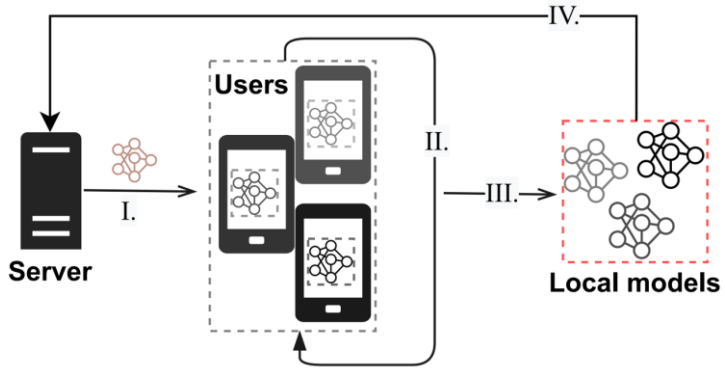
Recovery result of DLG in trajectory model.

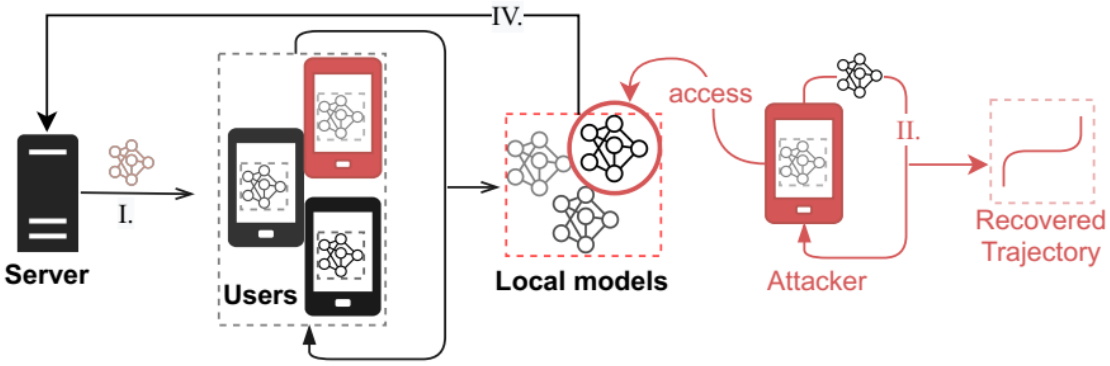
2022 Australian Computer Science Week

[1] *L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in Advances in Neural Information Processing Systems, 2019, pp. 14 774–14 784.*

Methods

- Unlike the traditional federated learning process shown below, the attacker of DLG algorithm will do reverse optimization to train the input data and finally get the recovered trajectory as shown in the right.





- Details of our proposed attack method *Deep Leakage from Gradients with Mobility Priors* (DLGMP)
  - Initial Search Stage
    - With the prior knowledge of the input and label, the attacker can search for a more appropriate dummy input and output convincingly, rather than completely randomly. (line1~3)
  - Regularization Term
    - With the prior knowledge of speed limit requirements, the attacker can further avoid the recovered trajectory from too long or too short. (line10, line12)
  - Adversarial Loss
    - We finally add the adversarial loss of a well trained *discriminator* of Wasserstein GAN (WGAN) to recover a reasonable shape. (line11, line 12)

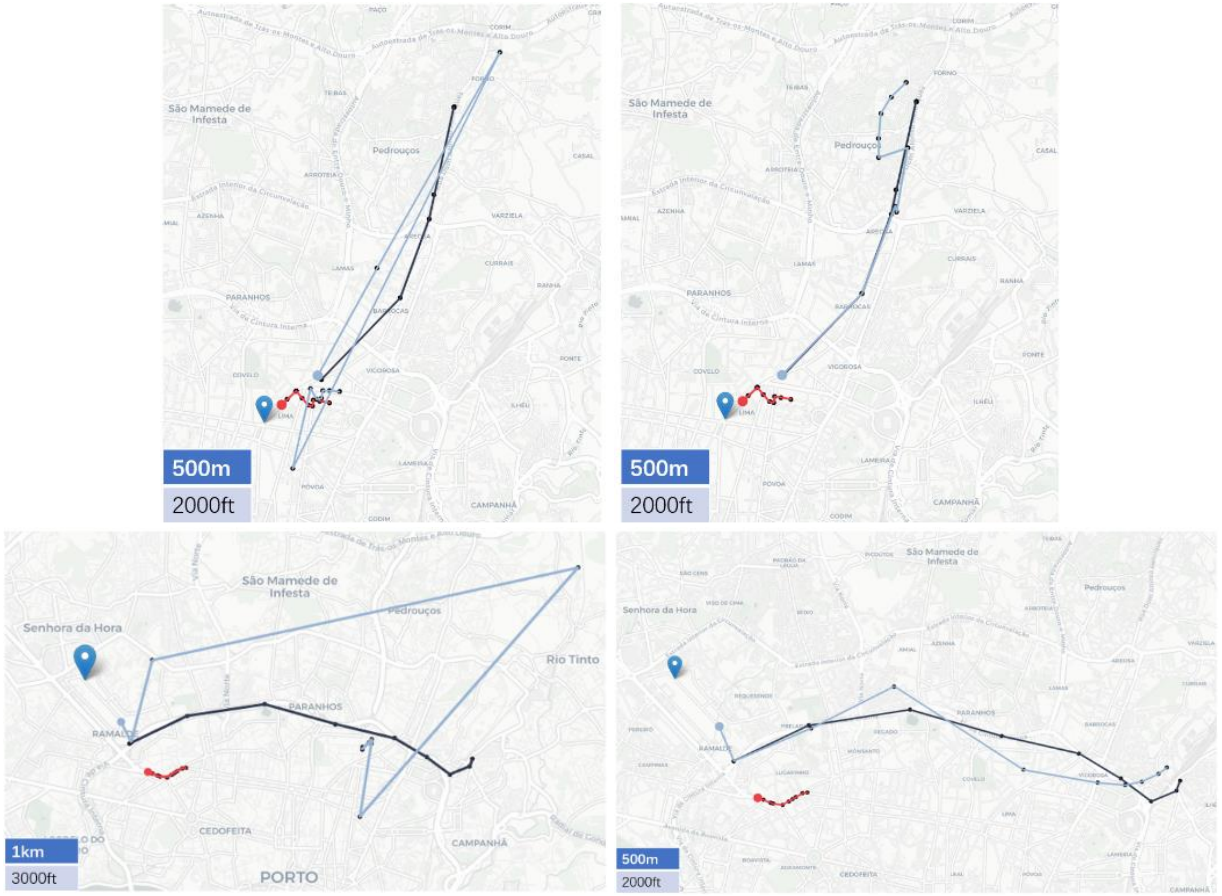
Algorithm 3 Deep Leakage from Gradients with Mobility Priors

Input: W: global model parameters; ∇W_k: parameters of the attacked model k; X, Y: the attacker's trajectory and destination set

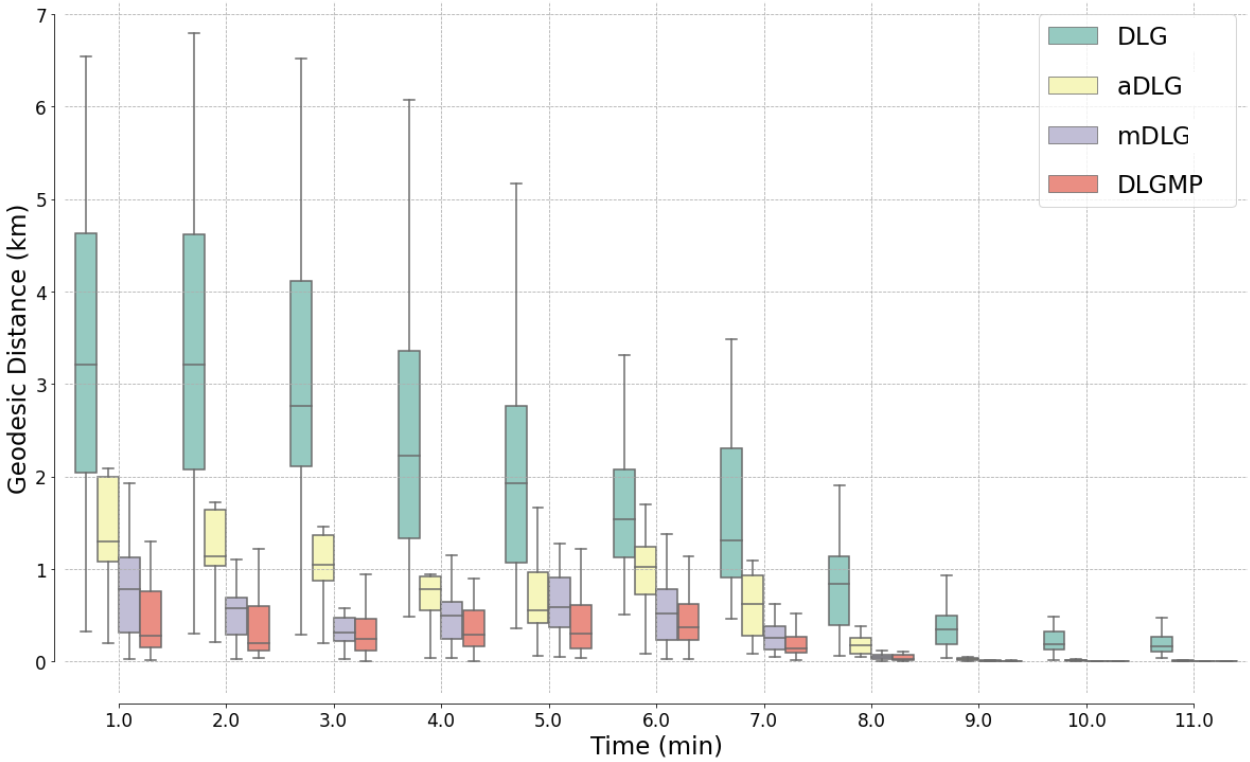
Output: recovered trajectory input and destination: x^*, y^*

1: // Search for best-match trajectory, narrow the search space to $\mathbb{S}$
2: $x^*, y^* \leftarrow \arg\min_{x,y} \| \frac{\partial \ell(F(x,W),y)}{\partial W} - \nabla W_k \|$;
3: // Set x^*, y^* as dummy input and label.
4: **for** $i = 1$ to n **do**
5: **for** $j = 1$ to n' **do**
6: $\ell_j^d \leftarrow \mathbb{E}_{x \sim X \setminus \{x^*\}}[logD(x)] - logD(x_i^*)$
7: Update Discriminator D with ℓ_j^d
8: **end for**
9: $\nabla W_i^* \leftarrow \frac{\partial \ell(F(x_i^*,W),y_i^*)}{\partial W}$
10: $\delta_i \leftarrow ReLU(v_{instant} - v_{max}) + ReLU(v_{min} - v_{avg})$
11: $\ell_i^g \leftarrow logD(x_i^*)$
12: $\mathbb{D}_i \leftarrow \delta_i + \ell_i^g + \|\nabla W_i^* - \nabla W_k\|^2$;
13: $x_{i+1}^* \leftarrow x_i^* - \eta \nabla_{x_i^*} \mathbb{D}_i, y_{i+1}^* \leftarrow y_i^* - \eta \nabla_{y_i^*} \mathbb{D}_i$;
14: **end for**
15: **return** x_{n+1}^*, y_{n+1}^*

Experiments



The comparison of recovery result between the existing attack algorithm DLG(left) and DLGMP (right). The *black line and blue pin*: attacked trajectory and its destination; *red line*: the original dummy trajectory; *blue line*: the final recovery result.



The geodesic distance between each time in the recovered trajectory and the corresponding time in the attacked trajectory. The results of different algorithms are shown in different colors. Since for the time-series data such as vehicle trajectories, the historical state has a certain influence on the prediction on the one hand, and on the other hand, the longer the memory is, the easier it is to be forgotten. The earlier information will give the less influence the parameter changes of the model, while the addition of prior knowledge can effectively alleviate this phenomenon.