FL-AMM: Federated Learning Augmented Map Matching With Heterogeneous Cellular Moving Trajectories

Huali Lu, *Graduate Student Member, IEEE*, Feng Lyu, *Senior Member, IEEE*, Huaqing Wu, *Member, IEEE*, Jie Zhang, *Student Member, IEEE*, Ju Ren, *Senior Member, IEEE*, Yaoxue Zhang, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— Map matching is a fundamental component for location-based services (LBSs), such as vehicle mobility analysis, navigation services, traffic scheduling, etc. In this paper, we investigate federated learning augmented map matching based on heterogeneous cellular moving trajectories from different operator systems, the goal of which is to improve matching accuracy without violating the user privacy. First, we develop a data collection platform with one Android-based application, and conduct rigorous data collection campaigns. Second, we perform systematic data analytics to reveal the data-driven technical challenges, including the impact of sampling rate, high location error of cellular moving data, and poor heterogeneous matching performance. Third, we propose an augmented map matching model, named *FL-AMM*, i.e., Federated Learning Augmented Map Matching, in which we i) adopt the vertical federated learning framework to achieve data collaboration and privacy protection for heterogeneous operators; ii) devise a data augmentation component to enhance the capability of representing the raw cellular data; and iii) design a map matching model to further learn the mapping function from cellular trajectory points to road segments. Finally, we conduct extensive data-driven experiments to corroborate the efficiency and robustness of the proposed *FL-AMM*.**

***Index Terms*— Vertical federated learning, map matching augmentation, heterogeneous trajectory fusion, stacked bidirectional gated recurrent unit, attention mechanism.**

I. Introduction

As ONE of the most important impetuses for the construction of smart city, location-based services (LBSs) have been widely exploited in intelligent transportation systems [1], [2], social networking services [3], [4], [5], location-based advertising [6], tracking systems [7], [8], navigation services [9], [10], etc. However, LBSs rely on mobile users' location information, which can be affected severely by imprecise and uncertain raw user positioning information. Hence, serving as a fundamental component for LBS, map matching is proposed to snap raw imprecise global positioning system (GPS) or cellular moving trajectories into the road networks to identify the true moving paths and true positions of moving objects [11]. Map matching is playing an increasingly important role in trajectory-based applications, such as vehicle mobility analysis [12], [13], traffic scheduling [14], [15], [16], [17], [18], route optimization [19], [20], etc.

Most conventional map matching approaches are based on GPS trajectories due to their relatively high data quality, which can achieve satisfactory matching accuracy. However, GPS data suffers from some inherent defects including intermittent availability, small coverage, extra data collection cost, and high risk of privacy disclosure. Therefore, GPS data has limitations to support large-scale map matching tasks, such as city-wide epidemiological survey in the pandemic era. On the contrary, cellular moving trajectories have significant advantages of continuous availability, complete coverage, and no additional cost due to the passive collection mode. Therefore, cellular trajectory data holds great potential to enable extensive map matching and related researches. However, cellular data-based map matching has not been widely studied in the literature due to its low data quality. Although there have been some studies carried out based on cellular trajectories, they usually achieve imprecise matching accuracy with data provided by one single operator. Therefore, an augmented map matching

Manuscript received 12 January 2022; revised 19 June 2023; accepted 13 August 2023. Date of publication 9 October 2023; date of current version 22 November 2023. This work was supported in part by the National Key Research and Development (K&D) Program of China under Grant 2022YFC2009805 and Grant 2022YFF0604504; in part by the National Natural Science Foundation of China under Grant 62320106006, Grant 62002389, and Grant 92067206; in part by the 111 Project under Grant B18059; in part by the Young Elite Scientist Sponsorship Program by the China Association for Science and Technology (CAST) under Grant YESS20200238; in part by the Central South University lnnovafion-Driven Research Program under Grant 2023CXQD029; in part by the K&D Program of Hunan Province of China under Grant 2022GK2013; in part by the Natural Science Foundation of Hunan Province of China under Grant 2021JJ20079; in part by the Young Talents Plan of Hunan Province of China under Grant 2021RC3004; in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2023-03759; and in part by the University of Calgary Start-Up under Grant 10040064. *(Corresponding author: Feng Lyu.)*

Huali Lu, Feng Lyu, and Jie Zhang are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: huali_lu@csu.edu.cn; fenglyu@csu.edu.cn; jie_zhang@csu.edu.cn).

Huaqing Wu is with the Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: huaqing.wu1@ucalgary.ca).

Ju Ren and Yaoxue Zhang are with the Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China (e-mail: renju@tsinghua.edu.cn; zhangyx@tsinghua.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2023.3322841.

Digital Object Identifier 10.1109/JSAC.2023.3322841

0733-8716 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

algorithm is required to provide fast and precise processing of large-scale cellular moving trajectories. Considering that the cellular moving trajectories provided by one single operator are sparse and noisy containing scant information about the underlying driving path, we propose to aggregate data from multiple operators to collaboratively improve the accuracy of the map matching algorithm.

However, it is quite challenging to conduct the aforementioned map matching model, which has the following three rigid hurdles to conquer. First, heterogeneous cellular moving trajectories provided by different operators are physically data isolated. Therefore, how to protect the data privacy of each participating operator is the first challenge to be tackled when utilizing heterogeneous cellular moving trajectories collaboratively to enhance the performance of map matching. Second, compared to GPS data, cellular moving trajectories can be of poor quality with low frequency sampling and high location noises, which may lead to some unexpected map matching errors. Third, the design of map matching model is difficult because traditional model-driven approaches or data-driven approaches, such as hidden Markov model (HMM) or recurrent neural network (RNN) models, are not suitable for large-scale and cellular data-based map matching tasks.

Prior efforts in the literature to explore map matching mainly fall into two categories. (1) *GPS data-based solutions.* In this field, most studies have focused on model-driven solutions with high-quality GPS moving trajectories, e.g., HMM-based solutions [21], [22], [23], maximum weights model [24], local path inference model [25], conditional random field model [26], etc. In addition, some data-driven map matching models [11], [27], [28] have been proposed for some scenes with a low sampling rate of GPS moving trajectories. For example, Feng et al. [27] proposed DeepMM to solve the map matching task by a deep learning approach. Nevertheless, these well-designed approaches fail to resolve the large-scale map matching tasks due to small coverage of GPS devices. Meanwhile, they cannot be directly applied to cellular moving trajectories due to the inherent coarse-grained attributes of cellular data. (2) *Cellular data-based solutions.* In this regard, there have been a few studies [29], [30], [31] investigating it. Most of existing researches have only focused on employing cellular moving trajectories data provided by one single operator to address map matching problems. For instance, Shen et al. [29] designed a RNN-based model called DMM to map cellular moving trajectories into road segments. However, the matching accuracies of these schemes are not satisfactory enough, and no study exists that adequately considers improving the map matching accuracy by heterogeneous cellular moving trajectories.

In this paper, we investigate federated learning augmented map matching based on heterogeneous cellular moving trajectories from different operator systems, with the goal to improve matching accuracy without violating the user privacy. We propose to train the map matching model by utilizing heterogeneous cellular moving trajectories provided by different operators, while the model can be used for map matching inference in general for both single-SIM-card and dual-SIM-card users.[1] Particularly, we first define the map matching problem based on cellular moving trajectories under the vertical federated learning framework, and illustrate the motivation of investigating the new map matching scheme for cellular moving trajectories. To substantiate the research, we then develop a data collection platform, consisting of one Android-based App (application) and 20 mobile phones, and conduct rigorous data collection campaigns in our city covering different road types and different transportation modes. With the collected dataset, we perform a systematic data analytics to disclose the data-driven technical challenges in terms of the negative correlation between matching accuracy and sampling rate, high location error of cellular moving trajectory, and matching difference among different operators. Inspired by the data-driven insights, we then propose a federated learning augmented map matching model for heterogeneous cellular moving trajectories, named *FL-AMM*, i.e., Federated Learning Augmented Map Matching. *FL-AMM* contains three major technical components, i.e., vertical federated learning (VFL), data augmentation (DA), and map matching model (MM). In *FL-AMM*, we i) construct a vertical federated learning framework to protect the data privacy of each participating operators when using heterogeneous cellular moving trajectories to learn the map matching model synergistically; ii) devise a data augmentation component that integrates sub-modules of data denoising, trajectory division and trajectory representation enhancement to enrich the amount of training data and enhance the expression capability of representing raw poor-quality cellular moving trajectories; iii) design a map matching model with two sub-modules, i.e., a stacked bidirectional gated recurrent unit (SBi-GRU) encoder and an attentional GRU decoder to learn the mapping function projecting cellular moving trajectories to road segment-based trajectories. Finally, we conduct extensive data-driven experiments to demonstrate the efficacy of *FL-AMM*, which can achieve an average precision and recall of 89% and 89%, respectively, outperforming the benchmarks significantly. In addition, it can also work robustly under other transportation modes of bus and subway, with an average F1-score of 77% and 81%, respectively.

The major contributions are summarized as follows.

- This is the first attempt to address map matching problem via aggregating heterogeneous cellular moving trajectories and protecting data privacy of each participating operator with federated learning. The proposed method can be adaptive with different types of inputs for inference, e.g., single-SIM-card trajectory and dual-SIM-card trajectories, and the map matching performance can increase with the amount of heterogeneous trajectories.
- To investigate the map matching problem, we first develop a data collection platform and collect sufficient heterogeneous cellular moving traces. We then conduct systematic analytics on these collected trajectories, and disclose the data-driven challenges, which can direct our solution design and implementation empirically.

[1] Dual-SIM-card users are those who have two SIM cards of different operators in their mobile phones simultaneously. Therefore, they have two heterogeneous cellular moving trajectories which can be used collaboratively, to improve the map matching accuracy.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

- We propose a federated learning augmented map matching model, named *FL-AMM*, for efficient map matching. In *FL-AMM*, we improve the map matching accuracy from three aspects, i.e., collaboration and privacy protection of heterogeneous data, data augmentation of raw poor-quality data, and innovation of map matching model. Extensive experiments are carried out to demonstrate both the efficiency and robustness of *FL-AMM*.

The remainder of this paper is organized as follows. We first describe the problem definition in Section II. The data collection is given in Section III. Then, we conduct an empirical data analytics in Section IV, followed by the design of *FL-AMM* in Section V. Extensive experiments are conducted in Section VI, and the related work is reviewed in Section VII. Finally, we conclude the paper in Section VIII.

II. Problem Definition

In this section, we start with definitions related to map matching problem based on user cellular moving trajectories under vertical federated learning framework. Then, we investigate the limitations of existing map matching solutions and the necessity of a novel map matching scheme.

A. Problem Definition

Definition 1 (Cellular Moving Trajectory): A cellular moving trajectory is a sequence of BSs that are accessed by one SIM card i on the move, denoted by $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k}, \ldots\}$, where $p_{i,k}$ is the k-th associated BS, containing the information of $(CID_{i,k}, t_{i,k})$ with the BS ID $CID_{i,k}$ at the timestamp of $t_{i,k}$. Each CID corresponds to one unique physical location (lon, lat) with lon and lat standing for longitude and latitude, respectively.

Definition 2 (Road Topology): Road topology can be described as a directed graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_k, \ldots\}$ is a set of nodes on the road topology, representing intersections or terminal points, and $E = \{e_1, e_2, \ldots, e_k, \ldots\}$ is a set of road segments connecting nodes v in V. In this study, the road topology is obtained from a public open-source website (OpenStreetMap [32]).

Definition 3 (Vertical Federated Learning Framework): We consider a two-party[2] VFL setting. Specifically, party A has dataset $\mathcal{D}^A := \left\{\left(X^{A,i}, L^{A,i}\right)\right\}_{i=1}^{n^A}$, where $X^{A,i}$ is the feature vector of the i-th sample and $L^{A,i}$ is the corresponding ground-truth label, while party B has dataset $\mathcal{D}^B := \left\{\left(X^{B,i}, L^{B,i}\right)\right\}_{i=1}^{n^B}$. $\mathcal{D}^A$ and $\mathcal{D}^B$ are held privately by the two parties and cannot be exposed to each other. n^A and n^B are numbers of samples of $\mathcal{D}^A$ and $\mathcal{D}^B$, respectively. $\mathcal{D} = \mathcal{D}^A \cap \mathcal{D}^B$ denotes the overlapping samples of $\mathcal{D}^A$ and $\mathcal{D}^B$. VFL aims to train a global model based on $\mathcal{D}$ from both parties collaboratively without data privacy leakage.

Definition 4 (Map Matching under VFL): Map matching is the process of projecting a raw trajectory P^r onto the road topology to get a matched trajectory Y^m. In our study, $P^r_{C_A} = \{p^r_{C_A,1}, p^r_{C_A,2}, \ldots, p^r_{C_A,k}, \ldots\}$ and $P^r_{C_B} = \{p^r_{C_B,1}, p^r_{C_B,2}, \ldots, p^r_{C_B,k}, \ldots\}$ denote a pair of heterogeneous raw trajectories collected by OP A and OP B, respectively. A matched trajectory Y^m is a segment-based trajectory, representing by $Y^m = \{e_1, e_2, \ldots, e_k, \ldots\}$. Map Matching under VFL aims to learn a function $F(.)$ transforming $P^r_{C_A}$ and $P^r_{C_B}$ collaboratively to Y^m in a privacy-preserving manner, i.e., $Y^m = F(P^r_{C_A}, P^r_{C_B})$.

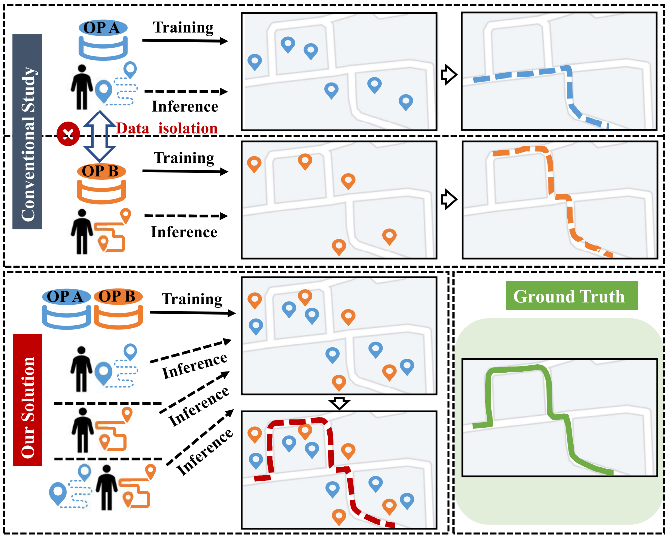


Fig. 1. Illustration of existing map matching solutions.

B. Motivation

In this study, we focus on map matching based on cellular moving trajectories instead of GPS trajectories. To our best knowledge, existing cellular data-based map matching studies only work with datasets provided by a single operator due to the considerations of physical data isolation and privacy security. However, unlike the GPS trajectories that can be collected with a fine collection granularity, the cellular moving trajectories are quite sparse with high location errors constrained by the deployment density of base stations (BSs). Therefore, most of works based on single-source cellular moving trajectories cannot achieve a satisfactory performance, as shown in Fig. 1. Particularly, OP A and OP B train their own map matching models by cellular data of themselves. However, the accuracies of these map matching models are not ideal enough, because the data of OP A and OP B only contain partial and scant information on the real path. In addition, due to the different BS deployment policies in overlapping areas, the cellular moving trajectories collected by different operators are usually complementary. Inspired by these investigations, in this work, we aim to enhance map matching accuracy by training the model with heterogeneous cellular moving trajectories provided by different operators. Specifically, as shown in Fig. 1, in our solution, in order to capture more abundant information about real moving paths, we use heterogeneous cellular moving trajectories from OP A and OP B to train the map matching model collaboratively. That is, OP A and OP B work cooperatively as participants to train a more powerful global model than local models of

[2] For the convenience of illustration, the number of participants of VFL is set to be two in this paper, and the two participants are operator A (OP A) and operator B (OP B).

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

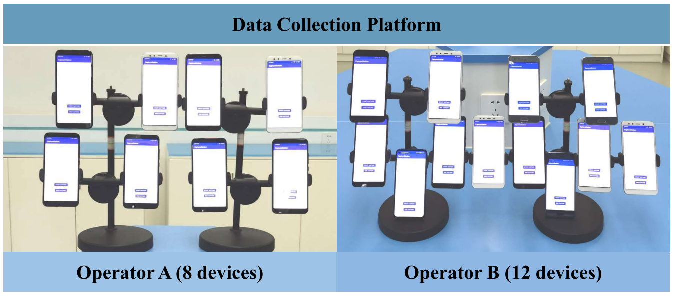
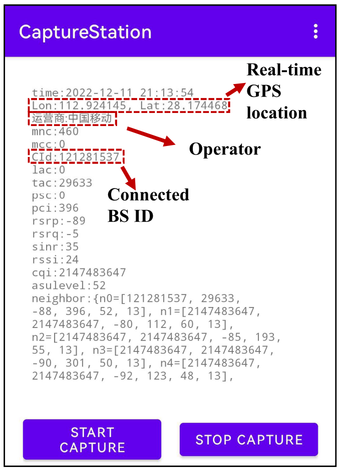


Fig. 2. Cellular data collection platform.



```
time:2022-12-11 21:13:54
Lon:112.924145, Lat:28.174468
运营商:中国移动
mnc:460
mcc:0
CId:121281537
lac:0
tac:29633
psc:0
pci:396
rsrp:-89
rsrq:-5
sinr:35
rssi:24
cqi:2147483647
asulevel:52
neighbor:{n0=[121281537, 29633,
-88, 396, 52, 13], n1=[2147483647,
2147483647, -80, 112, 60, 13],
n2=[2147483647, 2147483647, -85, 193,
55, 13], n3=[2147483647, 2147483647,
-90, 301, 50, 13], n4=[2147483647,
2147483647, -92, 123, 48, 13],
```

Fig. 3. Cellular data collection APP.

themselves, which can extract the unique features contained in both OP A and OP B to achieve higher map matching accuracy. When the global map matching model is trained, it can be deployed in different parties. First, it can be deployed in every participating operator locally to conduct map matching inference for its individual trajectories, i.e., single-SIM-card users. Second, it can be deployed in a trustworthy server for map matching inference of those users who own more than one cellular moving trajectories simultaneously, i.e., dual-SIM-card users. Note that, the matching accuracy in the second case is significantly better than that in the first case.

III. Data Collection

In this section, we first introduce the data collection platform, and then describe our data collection campaigns.

A. Data Collection Platform

Though cellular data is available at operators, due to privacy/security issues, it is not open to the public. To substantiate our research, we first design and implement a cellular data collection platform, which contains one real-time Android-based data collection App and a number of mobile phones, to capture sufficient research data between moving objects and BSs.

Fig. 2 shows the specific data collection platform we employed, consisting of 20 mobile phones, among which 8 devices are equipped with OP A SIM cards, and the other 12 devices are equipped with OP B SIM cards. Every device is installed with a real-time data collection App to collect the cellular moving trajectories. Specifically, the App is designed and implemented under Android 12, where Android.Telephony API is leveraged to collect the user network association information, Android.Location API is used to obtain the user position information, and IO API is used to store the data locally. The real-time running process of data collection App is shown in Fig. 3. The main fields of which are listed in Table I. Particularly, Table I shows all the data that can be collected by our self-designed App, which can be used to support other applications related to cellular data. In this work, to make our proposed method compatible to all cellular networks with generalization, we use only three fields that are passively collected by all OPs without violating user privacy. The first field, `Operator`, indicates the data source, i.e., the mobile network operator. The second field, `CID`, is the connected BS ID, standing for cell identification. The connected BS sequences form moving trajectories. A sample is collected if the moving object drops the original connection and re-associates to a new BS. The third field, `Lon` and `Lat`, includes the real-time GPS locations for position references and route labels, which are sampled per second. Therefore, our manually-collected data on the user equipment side can accurately reflect the data that the mobile network operator would have. However, considering the need for ground-truths in map matching, we use our own collected data rather than the operator data.

TABLE I
Data Collection Fields

Field	Value
Operator	OP A
MNC	460
CID	121281557
TAC	29633
PCI	398
RSRP	-67
RSRQ	-3
SINR	42
RSSI	37
CQI	2147483647
Neighbors	n_0=[97014856, -85, 301, 55, 13], …
Time	2022-05-20 12:31:36
Lon	112.92408595072482
Lat	28.17452090619502

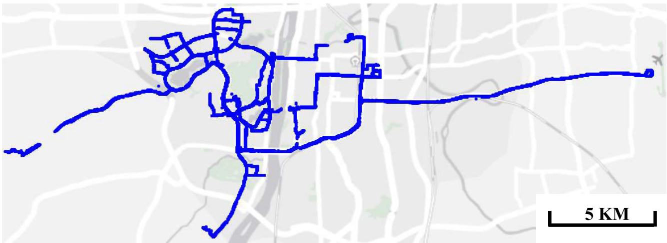


Fig. 4. Coverage map of our collected dataset.

B. Data Collection Campaigns

We recruited a group of volunteers for data collection using our designed real-time data collection App. All the volunteers gave their consents to participate in the experiments and use

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

their data for study. In order to mitigate the adverse effects caused by the limited data scale, we carefully design our data collection campaigns to increase the diversity of collected data. Specifically, we consider three major road types with differentiated BS densities and driving conditions, i.e., urban, suburban, and highway, and three transportation modes, i.e., car, bus, and subway, respectively. To obtain heterogeneous cellular moving trajectories, in each trip, volunteers were equipped with multiple mobile phones which were equipped with different OP SIM cards. The blue lines in Fig. 4 show a coverage map of the collected dataset. Particularly, for the car transportation, we have totally 234 user traces, consisting of 106 user traces from OP A and 128 user traces from OP B. The dataset has an overall time length of 210.9 hours and a total travel distance of 6,203.9 kilometers with 2882 BSs.

IV. Insights of Data Analytics

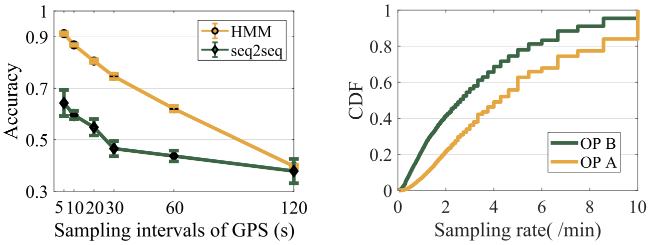
In this section, we conduct data analytics on the cellular moving trajectory and highlight the technical challenges, which direct our implementation solutions.

A. Impact of Sampling Rate

The sampling rate of data has an important impact on the accuracy of map matching for most of the existing map matching algorithms. In other words, the map matching accuracy degrades significantly as the sampling time interval increases. We reveal this phenomenon through the performance of two typical models, i.e., HMM (model-driven) and seq2seq model (data-driven), on GPS trajectories with different sampling intervals, as shown in Fig. 5a. We select 6 different sampling intervals, i.e., 5s, 10s, 20s, 30s, 60s, 120s, to test the map matching accuracy of HMM and seq2seq, respectively. From Fig. 5a, we can have the following three major observations. First, with the increase of sampling interval, the matching accuracy of both HMM and seq2seq decreases rapidly, especially for HMM. Second, HMM outperforms seq2seq for fine-grained GPS trajectories with a lower error bar. Third, seq2seq is more stable than HMM. When the sampling interval increases to a certain value, the performance of seq2seq will surpass that of HMM. Then, we investigate the sampling rates of cellular moving trajectories, and plot the cumulative distribution function (CDF) results in Fig. 5b. Since cellular moving trajectories are passively collected, we regard the number of connected BSs per minute as its sampling rate. We can observe that the medium sampling rate is just 4 and 2 samples per minute for OP A and B, respectively, resulting in significant sparsity for cellular moving trajectories. Based on observations on the GPS performance in Fig. 5a, it can be expected that the performance of traditional HMM and seq2seq algorithms on cellular moving trajectories will be unsatisfactory.

B. High Location Error of Cellular Moving Trajectory

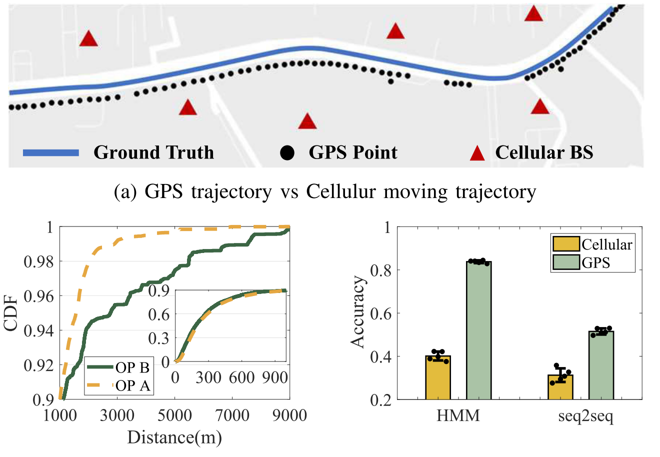
Unlike the GPS trajectories with high precision positioning, the cellular moving trajectories usually suffer from coarse location accuracy. We first visualize GPS and cellular moving trajectories of the same ground truth on the road map.



(a) Accuracy of sampling rates for GPS data

(b) CDF of sampling rate for cellular data

Fig. 5. Impact of sampling rate.



(a) GPS trajectory vs Cellulur moving trajectory

(b) CDF of location offset

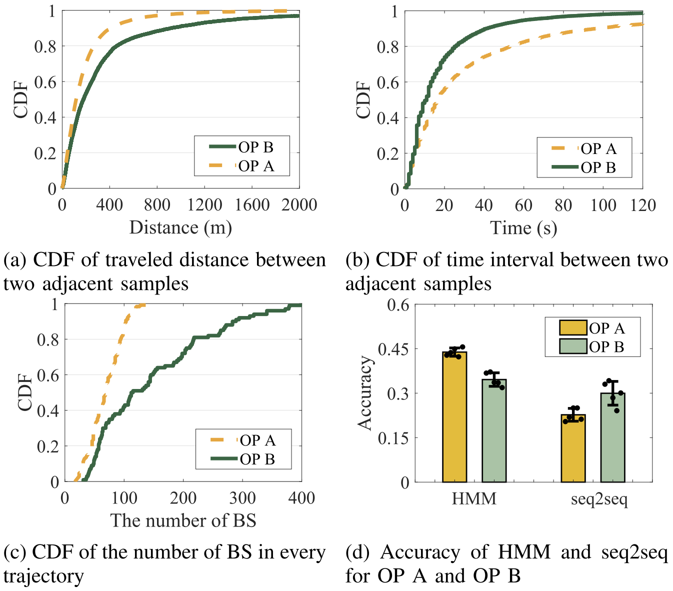
(c) Accuracy of HMM and seq2seq for GPS and cellular trajecories

Fig. 6. High location error of cellular moving trajectory.

As shown in Fig. 6a, the location errors of the cellular moving trajectory are far larger than that of GPS trajectory. The points of the cellular moving trajectory are scattered, while the points of the GPS trajectory are densely distributed along the truth route. Fig. 6b shows CDFs of the location offsets between the user GPS and associated BS positions. We can observe that about 40% of the location offsets are larger than 300 meters, and more than 10% of the location offsets can reach above 1,000 meters, distributed between 1,000 and 9,000 meters. Therefore, the cellular data trajectory is quite coarse to localize users efficiently. Then, we test the map matching performance of HMM and seq2seq for GPS trajectory and cellular moving trajectory with the same granularity sampling interval respectively. As depicted in Fig. 6c, we can observe that the matching accuracy for GPS trajectory is much better than that for cellular moving trajectory with smaller deviation.

C. Matching Difference With Operators

With different BS deployment policies and data collection mechanisms, the collected cellular trajectories of dual SIM cards have heterogeneous properties. In addition to the BS location differences, other heterogeneous features are shown in Fig. 7. Particularly, Fig. 7a shows the CDFs of traveled distance between two adjacent samples under two operator networks, which are calculated based on the two GPS locations of a user when the two adjacent samples are collected. First, the distributions of traveled distance are heterogeneous in different OPs with a clear distribution gap. For instance, about 80% of traveled distances in OP A are smaller than 300 meters,



(a) CDF of traveled distance between two adjacent samples

(b) CDF of time interval between two adjacent samples

(c) CDF of the number of BS in every trajectory

(d) Accuracy of HMM and seq2seq for OP A and OP B

Fig. 7. Matching difference vs. operator.

while this figure is about 60% in OP B. Fig. 7b shows the CDFs of time intervals between two adjacent samples, and the same phenomenon can be observed. Specifically, the time intervals are generally shorter than 120 seconds in both OPs, and about 80% of time intervals in OP B and OP A are shorter than 25 seconds, and 55 seconds, respectively. This is because the BS density of OP B is quite larger than that of OP A, resulting in more frequent BS handovers. Therefore, even with the same route, the number of trajectory points collected by different operators also varies greatly, as shown in Fig. 7c. Due to the heterogeneity of data from different operators, the performance of map matching will also have great distinction. Fig. 7d shows the performance of HMM and seq2seq for different operators. We can observe that different operators perform differently on the same model and should select different models to achieve a good map matching performance. Therefore, how to design a general model that can effectively handle heterogeneous moving trajectories from different operators while achieving excellent performance is a crucial challenge to address.

D. Implementation Directions

Inspired by the above observations, we propose the following three approaches to solve the current challenges of map matching for cellular data.

1) Data Augmentation: The raw cellular moving trajectory has a very obvious sparsity, which leads to the low accuracy of map matching. This observation inspires us to enhance the expression capability of representing low-granularity trajectories. Our key idea is to transform the sparse trajectory into the dense trajectory to enrich its information about true paths, thus improving the accuracy of map matching. Based on this idea, we build the data enhancement module to augment the raw data.

2) Map Matching Model: Through the previous data analysis, we can find that relying only on the model-driven or the typical data-driven models can hardly obtain a satisfactory matching performance on cellular data, due to its high sparsity and location error. Therefore, a novel and efficient map matching algorithm is necessary for providing high-accuracy map matching for cellular moving trajectories. Inspired by this, we devise a novel map matching model module which is an end-to-end deep learning model that can simultaneously capture historical and future features about the trajectory, and utilize the attentional encoder-decoder model to realize accurate mapping of cellular trajectories to road segments.

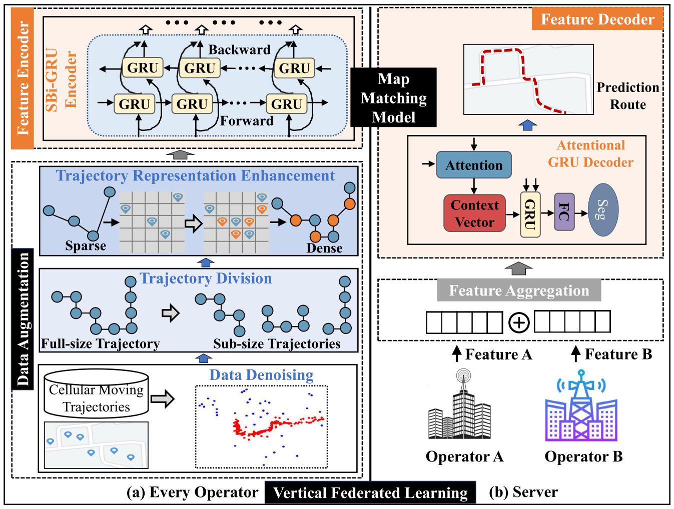


Fig. 8. Design overview of *FL-AMM*.

3) Vertical Federated Learning Framework: In addition to the heterogeneity of cellular trajectory data from different operators, the data isolation among operators is another issue due to the existence of sensitive user privacy data. Therefore, the map matching algorithm trained by each operator alone is not ideal, and the matching accuracy varies greatly among different operators. Therefore, how to design a general and enhanced map matching model for different operators is a challenge. In our map matching problem with multi-operator collaborative computing, the data is related to road segments and the features extracted from different cellular trajectories are complementary for characterizing road segments. Inspired by this, we propose to employ a vertical federated learning framework [33], [34] to train a powerful map matching model that can aggregate the data features from multiple operators to cooperatively improve the matching accuracy without data exchange among operators.

V. Design of *FL-AMM*

A. Overview

Figure 8 shows the overall architecture of *FL-AMM*, which is a vertical federated learning framework, consisting of two major modules, i.e., participants (operators) and a server. For every operator, it aims to extract local-specific features. In our *FL-AMM*, there are two major components for every operator, i.e., data augmentation and feature encoder. Particularly, considering the serious sparsity and location errors of cellular moving trajectories, we devise a data augmentation component integrating three sub-modules, i.e., data denoising, trajectory division and trajectory representation enhancement

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

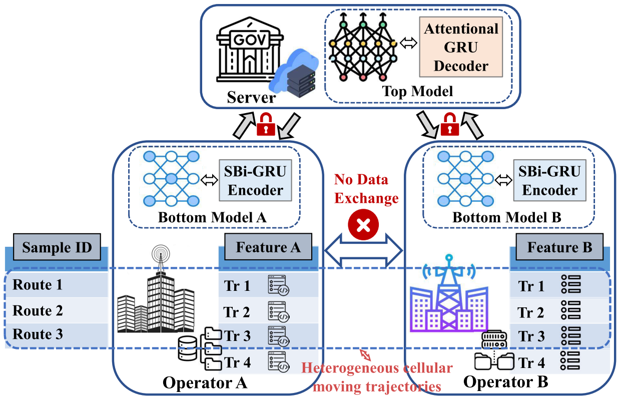


Fig. 9. An overall workflow for VFL.

for eliminating the side impacts of remote BSs and Ping-Pong effects, rising labeled training data, and enhancing the latent representation of sparse cellular trajectory. Then, the processed local trajectories are fed into the feature encoder to extract the hidden features. For the server, it integrates the features from each operator to build a powerful map matching model. In *FL-AMM*, the server is composed of two major components, i.e., feature aggregation and feature decoder. Specifically, feature aggregation aims to aggregate features provided by different operators. Then, we devise a feature encoder, to transform aggregated heterogeneous features into moving routes on the road network. Note that, feature encoder (i.e., SBi-GRU encoder) and feature decoder (i.e., attentional GRU decoder) form a complete map matching model. In what follows, we will elaborate on each individual component.

B. Vertial Federated Learning Framework

The key idea of vertical federated learning is to enhance a learning model by utilizing the distributed data with various features [35], [36], [37]. Fig. 9 shows the detailed workflow of our VFL framework. Specifically, considering a two-party VFL setting, firstly, we need to find the common identifiers served by all operators to align the training data samples, that is, we need to find $\mathcal{D} = \mathcal{D}^A \cap \mathcal{D}^B$, the overlapped heterogeneous cellular moving trajectories for dataset $\mathcal{D}^A$ and $\mathcal{D}^B$ of OP A and OP B, respectively. It is worth noting that we employ a secure multi-party protocol, which only allows multiple operators to find out the common IDs available across their data [38]. Secondly, every operator trains its bottom model M_{bottom} (i.e., SBi-GRU encoder) locally to extract feature map F_{map} of its specific and non-shared cellular trajectory data $\mathcal{D}$, given by,

$$\begin{aligned} F_{map}^{A} &= \mathrm{M}_{\text{bottom}}^{\mathrm{A}}\left(P_{C_A}^{r}\right) \\ F_{map}^{B} &= \mathrm{M}_{\text{bottom}}^{\mathrm{B}}\left(P_{C_B}^{r}\right) \end{aligned} \tag{1}$$

where $P_{C_A}^{r} \in \mathcal{D}_A$, $P_{C_B}^{r} \in \mathcal{D}_B$. $\mathcal{D}_A$ and $\mathcal{D}_B$ denote the overlapped training data samples from OP A and OP B, respectively. Thirdly, each operator transmits its local feature map extracted from original trajectory data to the server. Then, the server aggregates feature maps from all operators and trains the top model M_{top} (i.e., attentional GRU decoder) to obtain map matching result Y^m, given by,

$$F_{map}^{agg} = F_{map}^{A} \oplus F_{map}^{B} \tag{2}$$

$$Y^m = \mathrm{M}_{\text{top}}^{\mathrm{A}}\left(F_{map}^{agg}\right) \tag{3}$$

where $\oplus$ represents aggregation operation which can be achieved by concatenation, dot, and multi-layer perceptron (MLP). In our implementation, one-layer MLP is adopted. Afterward, the server performs backward transmission, sending back the gradients to every operator. Finally, each operator updates its local model based on the local data and gradients from the server. After several iterations, the server can obtain an enhanced powerful model that aggregates the data information of all operators without data exchange among operators. Besides, the trained powerful model can be deployed locally in each operator for map matching inferences of single-SIM-card trajectories, or deployed in a secure third party for map matching inferences of dual-SIM-card trajectories.

C. Data Augmentation

To address the data sparsity and low matching accuracy of cellular moving trajectories, we design a data augmentation component with three sub-modules, i.e., data denoising, trajectory division and trajectory representation enhancement, detailed as follows.

1) Data Denoising: The raw cellular moving trajectories are far from accurate, one of the reasons being that the noises of remote BSs and Ping-Pong effects prevail in sequential BS trajectories. To overcome this problem, we conduct data denoising to eliminate the side impact of those noises. Particularly, we employ the density-based spatial clustering of applications with noise (DBSCAN) algorithm [39], which is skilled in outlier detection, to eliminate noises from remote BSs. According to the empirical study in Fig. 6b, more than 90% of associated distances for both operators are smaller than 1,000 meters. Therefore, we empirically consider it as the distance threshold. Thus, the associated BSs can be clustered into two sets, i.e., close BS sets in proximity to user's actual geographical locations and remote BSs that are far away from the routes as noise. As for the Ping-Pong effect, we keep only one back-and-forth transition of BSs. For example, only A -> B -> C is kept when Ping-Pong effect happened in A -> B -> A -> B -> C.

2) Trajectory Division: To address the limitations of seq2seq models, such as their sensitivity to training data scale and inefficiency in processing long sequences, we have developed a trajectory division module. This module aims to enrich the training data by dividing trajectories into smaller sub-trajectories, thereby overcoming the challenges associated with limited labeled samples and the inability of seq2seq models to retain long-distance information. Specifically, we implement trajectory division based on sliding window model, where all trajectory points in the same window form a sub-trajectory. The size of the sliding window is set to T in terms of time interval and sliding step is set to H, such as $T = 10min$,[3]

[3] There are 10 to 20 points in every sub-trajectory when $T = 10min$, which is the sequence length that classical RNN-type models can obtain the best performance [40].

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Working Flow of the Trajectory Division

```
Input: Cellular moving trajectory $P_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,n}\}$ of the overlapped training samples $\mathcal{D}$.
Output: A collection of all sub-trajectories $\mathcal{S}$.
Initialize $\mathcal{S} = \emptyset$, $S_{temp} = \emptyset$, sliding window size $T = 10min$, and sliding step $H = 1hop$, index of sub-trajectory $index = 0$.
Set $t_{begin} = t_{i,1}$, $t_{end} = t_{begin} + T$, $index = 1$.
while $t_{begin} \leq t_{i,n}$ do
    for $p_{i,k}$ in $P_i$ do
        if $t_{begin} \leq t_{i,k} \leq t_{end}$ then
            Append $p_{i,k}$ to $S_{temp}$.
        end if
    end for
    if time length of $S_{temp} \geq \frac{T}{2}$ then
        Append $S_{temp}$ to $\mathcal{S}$.
        Set $t_{begin} = t_{i,1+H*index}$, $t_{end} = t_{begin} + T$.
        Set $index++$.
    end if
end while
```

and $H = 1hop$, which can be adjusted adapting to different environments. For a trajectory P_i, we first record the time $t_{i,1}$ of the first trajectory point $p_{i,1}$ as the begin time t_{begin} of the sliding window. Secondly, calculating the end time t_{end} after T-time sliding, i.e., $t_{end} = t_{begin} + T$. Thirdly, dividing all trajectory points whose time falls into the sliding window into a sub-trajectory. Then, sliding window slides H steps forward, and the begin time becomes $t_{i,1+H}$, which is the time of $(1 + H)$-th trajectory point of P_i. The above three steps are repeated to divide sub-trajectories until all trajectory points are processed. It is worth noting that if the length of the last sub-trajectory is too short, that is, the time interval is less than $\frac{T}{2}$, we ignore the last sub-trajectory. The working flow of trajectory division module is summarized in Algorithm 1.

3) Trajectory Representation Enhancement: Trajectory representation learning is the fundamental role for map matching, which converts the raw trajectories in the physical space into representation vectors in the latent space. However, the generated representation from raw cellular moving trajectories that governs the features of the trajectory is not satisfactory for map matching due to scant information on the true moving route. For every same path, there are two different types of trajectories, i.e., sparse cellular trajectory and dense GPS trajectory. To this end, we propose the trajectory representation enhancement component to enrich the cellular trajectory representation, which learns a function to transform sparse cellular trajectories into dense trajectories. Specifically, trajectory representation enhancement consists of two working stages, i.e., training and inference. In the training stage, we train the deep learning model using a tiny trickle of collected dense GPS trajectory data. In the inference stage, with the readily available deep learning models, the dense trajectories are returned with the sparse cellular trajectory inputs. In what follows, we will elaborate on the design details in these two stages as shown in Fig. 10.

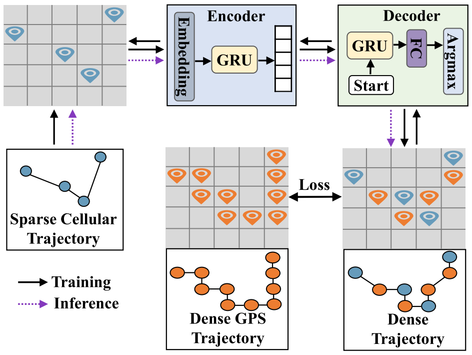


Fig. 10. Illustration of trajectory representation enhancement.

In the training stage, we first describe heterogeneous cellular trajectories and GPS trajectories in a unified way. Compared to the raw point-based trajectory description method, we employ a grid-based trajectory description method, which is more tolerant to location noises. Particularly, we firstly divide the study area into equally-sized grid cells with a given side-length σ in latitude and longitude. Then, we project point-based cellular and GPS trajectories into the grid cells according to their GPS coordinates. In this manner, the heterogeneity of cellular and GPS trajectories is eliminated effectively. We denote each grid cell by g_i, and the complete grid cell set by $G = \{g_i\}$. Thus, $\tau_L = \{g_1, g_2, \dots, g_{n_1}\}$ and $\tau_H = \{g_1, g_2, \dots, g_{n_2}\}$ can be derived from a raw point-based cellular trajectory and GPS trajectory, respectively. We propose to leverage the seq2seq model which learns the mapping function from the input sequence to the target sequence to automatically learn high-quality trajectory representation. As shown in Fig. 10, in our model, we take the sparse cellular trajectory as the input sequence and the corresponding dense GPS trajectory as the target sequence. The main seq2seq model consists of two modules, i.e., an encoder and a decoder, where the encoder extracts the features of variable-length input sequence to a fixed-dimensional feature vector, and the decoder decodes the feature vector into the target sequence. The encoder is implemented with an embedding layer and a GRU layer, while the decoder is comprised of a GRU layer, a fully connected (FC) layer, and an argmax layer. Specifically, trajectory representation enhancement consists of two stages, i.e., labeled training and unlabeled inference. For training stage, the grid-based cellular trajectory $\tau_L = \{g_1, g_2, \dots, g_{n_1}\}$ is fed into the encoder to generate the latent representation $\boldsymbol{z} \in \mathbb{R}^D$, where D represents the dimensionality of the latent space. The embedding layer transforms the one-hot vector into a dense vector of each input grid g_i. The GRU layer reads trajectory points of τ_L successively and generates a sequence of hidden state $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_{n_1}\}$. In i-th step, GRU layer reads g_i and h_{i-1} to produce the i-th hidden state h_i, given by

$$\boldsymbol{h}_i = \mathrm{GRU}\left(\boldsymbol{h}_{i-1}, g_i\right) \tag{4}$$

where GRU [41] is a non-linear function to be learned. After n_1 steps, we get the feature vector $\boldsymbol{z}$, where $\boldsymbol{z} = \boldsymbol{h}_{n_1}$. Then, we feed $\boldsymbol{z}$ into the decoder to generate the predicted trajectory τ'_H, where GRU layer, FC layer, and argmax layer are designed for output generating, size reshaping, and optimal grids ID

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

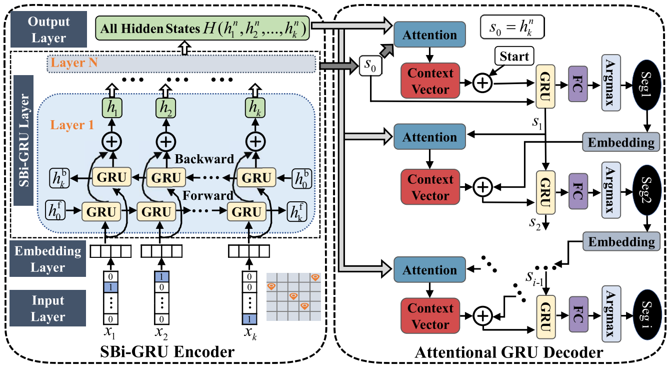


Fig. 11. The architecture of map matching model.

identifying, respectively. We optimize the loss between the predicted trajectory τ'_H and the expected dense GPS trajectory τ_H, where SparseCategoricalCrossentropy loss is used here. By back propagation, the rich details of the dense GPS trajectory can be learned and incorporated into the feature vector, thus enhancing its expressive capability.

After well trained, in the inference stage, the trajectory representation enhancement component with an input sparse cellular trajectory τ_L can produce a dense trajectory τ'_H with rich true route information.

D. Map Matching Model

We design a map matching model with an innovative seq2seq model, consisting of an SBi-GRU encoder and an attentional GRU decoder, as shown in Fig. 11. In our *FL-AMM*, map matching model is jointly implemented by each participant and the server, where the local participant implements the SBi-GRU encoder and the server conducts the attentional GRU decoder. In what follows, we will elaborate on the design details.

1) SBi-GRU Encoder: The input to the map matching model is a cellular trajectory, such as $\tau = \{g_1, g_2, \ldots, g_M\}$, which is a grid-based representation after being enhanced by the data augmentation component. The SBi-GRU encoder can be regarded as a function in the following form,

$$\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_M = f\left(g_1, g_2, \ldots, g_M\right) \tag{5}$$

where f represents the function the SBi-GRU encoder aims to learn, and $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_M\}$ denotes a list of fixed-dimensionality feature vectors, which are the outputs of the SBi-GRU encoder.

In our model, the SBi-GRU encoder is comprised of input layer, embedding layer, SBi-GRU layer, and output layer. Specifically, the input $\tau = \{g_1, g_2, \ldots, g_M\}$ is first fed into the embedding layer to learn a dense vector representation $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_M\}$ instead of employing one-hot representation. Then, the SBi-GRU layer extracts the features of the whole trajectory and transforms the input vector sequence into a sequence of hidden state $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_M\}$. Here, we use the bidirectional GRU (Bi-GRU) as the basic component of the SBi-GRU layer to extract latent features from input trajectories. Compared to normal GRU, bidirectional GRU can process the sequence from both forward and backward directions to extract both past and future information simultaneously. This is helpful in trajectory feature capture since every current trajectory point is determined by both the origin and the destination. Therefore, during the encoding process, the hidden state $\boldsymbol{h}_t$ is updated as

$$\begin{aligned}
\boldsymbol{h}_t^f &= \mathrm{GRU}\left(\boldsymbol{h}_{t-1}^f, \boldsymbol{g}_t\right) \\
\boldsymbol{h}_t^b &= \mathrm{GRU}\left(\boldsymbol{h}_{t+1}^b, \boldsymbol{g}_t\right) \\
\boldsymbol{h}_t &= \boldsymbol{h}_t^f \oplus \boldsymbol{h}_t^b
\end{aligned} \tag{6}$$

where $\boldsymbol{h}_t^f$ and $\boldsymbol{h}_t^b$ denote the hidden states obtained by forward layer and backward layer, respectively, and $\oplus$ represents concatenation operator. Furthermore, we stack several Bi-GRU layers in the SBi-GRU layer to further enhance the feature extracting ability. Let $\boldsymbol{h}_t^l$ be the hidden state of l-th Bi-GRU layer at time t, which is determined by the hidden state of the l-th Bi-GRU layer at time $(t-1)$ and the hidden state of the $(l-1)$-th layer at time t.

$$\boldsymbol{h}_t^l = \text{Bi-GRU}\left(\boldsymbol{h}_{t-1}^l, \boldsymbol{h}_t^{l-1}\right) \tag{7}$$

where $\boldsymbol{h}_t^l$ is concatenated by hidden states of forward layer and backward layer of Bi-GRU, and $\boldsymbol{h}_t^0 = \boldsymbol{g}_t$. Finally, we get a sequence of hidden states $\{\boldsymbol{h}_1^n, \boldsymbol{h}_2^n, \ldots, \boldsymbol{h}_M^n\}$ as outputs after n rounds of feature extracting in the SBi-GRU layer. The last hidden state $\boldsymbol{h}_M^n$ is regarded as the latent representation as well as the initial hidden state $\boldsymbol{s}_0$ for the attentional decoder, which conserves the location information of the whole input trajectory.

2) Attentional GRU Decoder: The traditional decoder of seq2seq model is a simple unidirectional GRU, which generates the map-matched route Y^m successively given the latent representation $\boldsymbol{s}_0$. Formally, the decoder can be represented as $p_\theta\left(Y^m \mid \boldsymbol{s}_0\right)$, where $Y^m = \{e_1, e_2, \ldots, e_m\}$, and θ represents the set of decoder parameters. However, in the decoder structure we described above, the decoder can take only one vector as an input, which is difficult to memorize the whole information on a long trajectory. As a result, the basic seq2seq model suffers accuracy degradation when processing long sequences. To this end, we plug an attention component into the decoder to capture more information from the encoder. Specifically, the attention component considers all hidden states $\{\boldsymbol{h}_1^n, \boldsymbol{h}_2^n, \ldots, \boldsymbol{h}_M^n\}$ of the encoding stage instead of the last latent representation $\boldsymbol{h}_M^n$.

At the beginning, we feed the decoder a Start Of Sequence (SOS) token to start a map matching process. At the i-th step of decoding, let $\boldsymbol{s}_{i-1}$ be the output from the $(i-1)$-th step, we first apply an attention layer to search for the most relevant representation vectors by computing the similarity between the current hidden state $\boldsymbol{s}_{i-1}$ and all hidden states $\{\boldsymbol{h}_1^n, \boldsymbol{h}_2^n, \ldots, \boldsymbol{h}_M^n\}$ of the encoding stage to generate a context vector $\boldsymbol{c}_i$ according to the following formulas,

$$\begin{aligned}
a_t &= F\left(\boldsymbol{s}_{i-1}, \boldsymbol{h}_t^n\right) \quad \forall t, \quad 1 \leq t \leq M \\
p_t &= \exp\left(a_t\right) / \sum_{t=1}^{M} \exp\left(a_t\right) \quad \forall t, \quad 1 \leq t \leq M \\
\boldsymbol{c}_i &= \sum_{t=1}^{M} p_t \cdot \boldsymbol{h}_t^n
\end{aligned} \tag{8}$$

where F denotes the attention function, which is a feed forward network in our implementation. a_t represents a score measuring the correlation between the current hidden state $\boldsymbol{s}_{i-1}$ and an encoding hidden state $\boldsymbol{h}_t^n$. p_t measures the importance of a_t, and M denotes the length of grid-based trajectory. Then, $\boldsymbol{c}_i$ and the embedding vector $\boldsymbol{e}_{i-1}$ of the output e_{i-1} from the $(i-1)$-th step are concatenated, and further fed to the GRU with the current hidden state $\boldsymbol{s}_{i-1}$ to generate the new hidden state $\boldsymbol{s}_i$ and output e_i. That is,

$$\boldsymbol{s}_i = \mathrm{GRU}\left(\boldsymbol{s}_{i-1}, \boldsymbol{e}_{i-1} \oplus c_i\right) \tag{9}$$

$$e_i = \mathrm{Argmax}\left(\mathrm{FC}\left(\boldsymbol{s}_i\right)\right) \tag{10}$$

where FC () represents the FC layer which converts the output of GRU $\boldsymbol{s}_i$ to a |G|-dimensional vector, and |G| represents the total number of grids. $\mathrm{Argmax}()$ denotes the argmax layer which aims to identify the matched road segment e_i. Therefore, the attentional GRU decoder accomplishes the conversion of the learned hidden feature from cellular moving trajectories by SBi-GRU encoder to road segments when an End Of Sequence (EOS) token is generated, and finally obtain a map-matched route $Y^m = \{e_1, e_2, \ldots, e_k, \ldots\}$.

VI. Performance Evaluation

A. Evaluation Methodology

1) Experiment Setup: In this work, we collected a large-scale vehicle trajectory dataset in Changsha under the transportation mode of car, along with related map data from OpenStreetMap, for exemplary evaluation and performance comparison. Specifically, the collected vehicle dataset contains routes in 5 different areas with 377 dual-SIM-card heterogeneous user trajectory pairs, totally lasting 211 hours and covering 6,204 km. Meanwhile, both sparse cellular moving data and dense GPS data are included in our collected dataset. Moreover, we also collect map data from OpenStreetMap to support the map-matching. The road network data contains 4,058 road segments of different types, such as motorways, primary roads, footways, and so on. Then, we first map dense GPS trajectories to road network to generate ground truth labels by an HMM-based method [21]. In order to guarantee a realistic estimation of generalization error, we split data in a trip-wise manner. Particularly, traces collected by 4 data collection campaigns are used for training, and the traces collected by the remaining 1 campaign are used for testing. We implement *FL-AMM* on a server with 4 CPUs each containing 192 Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz with 24 cores, and one graphics processing unit card (NVIDIA Tian X) is used to accelerate the training process. We develop *FL-AMM* in Python and the code is implemented in PyTorch, in which several library files are employed, such as pandas, numpy, sklearn, folium, etc.

2) Metrics: We assess the performance of all map matching algorithms by comparing the map-matched route to ground truth route. Denoted by $Y^m = \left\{e_1^m, e_2^m, \ldots, e_{N_1}^m\right\}$ and $Y^g = \left\{e_1^g, e_2^g, \ldots, e_{N_2}^g\right\}$ the map-matched route and ground truth route, respectively, where N_1 and N_2 denote the total number of data entries of Y^m and Y^g, respectively. The following five metrics are adopted for performance comparison.

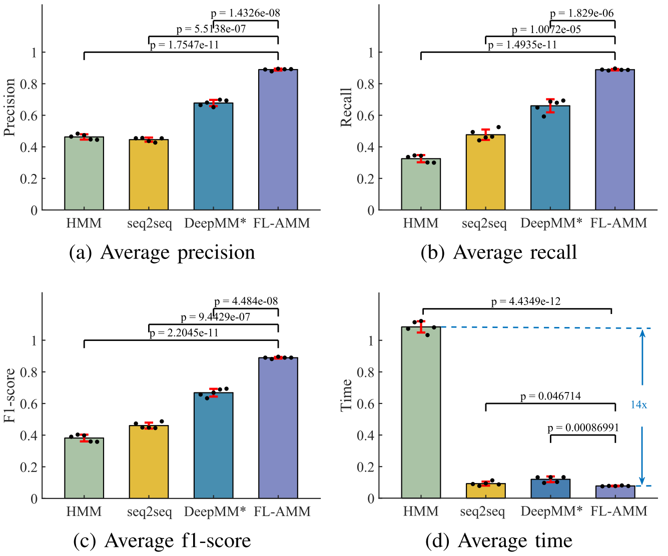
- **Precision:** refers to the ratio of the total length of the correctly matched route to the total length of the route, i.e., $Precision{=}\frac{|Y^m \cap Y^g|}{|Y^m|}$.
- **Recall:** refers to the ratio of the total length of the correctly matched route to the total length of the route in ground truth, i.e., $Recall{=}\frac{|Y^m \cap Y^g|}{|Y^g|}$.
- **F1-score:** is the weighted average of Precision and Recall, calculated by $F1{-}score{=}\frac{2*Recall*Precision}{Recall+Precision}$.
- **Time:** refers to the average inference time, which is defined as the running time of transforming cellular trajectories to road segments.
- **P-value:** refers to significant difference, which is a statistical measure used to determine the likelihood that an observed outcome is the result of chance.

3) Baselines: For performance evaluation, we adopt and implement the following baselines, including traditional model-driven algorithm and data-driven deep learning models.

- ***HMM*** [21]: is a conventional model-driven method to solve the map matching problem. It regards the location of raw trajectory as observation and the road segment as the hidden state for the map matching process. It is proficient in map matching for high-sampling-rate and low-location-error trajectories.
- ***Seq2seq*** [42]: is a widely-used data-driven model to process sequence-to-sequence problem, also known as Encoder-Decoder model, wherein the model used by encoder and decoder can be RNN, LSTM, or GRU. It involves two processes, the encoder is to extract the features from the input sequence, and the decoder is to generate a new sequence based on extracted features. Here, we adopt GRU as the basic processing unit.
- ***DeepMM*** [27]: is an advanced method along data-driven deep learning models in recent years. It is an improved seq2seq model, and the data augmentation technologies are used to improve the accuracy of the model. However, it is designed based on GPS trajectories, and its data augmentation methods proposed are not applicable to cellular data. Therefore, DeepMM is implemented here without the part of data augmentation when implemented in here. So, we rename it *DeepMM**.

B. Performance Comparison

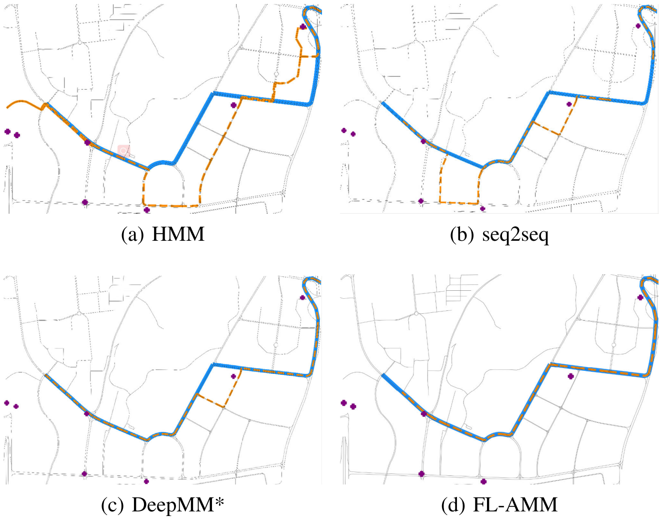
1) Overall Performance: We first conduct the overall performance comparisons between *FL-AMM* and other baselines. To guarantee fair comparison, heterogeneous trajectories of dual-SIM-card users are used for all the baselines by temporarily ignoring data isolation problem. Fig. 12 shows the average metric scores obtained by different models with error bars, where the testing samples are equally divided into 5 subgroups for cross validation. From Fig. 12, we have the following four major observations. First, for all accuracy metrics, i.e., precision, recall, and F1-score, *FL-AMM* can achieve superior performances and the gaps are significant. For instance, the average F1-scores in Fig. 12c are about 0.38, 0.46, 0.67, and 0.88 for *HMM*, *seq2seq*, *DeepMM**, and *FL-AMM*, respectively. That is, the *FL-AMM* can improve the F1-score performance by 132%, 91.3%, and 31.3%,



(a) Average precision (b) Average recall

(c) Average f1-score (d) Average time

Fig. 12. Overall performance comparison.



(a) HMM (b) seq2seq

(c) DeepMM* (d) FL-AMM

Fig. 13. Visualization of map matching. Purple points represent the cellular moving trajectory of operator B. Solid blue lines are ground truth routes, and the orange dotted lines represent the map-matched routes.

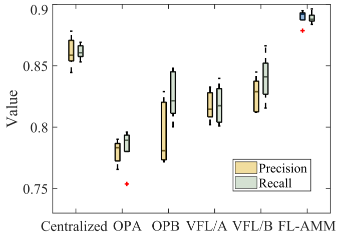


Fig. 14. Effectiveness of VFL.

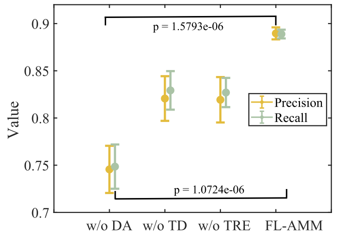


Fig. 15. Effectiveness of DA.

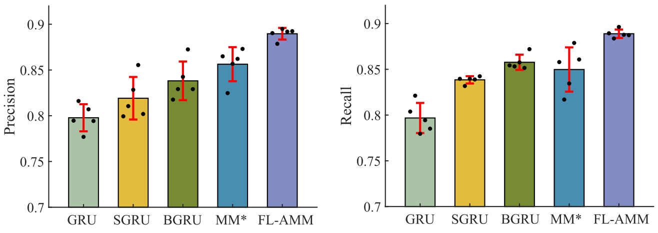
respectively. Note that, for the *HMM* solution, the performance for the precision and recall are 0.46 and 0.33 with a gap between them of 0.13, while for the other three schemes, that gap is less than 0.03. That is, the model-driven algorithm (*HMM*) is more susceptible to the quality of data than the data-driven algorithms (*seq2seq*, *DeepMM**, and *FL-AMM*). Therefore, traditional model-driven models are not suitable for cellular data-based map matching. Second, for the running efficiency metric, i.e., time, as shown in Fig. 12d, emFL-AMM can outperform other baselines, especially for *HMM*, which is 14 times less efficient than *FL-AMM*. Meanwhile, data-driven models have higher running efficiency than the model-driven model. Third, when observing the error bars for all metrics, we can find that the deviation achieved by *FL-AMM* is much smaller than that of other baselines, demonstrating *FL-AMM*'s reliability to guarantee the performance. Finally, for every metric, we calculate the p-value of *FL-AMM* and the other three baselines respectively. We can observe that all p-values are smaller than 0.05, which is generally considered statistically significant in statistics. The lower the p-value, the greater the statistical significance of the observed difference.

2) *Qualitative Analysis:* Fig. 13 shows visualizing examples of how four models perform for map matching problem based on cellular moving trajectories. We can see that our *FL-AMM* finds the right route and the matched result is more precise than the other two baselines. Particularly, Fig. 13a shows the map matching result of *HMM*, which obtains the candidate segment of each trajectory point through shortest path search. Therefore, the road segments matched by *HMM* are very close to the trajectory points but far away from ground truth routes due to high location errors of cellular moving trajectories, thus resulting in inferior matching accuracy. 13b and Fig. 13c show the matching results of two data-driven models, i.e., *seq2seq* and *DeepMM**, respectively. The performances are relatively satisfactory, despite some mismatched road segments of auxiliary lanes. Finally, we can observe from Fig. 13d that our *FL-AMM* achieves a satisfactory matching result in this example, which is because we make good use of the heterogeneous trajectories of dual-SIM-card users and enrich the information content of the real path.

C. Ablation Experiments

In this subsection, we verify the effectiveness of each individual component of *FL-AMM*, i.e., VFL, DA, and MM.

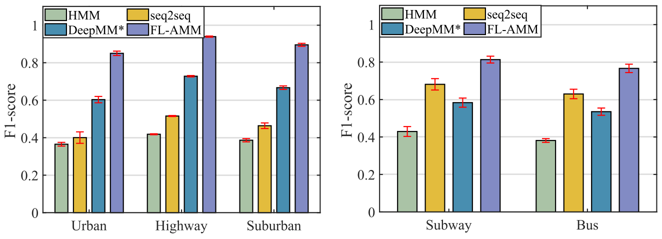
1) *Effectiveness of VFL:* We first verify the effectiveness of VFL for collaboratively using heterogeneous cellular moving trajectories from dual-SIM-card users to enhance the accuracy of the map matching model. We implement several variants of *FL-AMM*, including *Centralized*, *OPA*, *OPB*, *VFL/A*, and *VFL/B*. More specifically, *Centralized*, *OPA*, and *OPB* are the models without employing vertical federated learning framework, and the input of them are mixture of heterogeneous trajectories (A and B) of dual-SIM-card users, cellular trajectories only coming from operator A, and cellular trajectories only coming from operator B, respectively. On the contrary, *VFL/A*, *VFL/B*, and *FL-AMM* denote the models with VFL framework which are trained by dual-SIM-card



(a) Average precision

(b) Average recall

Fig. 16. Effectiveness of the MM.



(a) Performance vs. road types

(b) Performance vs. transportation modes

Fig. 17. Robustness experiments.

users' heterogeneous cellular moving trajectories. They differ in the deployment locations of the trained model. *VFL/A* and *VFL/B* indicate that the models are deployed locally in OP A and OP B, respectively, while *FL-AMM* indicates deploying in a secure third party. Therefore, they are used for map matching inferences for single-SIM-card cellular trajectories of OP A, single-SIM-card cellular trajectories of OP B, and dual-SIM-card cellular trajectories, respectively. Fig. 14 shows the average performance of precision and recall for all six compared variants. We can make the following three statements. First, *FL-AMM* can achieve better performance than the other variants. For instance, our *FL-AMM* can improve the precision performances of *Centralized*, *OPA*, *OPB*, *VFL/A*, and *VFL/B* by 3.5%, 14.1%, 12.7%, 8.5%, and 7.2%, respectively. In addition, when observing the variation of the precision and recall boxes, *FL-AMM* has the smallest deviation than other variants, demonstrating the stability of *FL-AMM*. Third, when comparing *OPA* and *VFL/A*, *OPB* and *VFL/B*, we can find that VFL framework can improve the performance of a single-data-source model because the model of VFL has been enhanced by aggregating data features of other participants.

2) Effectiveness of DA: In our model, DA consists of two sub-modules, i.e., trajectory division (TD), and trajectory representation enhancement (TRE). Therefore, we remove DA, TD, and TRE respectively, to show the performance of *FL-AMM* without these modules. Fig. 15 presents the average scores of precision and recall for map matching based on these variants. It can be seen that the matching performance can deteriorate rapidly after removing these design components. Specifically, the precision score can be reduced from 0.89 to 0.75, 0.82, and 0.82 after removing the DA, TD, and TRE component, respectively. In addition, when observing the error bars for all metrics, we can find the deviation achieved by *FL-AMM* is much smaller than other three variants. Finally, all p-values are smaller than 0.05, which means the observed difference is considered statistically significant.

3) Effectiveness of MM: In our map matching model, there are two outstanding design points, one is the design of SBi-GRU-based data feature extraction module in the encoder, the other is the application of attention mechanism. Therefore, we compare *FL-AMM* with four variants. Specifically, *GRU*, *SGRU*, and *BGRU* represent the variants employing GRU, stacked GRU, and bidirectional GRU as feature extraction module in the encoder, respectively, and *MM** denotes *FL-AMM* without attention module. As shown in Fig. 16, modified GRU models, i.e., *SGRU*, *BGRU*, and *MM** (SBi-GRU), achieve better accuracy than simple *GRU*, among which SBi-GRU has the best performance. In addition, *BGRU* works better than *SGRU* because it can extract features from both past and future directions of input sequence. Second, the performance of *MM** deteriorates dramatically with the largest deviation. Overall, it indicates that the design of MM has significant impacts on map matching performance.

D. Robustness Experiments

1) Impacts of Road Types: To improve the generalization ability of *FL-AMM*, we consider different road types, i.e., urban, suburban, and highway, in the data collection campaigns. The impacts of different road types are shown in Fig 17a. From Fig 17a, we have two main observations. First, the proposed *FL-AMM* consistently outperforms all baselines with an noticeable performance advantage under different road types. Second, for all models, the best performance is achieved in highway areas while the worst result is observed in urban areas. These disparities can be attributed to two factors: BS density and road complexity. Particularly, for highway areas, although BS density is low, the simplicity of road conditions allows for achieving good map matching accuracy even with a limited number of collected cellular trajectories. In contrast, in urban areas, the high BS density is accompanied by increased road complexity. This complexity, characterized by numerous misleading lanes and side roads, poses challenges to achieving satisfactory matching accuracy.

2) Impacts of Transportation Modes: To evaluate the ability of *FL-AMM* to work efficiently under different transportation modes, we adopt two additional cellular trajectory datasets under the transportation modes of bus and subway. Fig. 17b shows the average results of F1-score achieved by different models under two types of transportation modes. Particularly, *FL-AMM* exhibits a significant performance advantage over the other baselines in both bus and subway scenarios. Second, compared to the transportation mode of car, the performance variations in bus and subway modes are different for each model. For instance, the performance of *HMM* increases due to regular schedule routes, while the performance of *FL-AMM* decreases because of the smaller scale of data and more sparse cellular moving trajectories.

VII. Related Work

We review the related work of map matching solutions in two categories, i.e., GPS trajectory-based map matching and cellular trajectory-based map matching.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

A. GPS Data-Based Map Matching

In the literature, various techniques have been proposed to address the map matching problem based on GPS data, which can be roughly divided into two categories, i.e., traditional model-driven approaches [21], [22], [24], [25], [26], [43], [44], [45], [46], [47], [48], [49], and learning-based data-driven approaches [11], [27], [28], [50]. Traditional model-driven map matching approaches can be further divided into several categories, but from the technical perspective, HMM-based model [21], [22], [23], [43] and its variants are the most commonly used in existing solutions. For instance, Jagadeesh et al. [43] proposed a solution combing a HMM model with a route choice model, to develop an accurate, timely and robust map matching algorithm. To achieve satisfactory performance in many ambiguous cases, Hu et al. [46] designed IF-Matching for map matching by utilizing the related meta-information to describe a moving object. However, the performance of these model-based algorithms is usually unsatisfactory, especially for low-sampling-rate GPS trajectories. Adopting learning-based data-driven approaches, Taguchi et al. [50] presented an online map matching algorithm, which employed a probabilistic route prediction model trained by historical data. To overcome the drawbacks of traditional HMM-based models, Feng et al. [27] proposed to solve the map matching task by deep learning approaches. Jiang et al. [11] devised a general and robust deep learning-based model, named L2MM, to achieve a satisfactory mapping accuracy. Liu et al. [28] proposed a deep learning enhanced GPS positioning system to correct GPS estimations. While most of these works can achieve high accuracy for fine-grained GPS trajectories, they cannot be directly used for our work because of the large location error and the inherent sparsity of cellular trajectories.

B. Cellular Data-Based Map Matching

In recent years, some works [29], [30], [31], [51], [52], [53], [54], [55], [56], [57] have explored map matching using the cellular data. For instance, Mohamed et al. [52] proposed a real-time map matching system for cellular trajectories, named SnapNet, which is an incremental HMM-based algorithm with digital map hints and a number of heuristics. To address the challenge of highly inaccurate cellular BS fingerprints, Thiagarajan et al. [51] designed a CTrack system employing a two-pass HMM to identify the most likely roads. Algizawy et al. [30] proposed an extended HMM to tackle map matching problems with large mobile data volume. To solve the problem of unsatisfactory performance at complex road conditions, Han et al. [53] presented a HMM-based model by taking the attributes of road segment, the driver's driving speed and driving directions into consideration. However, these HMM-based works cannot well address high-order historical trajectory information and consume long inference time. Therefore, several other works [29], [31] were proposed to improve the map matching accuracy. Shen et al. [29] proposed a fast map matching model, named DMM, which employs RNN to project cellular trajectories into road segments. To cope with the imprecise performance caused by lacking the labeled data, Rizk et al. [31] presented a data augmentation framework to generate synthetic data. In addition, several works have been proposed to address map matching problems utilizing cellular measurement record data, such as CTS [55], DeepLoc [56], ReculSTM [57], etc. However, most of these works focused on utilizing single cellular data of individual users. Since individual users possess dual SIM cards which can generate heterogeneous cellular data, and this can be of great use to improve the accuracy of map matching.

VIII. Conclusion and Future Work

In this paper, we have proposed FL-AMM for map matching problems based on heterogeneous cellular moving trajectories. Our approach can enhance the representation capability of raw cellular moving trajectories, and transform cellular moving trajectories into segment-based trajectories with high accuracy. In addition, heterogeneous data from different network operators can be effectively utilized when preserving users' data privacy. In the future, we will further improve the performance of map matching by exploring more potential valuable factors (e.g., users' driving patterns, points of interest, etc.). We will also cooperate with mobile operators to fit and implement FL-AMM to the big data platform in production environments in order to accomplish specific map matching tasks.

References

[1] H. Wu, F. Lyu, C. Zhou, J. Chen, L. Wang, and X. Shen, "Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2783–2797, Dec. 2020.

[2] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 26–32, Sep. 2018.

[3] Q. Wu, M. Wu, X. Chen, Z. Zhou, K. He, and L. Chen, "DeepCP: Deep learning driven cascade prediction-based autonomous content placement in closed social network," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1570–1583, Jul. 2020.

[4] T. H. Silva et al., "Urban computing leveraging location-based social network data: A survey," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–39, 2019.

[5] J. Zhou, Z. Cao, Z. Qin, X. Dong, and K. Ren, "LPPA: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 420–434, 2020.

[6] C. Jebarajakirthy, H. I. Maseeh, Z. Morshed, A. Shankar, D. Arli, and R. Pentecost, "Mobile advertising: A systematic literature review and future research agenda," *Int. J. Consum. Stud.*, vol. 45, no. 6, pp. 1258–1291, Nov. 2021.

[7] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann, "ELECTRICITY: An efficient multi-camera vehicle tracking system for intelligent city," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 2511–2519.

[8] R. Xu et al., "Real-time human objects tracking for smart surveillance at the edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[9] W. Shi, R. Shan, and Y. Okada, "A navigation system for visual impaired people based on object detection," in *Proc. 12th Int. Congr. Adv. Appl. Informat. (IIAI-AAI)*, Jul. 2022, pp. 354–358.

[10] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 4th Quart., 2018.

[11] L. Jiang, C.-X. Chen, and C. Chen, "L2MM: Learning to map matching with deep models for low-quality GPS trajectory data," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 3, pp. 1–25, 2022.

[12] H. Wang, Y. Li, D. Jin, and Z. Han, "Attentional Markov model for human mobility prediction," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2213–2225, Jul. 2021.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

[13] F. Lyu et al., "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.

[14] H. Lu et al., "CODE: Compact IoT data collection with precise matrix sampling and efficient inference," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 743–753.

[15] J. Xiong, H. Hu, P. Cheng, C. Yang, Z. Shi, and L. Gui, "Wireless resource scheduling for high mobility scenarios: A combined traffic and channel quality prediction approach," *IEEE Trans. Broadcast.*, vol. 68, no. 3, pp. 712–722, Sep. 2022.

[16] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 858–866.

[17] J. Liu, G. Lin, S. Huang, Y. Zhou, C. Rehtanz, and Y. Li, "Collaborative EV routing and charging scheduling with power distribution and traffic networks interaction," *IEEE Trans. Power Syst.*, vol. 37, no. 5, pp. 3923–3936, Sep. 2022.

[18] F. Lyu et al., "LeaD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2607–2623, Aug. 2021.

[19] Y.-H. Jia, Y. Mei, and M. Zhang, "A bilevel ant colony optimization algorithm for capacitated electric vehicle routing problem," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10855–10868, Oct. 2022.

[20] J. Li et al., "Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13572–13585, Dec. 2022.

[21] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. ACM SIGSPATIAL*, 2009, pp. 336–343.

[22] G. Cui, W. Bian, and X. Wang, "Hidden Markov map matching based on trajectory segmentation with heading homogeneity," *GeoInformatica*, vol. 25, no. 1, pp. 179–206, Jan. 2021.

[23] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," *Int. J. Geographical Inf. Sci.*, vol. 32, no. 3, pp. 547–570, Mar. 2018.

[24] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. ACM SIGSPATIAL*, 2009, pp. 352–361.

[25] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 1144–1155.

[26] T. Hunter, P. Abbeel, and A. Bayen, "The path inference filter: Model-based low-latency map matching of probe vehicle data," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 507–529, Apr. 2014.

[27] J. Feng et al., "DeepMM: Deep learning based map matching with data augmentation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2372–2384, Jul. 2022.

[28] Z. Liu, J. Liu, X. Xu, and K. Wu, "DeepGPS: Deep learning enhanced GPS positioning in urban canyons," *IEEE Trans. Mobile Comput.*, early access, Sep. 21, 2022, doi: 10.1109/TMC.2022.3208240.

[29] Z. Shen, W. Du, X. Zhao, and J. Zou, "DMM: Fast map matching for cellular data," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–14.

[30] E. Algizawy, T. Ogawa, and A. El-Mahdy, "Real-time large-scale map matching using mobile phone data," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 4, pp. 1–38, Nov. 2017.

[31] H. Rizk, A. Shokry, and M. Youssef, "Effectiveness of data augmentation in cellular-based localization using deep learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[32] (2022). *OpenStreetMap*. [Online]. Available: https://www.openstreetmap.org

[33] H. Tian, H. Zhang, J. Jia, M. Dong, and K. Ota, "FedBroadcast: Exploit broadcast channel for fast convergence in wireless federated learning," *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4652–4663, Mar. 2023.

[34] H. Zhang, H. Tian, M. Dong, K. Ota, and J. Jia, "FedPCC: Parallelism of communication and computation for federated learning in wireless networks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 6, pp. 1368–1377, Dec. 2022.

[35] C. Fu et al., "Label inference attacks against vertical federated learning," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 1397–1414.

[36] Y. Kang, Y. Liu, and X. Liang, "FedCVT: Semi-supervised vertical federated learning with cross-view training," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–16, Aug. 2022.

[37] S. Duan et al., "Distributed artificial intelligence empowered by end-edge-cloud computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 591–624, 2023.

[38] L. Lu and N. Ding, "Multi-party private set intersection in vertical federated learning," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 707–714.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD*, 1996, pp. 1–6.

[40] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[41] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 1–9.

[43] G. R. Jagadeesh and T. Srikanthan, "Online map-matching of noisy and sparse location data with hidden Markov and route choice models," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2423–2434, Sep. 2017.

[44] Y. Zhang and X. Sui, "RCIVMM: A route choice-based interactive voting map matching approach for complex urban road networks," *IEEE Trans. Big Data*, vol. 8, no. 5, pp. 1414–1427, Oct. 2022.

[45] Y. Yin, R. R. Shah, G. Wang, and R. Zimmermann, "Feature-based map matching for low-sampling-rate GPS trajectories," *ACM Trans. Spatial Algorithms Syst.*, vol. 4, no. 2, pp. 1–24, Jun. 2018.

[46] G. Hu, J. Shao, F. Liu, Y. Wang, and H. T. Shen, "IF-matching: Towards accurate map-matching with information fusion," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 114–127, Jan. 2017.

[47] X. Liu, K. Liu, M. Li, and F. Lu, "A ST-CRF map-matching method for low-frequency floating car data," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1241–1254, May 2017.

[48] Y. Huang, W. Rao, Z. Zhang, P. Zhao, M. Yuan, and J. Zeng, "Frequent pattern-based map-matching on low sampling rate trajectories," in *Proc. 19th IEEE Int. Conf. Mobile Data Manag. (MDM)*, Jun. 2018, pp. 266–273.

[49] A. U. Peker, O. Tosun, and T. Acarman, "Particle filter vehicle localization and map-matching using map topology," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 248–253.

[50] S. Taguchi, S. Koide, and T. Yoshimura, "Online map matching with route prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 338–347, Jan. 2019.

[51] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. 8th USENIX Symp. Networked Syst. Design Implement.*, 2011, pp. 1–14.

[52] R. Mohamed, H. Aly, and M. Youssef, "Accurate real-time map matching for challenging environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 847–857, Apr. 2017.

[53] B. Han, X. Tang, Z. Hu, and K. Yu, "A map matching algorithm for complex road conditions based on base station data," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 426–431.

[54] A. Viel et al., "Map matching with sparse cellular fingerprint observations," in *Proc. Ubiquitous Positioning, Indoor Navigat. Location-Based Services (UPINLBS)*, Mar. 2018, pp. 1–10.

[55] X. Huang, Y. Li, Y. Wang, X. Chen, Y. Xiao, and L. Zhang, "CTS: A cellular-based trajectory tracking system with GPS-level accuracy," in *Proc. ACM IMWUT*, 2018, pp. 1–29.

[56] A. Shokry, M. Torki, and M. Youssef, "DeepLoc: A ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 339–348.

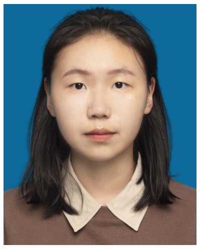
[57] Y. Zhang, W. Rao, and Y. Xiao, "Deep neural network-based Telco outdoor localization," in *Proc. 16th ACM Conf. Embedded Networked Sensor Syst.*, Nov. 2018, pp. 307–308.

Huali Lu (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Central South University, Changsha. Her researches mainly focus on spatial–temporal data mining, compact data collection, and trajectory similarity computing.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.

Feng Lyu (Senior Member, IEEE) received the B.S. degree in software engineering from Central South University, Changsha, China, in 2013, and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018. He was a Post-Doctoral Fellow and a Visiting Ph.D. Student with the BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada, from September 2018 to December 2019 and from October 2016 to October 2017, respectively. He is currently a Professor with the School of Computer Science and Engineering, Central South University. His research interests include vehicular networks, beyond 5G networks, big data measurement and application design, and edge computing. He is a member of the IEEE Computer Society, the Communication Society, and the Vehicular Technology Society. He was a recipient of the Best Paper Award of IEEE ICC 2019. He currently serves as an Associate Editor for IEEE SYSTEMS JOURNAL and a leading Guest Editor for *Peer-to-Peer Networking and Applications* and served as a TPC member for many international conferences.

Huaqing Wu (Member, IEEE) received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Waterloo, ON, Canada, in 2021. She was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, McMaster University, from 2021 to 2022. She is currently an Assistant Professor with the Department of Electrical and Software Engineering, University of Calgary, AB, Canada. Her current research interests include B5G/6G, space-air-ground integrated networks, the Internet of Vehicles, edge computing/caching, and artificial intelligence (AI) for future networking. She received the Best Paper Award at IEEE GLOBECOM 2018, the *Chinese Journal on Internet of Things* 2020, and IEEE GLOBECOM 2022. She also received the prestigious Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Fellowship Award in 2021.



Jie Zhang (Student Member, IEEE) received the B.Sc. degree from the School of Computer Science and Technology, Yanshan University, Qinhuangdao, China, in 2022. She is currently pursuing the M.Sc. degree with the School of Central South University, Changsha, China. Her research interests include trajectory data mining and federated learning.

Ju Ren (Senior Member, IEEE) received the B.Sc., M.Sc., Ph.D. degrees in computer science from Central South University, China, in 2009, 2012, and 2016, respectively. He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include the Internet of Things, edge computing, distributed and embedded AI, and operating systems. He is a member of ACM. He received the IEEE ComSoc Asia–Pacific Best Young Researcher Award in 2021. He was recognized as a highly cited researcher by Clarivate, from 2020 to 2022.

Yaoxue Zhang (Senior Member, IEEE) received the B.Sc. degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the Ph.D. degree in computer networking from Tohoku University, Japan, in 1989. Currently, he is a Professor with the Department of Computer Science and Technology, Tsinghua University, China; and with the School of Computer Science and Engineering, Central South University, China. He has published over 200 technical papers in international journals and conferences, nine monographs, and text-books. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and big data. He is a fellow of the Chinese Academy of Engineering. He is serving as the Editor-in-Chief for *Chinese Journal of Electronics*.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular networks.

He is a registered Professional Engineer of Ontario, Canada; an Engineering Institute of Canada Fellow; a Canadian Academy of Engineering Fellow; a Royal Society of Canada Fellow; a Chinese Academy of Engineering Foreign Member; and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the West Lake Friendship Award from Zhejiang Province in 2023; the President's Excellence in Research from the University of Waterloo in 2022; the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021; the R. A. Fessenden Award from IEEE, Canada, in 2019; the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019; the James Evans Avant Garde Award from the IEEE Vehicular Technology Society in 2018; the Joseph LoCicero Award in 2015; the Education Award from the IEEE Communications Society (ComSoc) in 2017; and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He also received the Excellent Graduate Supervision Award from the University of Waterloo in 2006; and the Premier's Research Excellence Award (PREA) from the Province of Ontario, Canada, in 2003. He serves/served as the General Chair for the 6G Global Conference'23 and ACM Mobihoc'15; the Technical Program Committee Chair/the Co-Chair for IEEE Globecom'24, 2016, and 2007, IEEE Infocom'14, and IEEE VTC'10 Fall; and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President of the IEEE ComSoc. He was the Vice President of technical and educational activities and publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.

Authorized licensed use limited to: University of Waterloo. Downloaded on January 09,2024 at 18:34:49 UTC from IEEE Xplore. Restrictions apply.