Decentralized Attention-based Personalized Human Mobility Prediction

ZIPEI FAN, SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China and University of Tokyo, Japan
XUAN SONG*, SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China and University of Tokyo, Japan
RENHE JIANG, University of Tokyo, Japan
QUANJUN CHEN, University of Tokyo, Japan
RYOSUKE SHIBASAKI, University of Tokyo, Japan

Human mobility prediction is essential to a variety of human-centered computing applications achieved through upgrading of location-based services (LBS) to future-location-based services (FLBS). Previous studies on human mobility prediction have mainly focused on centralized human mobility prediction, where user mobility data are collected, trained and predicted at the cloud server side. However, such a centralized approach leads to a high risk of privacy issues, and a real-time centralized system for processing such a large volume of distributed data is extremely difficult to apply. Moreover, a large and dynamic set of users makes the predictive model extremely challenging to personalize. In this paper, we propose a novel decentralized attention-based human mobility predictor in which 1) no additional training procedure is required for personalized prediction, 2) no additional training procedure is required for incremental learning, and 3) the predictor can be trained and predicted in a decentralized way. We tested our method on big data of real-world mobile phone user GPS and on Android devices, and achieved a low-power consumption and a good prediction accuracy without collecting user data in the server or applying additional training on the user side.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; *Collaborative and social computing design and evaluation methods.*

Additional Key Words and Phrases: human mobility prediction, neural networks, information retrieval

*The corresponding author

Authors' addresses: Zipei Fan, SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China, University of Tokyo, Center for Spatial Information Science, Kashiwa, Chiba, Japan; Xuan Song, SUSTech-UTokyo Joint Research Center on Super Smart City, Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China, University of Tokyo, Center for Spatial Information Science, Kashiwa, Chiba, Japan; Renhe Jiang, jiangrh@csis.u-tokyo.ac.jp, University of Tokyo, Center for Spatial Information Science, Kashiwa, 277-8568, Japan; Quanjun Chen, chen1990@iis.u-tokyo.ac.jp, University of Tokyo, Center for Spatial Information Science, Kashiwa, 277-8568, Japan; Ryosuke Shibasaki, shiba@csis.u-tokyo.ac.jp, University of Tokyo, Center for Spatial Information Science, Kashiwa, 277-8568, Japan.

2474-9567/2019/12-ART133 $15.00
https://doi.org/10.1145/3369830

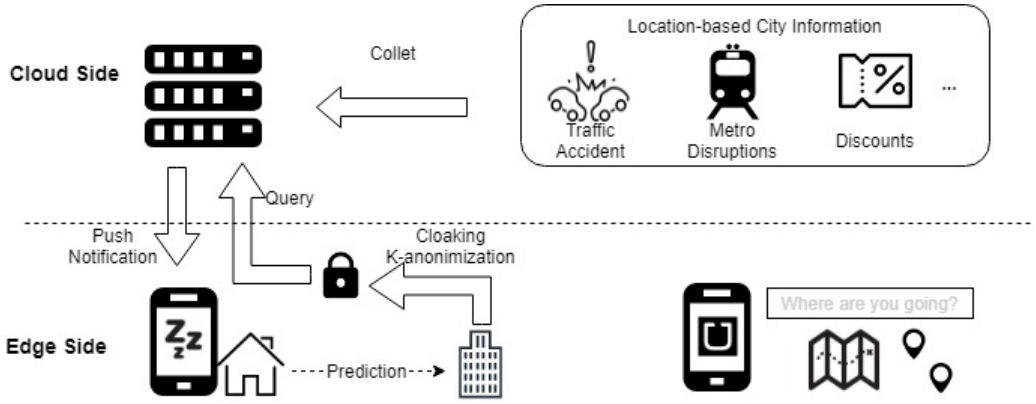

Fig. 1. Applications of mobility prediction on the edge devices.

ACM Reference Format:
Zipei Fan, Xuan Song, Renhe Jiang, Quanjun Chen, and Ryosuke Shibasaki. 2019. Decentralized Attention-based Personalized Human Mobility Prediction. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 133 (December 2019), 26 pages. https://doi.org/10.1145/3369830

1 INTRODUCTION

An accurate human mobility prediction is the building block of numerous applications related to the design of location-based services, which can be both beneficial to society and improve the user experience when applying a smart device. However, most conventional location-based services focus on the user's current location, whereas in some application scenarios, the user's future location has more importance in practice.

As shown in Figure 1, if a traffic accident or a large crowd-drawing event occurs, such information should be timely notified to those users who will likely be effected. The users can then reschedule their route or plan before they set off. Moreover, human mobility prediction also boosts mobile advertising. When we go to a shopping mall by metro or bus, we may spend more time using a smart phone to pass the time along the way than we do while in the shopping mall. Thus, if we make an accurate prediction of human mobility, more accurate location-based advertisement or coupon information can be pushed to users even they are at home or on the way to their destination. In addition, predicting the user's future movement can also enable a wide range of location-based on-device AI applications, such as destination prediction (as shown in Figure 1), smart assistants and smart homes. In such application scenarios, the future location of the user plays a more important role than the current location. Thus, by making an accurate mobility prediction, we can empower location-based on-device AI, and upgrade these location-based services to future-location-based services.

From a data processing perspective, with an increasing number of smart mobile devices and the rapid development of Internet of Things (IoT) technology, a larger volume of human-centered data are being generated in a more distributed way. However, the collection and processing of such tremendous amount of distributed data are costly in terms of both transmission and computation for conventional centralized computations, in which all of the processing is conducted at the server side. Moreover, uploading local user data to the server is also problematic as a data privacy issue, particularly for real-time systems. Against this background, the offloading of computational tasks to the device where the data are generated has drawn increased attention in recent years from both academia and industry.

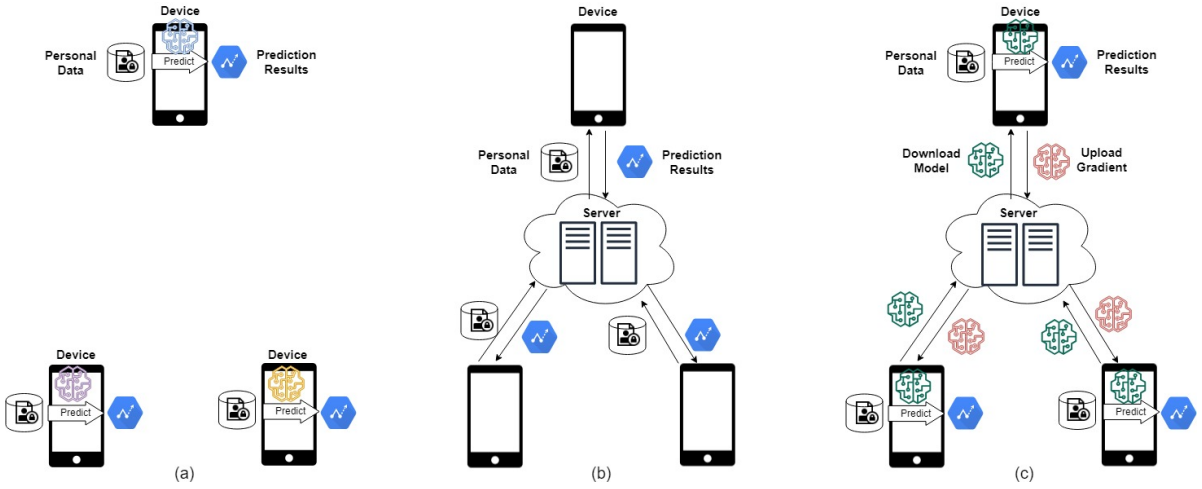

Fig. 2. Three different solutions of personalized human mobility prediction: (a) training and prediction of human mobility at the device end independently, (b) a centralized prediction structure in which all user data are collected in the cloud server, and both the training and prediction are conducted at the server side, and (c) the proposed federated human mobility prediction framework for training and prediction in a distributed way. The predictor exploits the information from all users and maintains the user data locally in the users' smart phones.

These issues mentioned above have also become the bottlenecks of numerous human mobility prediction applications. Human mobility data, particularly for GPS data (with a higher localization precision compared with other localization methods), is much more widely available at the device side than the server side, and transmitting the human mobility data to the server incurs the risk of invasion of user privacy. In addition, many applications (e.g., destination suggestion and location-based recommendation system) have a requirement of low-latency, where making predictions at the device side will be much more advantageous.

Differing from simple tasks that only involve low-level predefined feature extraction and aggregation, human mobility prediction requires higher-level perception of the spatial-temporal properties of both the city layouts (major roads and railways in the city) and human mobility patterns (both personalized and crowd patterns). However, the data of each device are insufficient to support such a higher-level perception. For centralized human mobility prediction, as shown in Figure 2 (b), data from a large number of users are stored and integrated at the server side, and are utilized to minimize the overall predictive loss. In this sense, the final model carries information from all users, which are complementary with each other. This is particularly important for predicting the irregular behavior of a user, or a new user with very little past data.

Recent progress in federated learning has shed light on how to keep user data local (no need to upload local data to a server) while training a model using information from all users. As shown in Figure 2 (c), the model updates are made at the device side, and sent to the model aggregation server, where the distributed model updates are integrated to improve the current model. Numerous model updates from different users are then averaged at the secured server, and the parameters of the predictor at the server side are updated using the averaged model weights. To save battery life and decrease data usage, a smart phone joins federated learning only when it is charging and connected to WiFi. Thus, federated learning decouples the need for training the model with the need for storing all user data in the server, and makes training the predictor at a large scale possible with little compromise in user privacy.

Learning from non-independent and identically distributed (Non-IID) data is the one of the major challenges of federated learning. In a centralized learning procedure, batches are constructed by fully shuffling the entire training dataset. Thus, data from each user are well mixed, and each batch is drawn from the same distribution as the entire data distribution. However, in a decentralized training procedure, only local user data are visible to the local learner, therefore each batch from the entire trajectory dataset of all users is highly non-IID. Limited communication is another bottleneck because the time period in which a mobile phone is suitable for federated learning is limited, and Internet connections are far slower than direct access to memory or GPU in one particular server. To address these problems, we formulate our predictive model as a few-shot learning problem that is capable to transfer the knowledge from one user to another to alleviate the non-IID problem [10], and implement the *FedAvg* algorithm introduced in [27] with an acceleration on the training procedure by pre-training the predictor on another open dataset, or on a small dataset, or by applying a pre-trained model from another city.

Another characteristic of human mobility is its highly personalized aspect, which indicates that the predictor should be well-customized for each user. When training the predictor for each user independently (Figure 2 (a)) the information from all the users cannot be exploited, and the data from a single user are insufficient for training a more sophisticated model. Many previous studies have accepted user ID as an input variable to train a conditional model over the user set, which is more suitable for centralized training and prediction for a fixed user set. In this study, we propose a novel few-shot learning human mobility predictor that does not predict by inferring the user features for personalized prediction, but uses an attention mechanism to learn how to extract useful information from user's past data, which is stored only at the device side. Note that our predictor does not require an additional training procedure to customize the predictor; hence, all predictors at the device side are identical, whereas each predictor is personalized by the user's own local data.

The contributions of this paper can be summarized as follows:

- We propose a novel few-shot learning human mobility predictor that makes personalized predictions based on few records for each user and requires no additional training procedure.
- A federated learning paradigm is applied to train our predictor in a distributed way without directly accessing the user data, and pre-training strategies are proposed to accelerate the federated learning procedure.
- We tested our proposed model and the training paradigm using real-world GPS big dataset, and achieve the state-of-the-art prediction accuracy.

2 RELATED WORK

Internet of Things (IoT) Big Data: With the popularization of smart phones and smart devices, an increasing amount of human mobility big data are being collected, such as GPS logs [13, 36], social network data [7, 17, 40, 43, 52], query data from routing applications [28, 51], mobile crowd-sourcing data [47] and bike rental data [6]. Massive datasets and relevant studies have shown the significant potential of exploiting big data for both improving user experience and solving long-existing urban problems. These data are collected either in a active way (e.g. check-in [7, 17, 40, 43, 52], query data [28]) or a passive way (e.g. GPS trajectory collected in the background [13, 36]). Users are more aware of publishing their location, and relatively less privacy issues are involved, while the latter, location is sensed and recorded as a background service, and thus a higher risk of privacy invasion. However, the latter ones have a much dense trajectories, which is very important in many application scenarios (e.g. traffic jam pre-warning,). In this work, we aim at exploiting the advantages of dense trajectories and maximize the user privacy protection in an edge computing paradigm.

Human Mobility Prediction: Human mobility prediction has been a popular research topic in the fields of traffic engineering [5, 39], ride-sharing [45], computer vision [2, 33], and emergency management [48**?**].

Traditional methods predict future movements by searching for similar trajectories in the past trajectories [32], modeling periodical mobility patterns [7], or integrating heterogeneous data sources [37].

Recent studies on big data have shown that the performance of traditional machine learning methods stops increasing after the amount of data exceeds the model capability, which is limited to the manually designed features and domain expertise. Deep learning has shown significant potential in increasing the model capability. There have also been some pioneering studies on introducing deep learning into human mobility prediction. In the direction of human mobility prediction at an aggregated level, [48] successfully applied residue networks and a convolutional neural network to predict citywide crowd flows. [45] proposed a neural network that can incorporate different data sources well, particularly encoding a road network using a graph embedding technique. [44] proposed a transfer learning approach that predicts the citywide population density by transferring the knowledge from other cities. These studies are mainly used for predicting the population density of the city, which is helpful in urban emergency management and traffic regulation, but loses rich information of individual movement, which is important for various location-based services.

For a prediction at the individual level, [**?**] proposed a multi-task deep learning framework to simultaneously predict the transportation mode and location. These two tasks are highly correlated, and therefore each can boost performance of the other. [3] predicted a user's next check-in behavior by fusing the user's check-in history, spatial distance and user friendship. [14] proposed an deep ensemble framework to predict the both the regular and irregular movement of the user. Most related research in this area was conducted by [16].We improved this method 1) using an key-value attention mechanism that can better preserve the structural information of past trajectories, and 2) adapted the predictor to be more suitable for a decentralized prediction through removing the user embedding and encode all the personalized information in the document trajectories. Experimentally, our predictor is better at extracting the long-term dependency in human mobility and significantly improving the prediction results for periodical behavior.

Federated Learning & Privacy Protection: With a dramatic increase in the number of smart devices connected to the Internet, more and more privacy data is being leaked unconsciously. In recent years, people have become more aware of privacy issues. In 2016, the European Union enforced a General Data Protection Regulation [12] to protect user privacy by granting the users' right to delete their personal data and urge the companies to clarify what data are being collected and how the personal data are being used through the user agreement.

However, personal data can be profitable to a company, and beneficial to the society and improving the user experience. Thus, in recent years, different forces (companies, users and academics) have begun searching for a balancing point that can exploit both the commercial and societal value of personal data while at the same time protect user privacy. Differential privacy [11], which provides a mathematical definition of privacy, has been deployed by Apple to protect the privacy of iOS users. [34] proposed the use of k-anonymity algorithm to protect user privacy by mixing the true data with fake data, and [19] and [22] introduced the k-anonymity algorithm into location and human trajectory privacy protection, respectively. A more recent study [35] showed that conventional k-anonymity protection for human trajectory data may fail with a semantic attack, and a countermeasure was proposed in the same paper to deal with this issue. Another technique for protecting user trajectory privacy is aggregation [53], whereas [38] showed that simply aggregating user trajectories does not completely preserve user privacy. However, most of these methods sacrifice too much of the utility of the data, and thus a higher level of analysis, such as prediction, is hardly to conduct for such protection measures.

In recent years, smart devices with stronger computational and data communication capabilities have become more popular, and hence more complex computational tasks can be conducted by the device side at a low latency. Against this background, federated learning [27] has shed light onto large-scale distributed machine learning. Edge devices update on the models based on their own local data, and only updated model are uploaded and integrated at server side, without transmitting the user data. [4] provides more details on implementing a federated learning

system in a practical way. In addition, [18] analyzed the vulnerability of federated learning under differential attacks, and proposes a counter-measure to enhance the client-side security. [50] improved a global shared data to address the non-IID data learning problem in federated learning. [42] provides a comprehensive review on the recent progress in federated learning and the prospects in this direction. For the present study, we were inspired from these studies to design our human mobility prediction experiments in a federated learning way, and propose pre-training techniques inspired from [50] for training the human mobility predictor that accelerates the federated learning procedure, which is insufficiently discussed in the previous work.

Edge Computing: Edge computing is a popular topic in ubiquitous computing, and many interesting applications have been proposed [1, 31] in recent years. Edge computing aims at both preserving and processing the user's data at the edge devices, laying the foundation of federated learning on smart devices. One benefit of edge computing is that user privacy is better protected because no data transmission is required, which is discussed in the paper [46] comprehensively. Another important benefit of edge computing is that the computing task is conducted close to where the data are generated and utilized. Thus, the latency can be minimized and only a negligible communication cost is required. In this study, we decentralize the human mobility prediction task to each smart phone user following the paradigm of edge computing to achieve the benefits of both privacy protection and low latency. In particular, with the help of federated learning and our proposed prediction model, we can offload a human mobility prediction task, which is conventionally conducted at the cloud side, to edge devices (smart phones or tablets).

There are also an emerging trend on transplanting deep learning frameworks to mobile edge-devices (TensorFlow Lite [21], NNAPI [20], Caffe2 [25] and network structures (MobileNets [23], ShuffleNet [49])to enable a wide-range of on-device deep learning tasks. Computer vision community has been more enthusiastic in on-device deep learning [15, 24] thus convolutional neural network is better supported, while recurrent neural network and dynamic network is not supported in most of the existing implementations. In this paper, we implement the on-device inference of our proposed model using our original library on Android that supports dynamic network structure for sequential modeling.

3 PRELIMINARIES

In this section, we define the terms and concepts frequently used throughout this paper.

Definition 3.1 (Raw GPS data). Each reading from a localization sensor can be described as a 3-tuple of the time stamp, latitude and longitude. Thus, raw GPS data collected by the mobile device u can be formally represented as follows:

$$X_u = \{(t,\ lat,\ lon)\} \tag{1}$$

Thus, the trajectory of each user raw_traj_u can be defined in the following manner of:

$$raw\_traj_u = x_{u,0}, x_{u,1}, \ \ldots\ , x_{u,n_u}, \qquad x_{u,i} \in X_u \tag{2}$$

where $x_{u,i}$ is the i-th record (ordered by time) of user u.

Definition 3.2 (Location Cluster). To make it easier for learning a spatial multimodality, we introduce location clusters $\{c_m = (lat_m,\ lon_m) \mid m = 1, \ \ldots\ , M\}$ (as shown in Figure 3) to transform the geo-coordinates (latitude, longitude) in will to the nearest location cluster.

Definition 3.3 (Cluster-level trajectory). A cluster-level trajectory $traj$ is represented by the sequence of cluster IDs:

$$traj = c_0,\ c_1,\ \ldots\ , c_{T-1} \tag{3}$$

where T is the length of the trajectory. Typically, the time interval between two adjacent cluster IDs is 15 min.

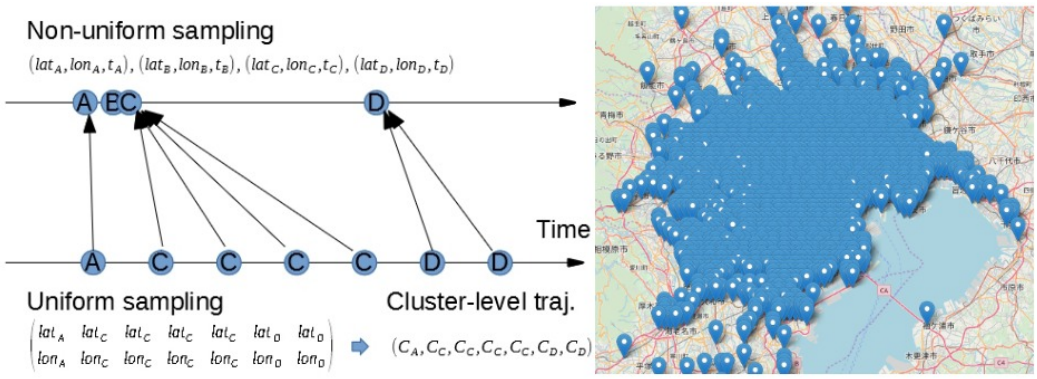

Fig. 3. Illustration of interpolating a non-uniform sampling trajectory to obtain a uniform sampling trajectory (left) and cluster distributions in the Tokyo area (right).

Definition 3.4 (Past/most recent trajectory). To model both the long- and short-term dependency respectively, we slice the trajectory of user u into the past trajectories $\left\{traj_u^d \mid d \in D_u\right\}$ on day d and the most Δt recent trajectory $traj_{u,\, t-\Delta t:t}$. Note that users may leave the studied region or switch off their mobile phone, and thus we do not expect we have user data on every day in our dataset. Thus, we use D_u to denotes the set of days in which the user has records within the studied region.

Definition 3.5 (Human Mobility prediction). In this study, we predict the future Δt-ahead location $traj_{u,t+\Delta t}$ based on the most recent Δt observations $traj_{u,t+\Delta t}$ and past trajectories $\left\{traj_u^d \mid d \in D_u\right\}$ from the same user. Thus, we can formulate the prediction task at time t as modeling the conditional probability:

$$p\left(traj_{u,\, t+\Delta t} \mid traj_{u,\, t-\Delta t:t}, \left\{traj_u^d \mid d \in D_u\right\}\right) \tag{4}$$

Note that, Δt is used for both Δt-ahead prediction and Δt most recent observations. They have different semantic meanings but we use the same symbol and we set it both 4 (1 hour) in our experiments.

4 TRAJECTORY PRE-PROCESSING

In this section, we present the details of how to process raw trajectories to obtain the cluster-level trajectories. To model the long-distance and long-term dependencies from the user trajectories, we need to transform the raw GPS trajectory into the cluster-level trajectory. As shown in Figure 3, we obtain a cluster-level trajectory from raw GPS log data using a two-step process:

1) Forwarding-filling (interpolation using the last observation) the trajectory to obtain a uniform-sampling trajectory. Note that forward-filling is suitable for online systems because no future information is required. An illustration of this process is given on the left side of Figure 3.

2) We divide Tokyo/Osaka and the surrounding area into grid cells using the method defined in [9] into K clusters and represent each coordinates from 1) through its nearest cluster center. We select the most frequently visited grid cells in the city and assign the cluster IDs by the ranking. A visualization of the distributions of $K = 1600$ clusters in Tokyo is given on the right side of Figure 3, and basic statistics of the adjacent cluster center distance is provided in Table 1.

Therefore, we regularize the raw GPS trajectory from both spatial and temporal aspects. In a temporal aspect, we re-sample the non-uniform sampling trajectory to obtain a uniform sampling trajectory. This makes it easier to extract spatial-temporal features and accelerate the processing through batching (non-uniform sampling

Table 1. Adjacent Cluster Center Distance Statistics

	min (km)	max (km)	median (km)	mean (km)
Tokyo	0.92	507.02	0.92	1.97
Osaka	0.92	192.42	0.92	1.32

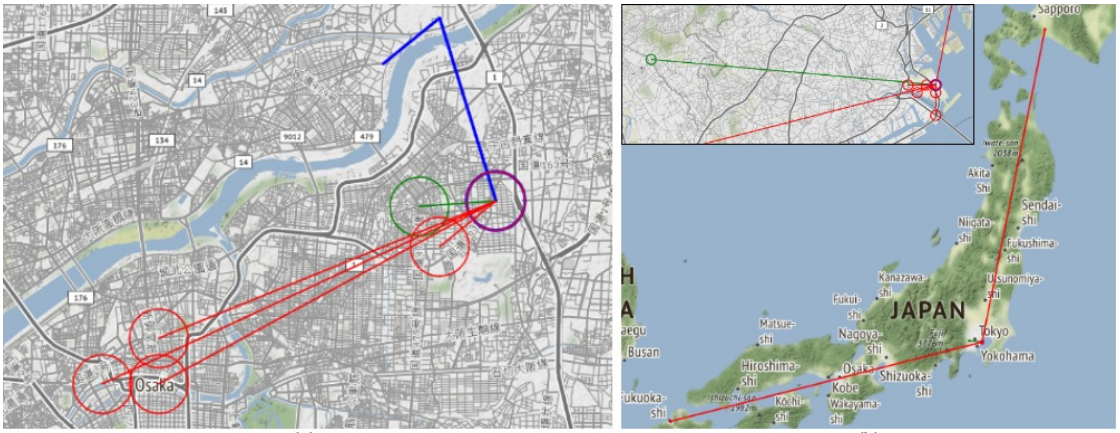

Fig. 4. An Illustration of the spatial multimodality problem. The observed trajectory is shown through the bold black line, two candidates (each with a 50% probability) of future movements are shown with the dashed line, and the red line indicates the predictions with (right) or without (left) considering the spatial multimodality problem.

trajectories have variant length, which is difficult for batching). In a spatial aspect, we prefer a discrete (cluster IDs) location representation of the trajectory rather than continuous values (coordinates) because a complex transportation network makes the spatial dependency highly non-linear; in particular, the locations along highways or railways can be regarded as singularities, making them difficult to model through continuous functions. Another reason why we prefer a discrete location representation is that discrete variables are more powerful in characterizing spatial multimodal distributions. As shown in Figure 4, prediction ignoring a spatial multimodality is a weighted average of the candidate locations (shown in green), which is meaningless particularly when the user uses high-speed transport such as an airplane or high-speed express train. Figure 4 (b) shows the prediction of a user at Haneda airport, which is the major airport in Tokyo, particularly for domestic airlines. Our predictions (shown in red) can well characterize such a spatial multimodality while ignoring the spatial multimodality generating unrealistic prediction results (green).

The location clusters $\{c_m \mid m = 1, \ldots, M\}$ are determined based on the centers of the most frequently (experimentally, we set the number of clusters $M = 1600$) visited grid cells in the studied area. On the right side of Figure 3, we visualize the centers of the clusters $c_m = (lat_m, lon_m)$ in the Tokyo area.

5 HUMAN MOBILITY PREDICTION

Future human mobility prediction relies on both short- and long-term dependency. The current location and the most recent movement of the user, which are modeled using the short-term dependency, enforce a strong spatiotemporal continuity in which the user will be more likely to move somewhere nearby than to a distant place. By contrast, the user's past trajectories will also provide rich information such as important locations, life patterns, and transport preference. Such information and how it relates to the current movement of the

user are modelled as long-term dependencies. From the perspective of a personalized prediction, short-term prediction is not concerned about individual information. This is, two users are regarded as the same and make the same predictions (same probability distribution over the locations) if their most recent Δt trajectories are identical. Personalized prediction is mainly handled in the long-term dependency, whereas we utilize an attention mechanism to extract supportive information from the user's past trajectories. In the following subsections, we introduce how to model short- and long-term dependencies, respectively, and how to fuse these two aspects to make a more accurate personalized human mobility prediction.

5.1 Short-term Predictor

A short-term predictor describes the adjacency of clusters in a city layout, and acts as a prior of human mobility patterns without any knowledge of the user. To this end, we built a predictor F_S to model the sequential patterns from the cluster-level trajectories of all users, where the predictor $_S$ is a function that takes the past cluster-level trajectories $traj_{t-\Delta t:t}$ and the current time-of-day t as inputs, and outputs the conditional probability distribution $p\left(traj_{t+\Delta t} \mid traj_{t-\Delta t:t}\right)$ of the nearest cluster of the user 1h later. Thus, we are essentially training a **multi-class** classifier given a **sequence** of **discrete** data. In correspondence to these three keywords, we design our model as follows:

- **Discrete**: We use an embedding layer EL with the vocabulary size of the number of clusters to map the cluster ID to a N_E-dimensional vector. Time-of-day is mapped in the same way using the embedding layer ET.
- **Sequence**: To model the spatiotemporal patterns, we choose a gated recurrent unit (GRU) [8] as a key part of the sequential modeling. A GRU is a simplified version of the long short-term memory (LSTM) model, which is the most popular neural network for long sequence modeling. The GRU maintains most of the design of LSTM, except that the update and reset gate vector are merged into a forget gate.
- **Multi-class**: We use a SoftMax layer that outputs a normalized exponential prediction of the probability distribution over the clusters.

The formulation of the cluster-level predictor can be summarized as follows:

$$h_0 = \mathbf{0} \tag{5}$$

$$h_\tau = GRU\left(\left[EL\left(traj_{t-\Delta t+\tau}\right), ET\left(t\right)\right], h_{\tau-1}\right) \qquad for\ \tau = 1, \cdots, \Delta t \tag{6}$$

$$o_{\Delta t} = SoftMax \circ FC \circ ReLU \circ FC\left(h_{\Delta t}\right) \tag{7}$$

where h is the hidden state of the GRU at each time step and o is the output vector representing the target distribution over the clusters. In addition $\circ$ is a function composition. FC denotes the fully connected linear layer, and $ReLU$ is rectified linear unit [29], which is a widely used non-linear activation function used in a neural network.

A short-term predictor is trained by minimizing the cross entropy loss between the predicted distribution $o_{\Delta t}$ and the one-hot distribution of the real movement $traj_{t+\Delta t}$. As the key part of our proposed model, in particular, when a user is new to the studied area, we show the short-term predictor as purple in Figure 5 (the merged latent layer is skipped).

5.2 Attention-based Personalized Human Mobility Predictor

In this subsection, we elaborate on the details of the attention mechanism to model the individual aspects of each user. The network structure is shown in Figure 5. We use "keys" for positioning the useful long-term information from past trajectories (blue), and "values" to gather and integrate the information (green). Thereafter, the retrieved long-term information is fused with short-term information (purple) to predict future movement.

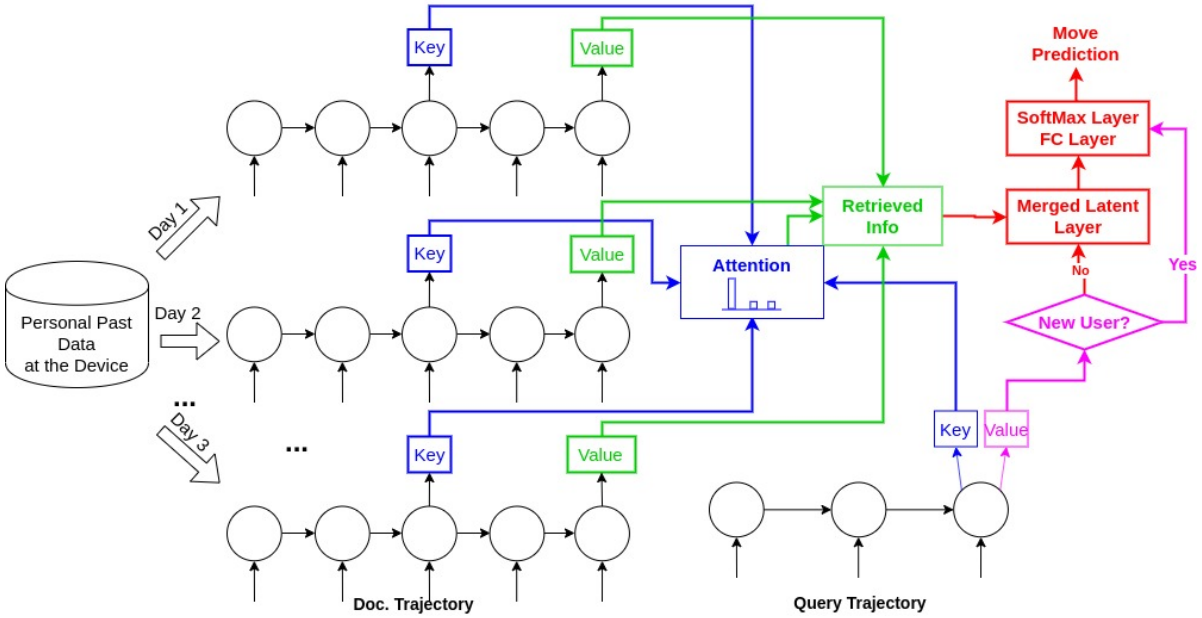

Fig. 5. Network structure of personalized human mobility prediction. The purple color indicates the short-term module, blue and green indicate the key-value attention mechanism for modeling the long-term dependency, and red shows the fusion part integrating the short- and long-term dependencies to make the final predictions.

5.2.1 New Users. Not every user has past trajectories. Figure 6 shows the new user rates in both Tokyo and Osaka for November 2012, from which we can see that the new user rate is approximately 2% on average in both cities (weekends have rates 2- or 3-times higher than weekdays). For those new users who have no past trajectories in the studied area, we do not have any cues in the long-term dependency for prediction. In this case, our predictor switches to the short-term predictor introduced in Section 5.1.

5.2.2 Key-value Attention. As a difficulty of modeling long-term dependencies, most of the past trajectories are irrelevant to the current prediction, which is quite different from short-term prediction. In other words, we need to **localize** and **summarize** the useful information from past trajectories. Therefore, this problem can be formulated into a neural information retrieval problem [26], and a key-value attention mechanism is a natural choice, where "keys" are used for localizing the information and "values" are used for summarizing useful information.

In the sense of information retrieval, the most recent trajectory $traj^{qry}_{u,t:t+\Delta t}$ can be regarded as a query, whereas each past trajectory at the same time of day $traj^{doc}_{u,t:t+2\Delta t}$ can be regarded as a document. As shown in the blue part in Figure 5, we first calculate the keys **Key** from both the query and documents, and the relevance $r(qry, doc)$ between each query and document trajectory pair $\left(traj^{qry}_{u,t:t+\Delta t}, traj^{doc}_{u,t:t+\Delta t}\right)$ is estimated through the dot product of the keys:

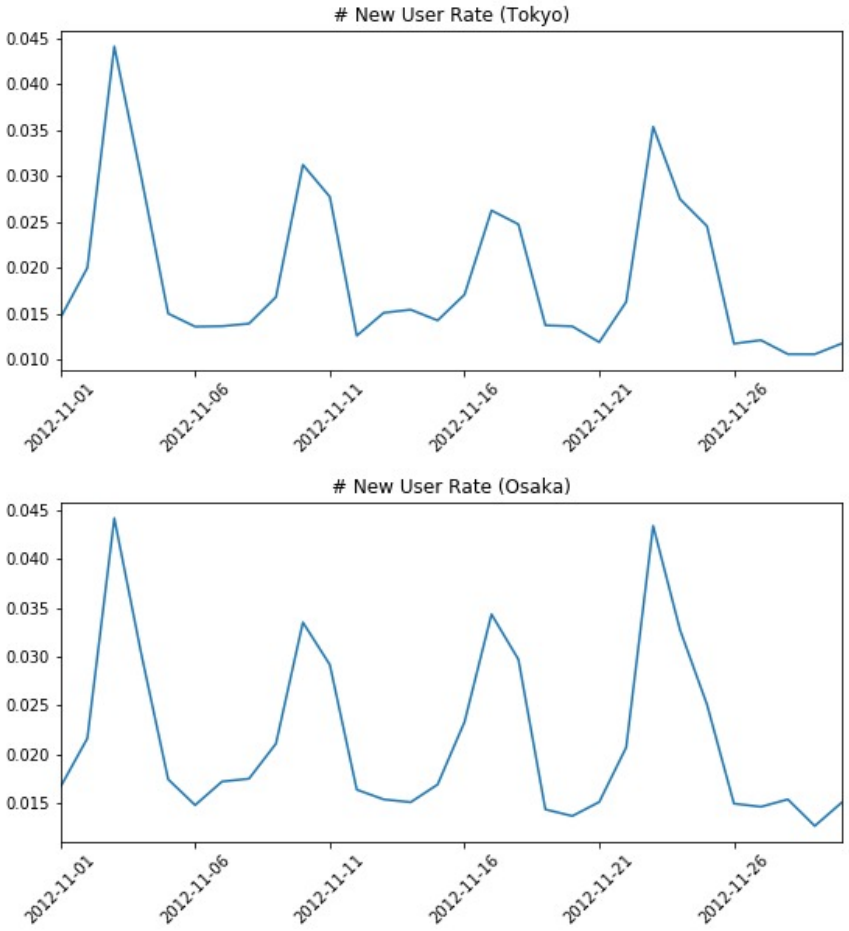

Fig. 6. New user rate in Tokyo (left) and Osaka (right) in November of 2012

$$\begin{cases} \mathbf{Key}^{doc}(\mathbf{h}) &= f^{doc}(\mathbf{h}) \\ \mathbf{Key}^{qry}(\mathbf{h}) &= f^{qry}(\mathbf{h}) \\ r(qry,\, doc) &= \mathbf{Key}^{qry}\left(\mathbf{h}^{qry}_{t+\Delta t}\right)^{T} \cdot \mathbf{Key}^{doc}\left(\mathbf{h}^{doc}_{t+\Delta t}\right) \end{cases} \tag{8}$$

where $f = FC \circ ReLU \circ FC$ is a multi-layer fully connected network with ReLU as an activation function. The attention (normalized relevance) $a^{qry,\, doc}$ is then calculated by normalizing the relevance through a $SoftMax$ as follows:

$$a(qry,\, doc) = \frac{exp(r(qry,\, doc))}{\sum_{doc'} exp(r(qry,\, doc'))} \tag{9}$$

where exp is the exponential function with a base of e. The work-flow of estimating $a(qry,\, doc)$ is shown in blue in Figure 5.

It is worth noting that the relevance is estimated using the keys generated from $\mathbf{h}^{qry}_{t+\Delta t}$ and $\mathbf{h}^{doc}_{t+\Delta t}$. That is, the relevance estimates the similarities between the query trajectory $traj^{qry}_{u,\,t:\,t+\Delta t}$ and $traj^{doc}_{u,\,t:\,t+\Delta t}$. However, for the task of Δt-ahead prediction, useful information is more likely to be located at the time as $t + 2\Delta t$. Thus, as shown in Figure 5, we extract values at Δt time steps after applying the keys. With the help of the estimated attention $a\,(qry,\ doc)$, we retrieve the information from all document trajectories through a weighted average:

$$\textbf{Context}(\{traj^{doc}_{u,\,t:\,t+2\Delta t}\} \mid traj^{qry}_{u,\,t:\,t+\Delta t}) = \sum_{doc'} a\,(qry,\ doc)\,\textbf{Val}(\mathbf{h}^{doc'}_{t+2\Delta t}) \tag{10}$$

where Val is the value function that has the same form of key function defined in Equation 8. Based on this, the context information $\textbf{Context}(\{traj^{doc}_{u,\,t:\,t+2\Delta t}\} \mid traj^{qry}_{u,\,t:\,t+\Delta t})$ is retrieved from the past trajectories $\{traj^{doc}_{u,\,t:\,t+2\Delta t}\}$ conditioned on the query trajectory $traj^{qry}_{u,\,t:\,t+\Delta t}$.

Thus, a merged representation is obtained by concatenating the long-term context information **Context** with the short-term information $\textbf{Val}^{qry}\left(\mathbf{h}^{qry}_{t+\Delta t}\right)$, and fed into a linear layer using the $SoftMax$ activation function to predict the user's future movement $p\left(traj_{u,\,t+\Delta t} \mid traj_{u,\,t-\Delta t\,:\,t},\ \left\{traj^{d}_{u} \mid d \in D_u\right\}\right)$. The training cost is defined as the cross entropy between the predicted distribution of future movement and Δt-ahead real movement.

In Figure 7, we demonstrate how our attention-based predictor handles long-term dependencies. In (b), the query trajectory (blue) and the user's real future movement (green) are shown. The document trajectories from the same user are visualized in (c). We calculate the attention of each document trajectory from Equations 8 and 9. In general, the attention is proportional to the similarity between the query and document trajectories. The brown and yellow trajectories, the attention of which are approximately 40%, show quite a similar moving pattern with the query trajectory, and provide accurate future movement information. By contrast, the purple trajectory, which remains nearly static and differs significantly from the query trajectory, has the lowest attention. The red trajectory, which seems to be a detour from the most frequent route, also draws a reasonably high amount of attention (but much lower than those of the brown and yellow trajectories). Integrating the information from the document trajectories, we can make an accurate prediction with high confidence for the user's future movement, as shown in (e).

The three important features of our prediction method are as follows:

- The attention mechanism provides a permutation-invariant "reduce" operation on the set of document trajectories. This means we do not make any assumption on the order and size of the set of document trajectories. In this sense, we are free to add new trajectories or delete old trajectories for **incremental learning** without any additional training procedure. In the later section, we demonstrate how incremental learning will improve the prediction accuracy. In addition, differing from most of permutation-invariant operation such as "MAX" and "MEAN" used in [16], our trajectory-level attention mechanism provides a weighted average that can automatically extract useful information and filter out irrelevant samples.
- We model the personalized human mobility prediction problem as a few-shot learning problem; hence, we regard the prediction for each individual user as a learning task that has only a few samples for training. Under the paradigm of few-shot learning, we share the model weights among the prediction task for each individual. More specially, we train only one model that addresses all the human mobility prediction tasks for each individual user. This is the important premise for federated learning and decentralized prediction, as shown in Figure 2 (c). We introduce this part in more detail in the following subsection.
- The Key and Val for each document trajectory can be pre-computed and thus significantly improve the online prediction efficiency.

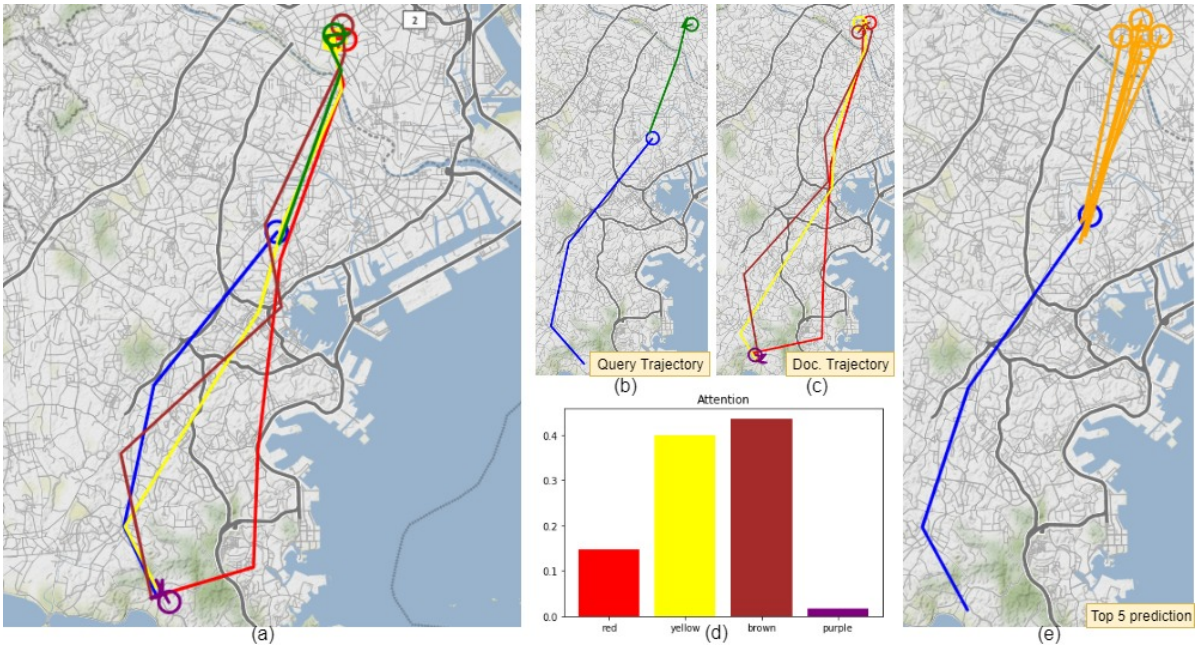

Fig. 7. An illustration of the attention mechanism in personalized human mobility prediction of a volunteer user. (a) The query trajectory (blue), real movement (green) and document trajectories (red, yellow, brown, and purple). (b), (c) The query trajectory (along with the real movement) and document trajectories separately. (d) The attention value of each document trajectory. (e) The top-5 predictions from our proposed method.

5.3 Decentralized Learning

Many previous studies have focused on either ignoring personalized information or training a human mobility predictor (shown in Figure 2 (b)) in a centralized way. This can incur either a low prediction accuracy (where no personal information is used) or a privacy invasion (particularly for an online data stream). Thus, in this work, with the merits discussed in the previous subsection, we train the human mobility predictor in a decentralized way with both a good utilization of the trajectory data from all users and a high level of protection of user privacy by maintaining all the user data at their device non-transferred.

Following the design of practical federated learning paradigm introduced in [4], we implement our learning strategy shown in Figure 8. When the user's mobile phone is in use or is unplugged or the device has no WiFi connection, we simply collect the user's location data using a constant time interval and store the record at the device side. Future movement is predicted at a constant time interval or when requested by certain applications. With the user's agreement to join federated learning, when the mobile phone is idle, plugged in, and has a good WiFi connection, the mobile phone will be regarded as a candidate participant of the federated learning. Considering the procedure of each learning iteration, we refer to the design of [27] and implement our decentralized learning framework that each data communication round includes the following steps:

(1) A set of available candidates are randomly selected.
(2) The current model is synchronized from the server to the candidate devices.
(3) Personal historical data are retrieved and form a batch of training data at the device side.

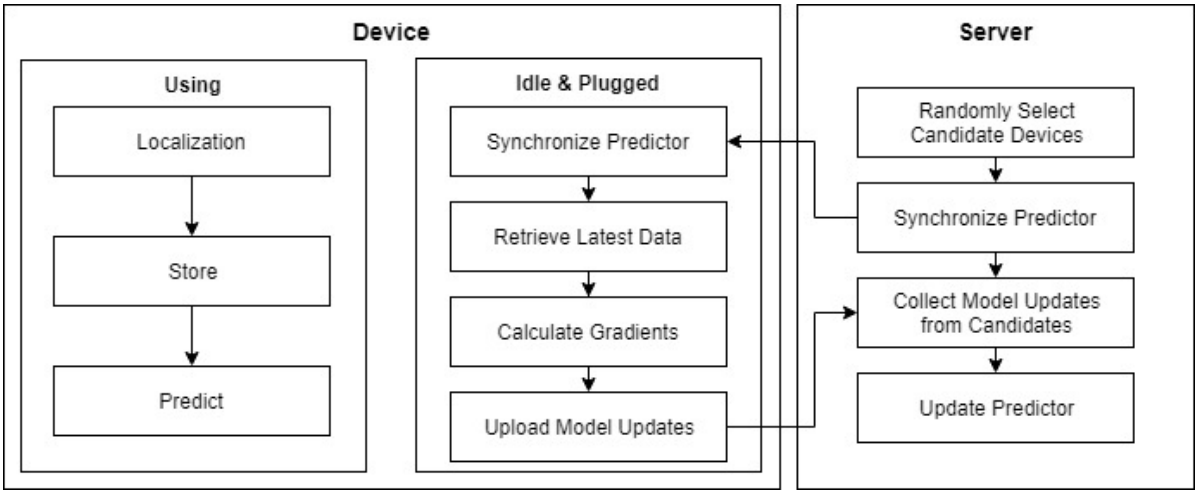

Fig. 8. Flowchart of federated learning method for decentralized human mobility prediction.

(4) Stochastic gradients are computed and slightly corrupted with random noise to increase the level of security, and the models are updated at the device side using the local batch.
(5) The updated model is uploaded from the device to the server.
(6) The updated models from all candidates are averaged, and the server model is upgraded using the averaged model.

Federated learning suffers from the problem of severe Non-IID and limited and expensive communication cost. Even in centralized learning, a deep neural network with a complex structure (e.g., attention and GRU) requires numerous training steps for convergence, which can be unfeasible for federated learning. In this study, we apply a pre-training strategy, namely a widely used transfer learning technique, to accelerate the training process and improve the modelling accuracy. The idea behind this is to solve another prediction task, in which the training data are easier to obtain, and when we conduct federated learning for our target task, the pre-trained information can be transferred automatically, particularly the attention part, which share more common parts between different human mobility prediction tasks.

6 EXPERIMENT RESULTS

6.1 Data

“Konzatsu-Tokei (R)" data refers to people flows data collected by individual location data sent from mobile phone with enabled AUTO-GPS function under users’ consent, through the “docomo map navi" service provided by NTT DOCOMO, INC. Those data is processed collectively and statistically in order to conceal the private information. Original location data is GPS data (latitude, longitude) sent in about every a minimum period of 5 minutes and does not include the information to specify individual such as gender or age.

In our experiment, we use the “Konzatsu-Tokei (R)" data in Tokyo (and surrounding area) and Osaka (and their surrounding area) from Oct 2012 to Feb 2013. Our data is separated into training/validation set (from Oct 2012 to Nov 2012) and testing set (From Dec 2012 to Feb 2013) for all our experiments. The basic statistics of the data are shown in Figure 9 (Feb 11 is the National Foundation Day in Japan.).

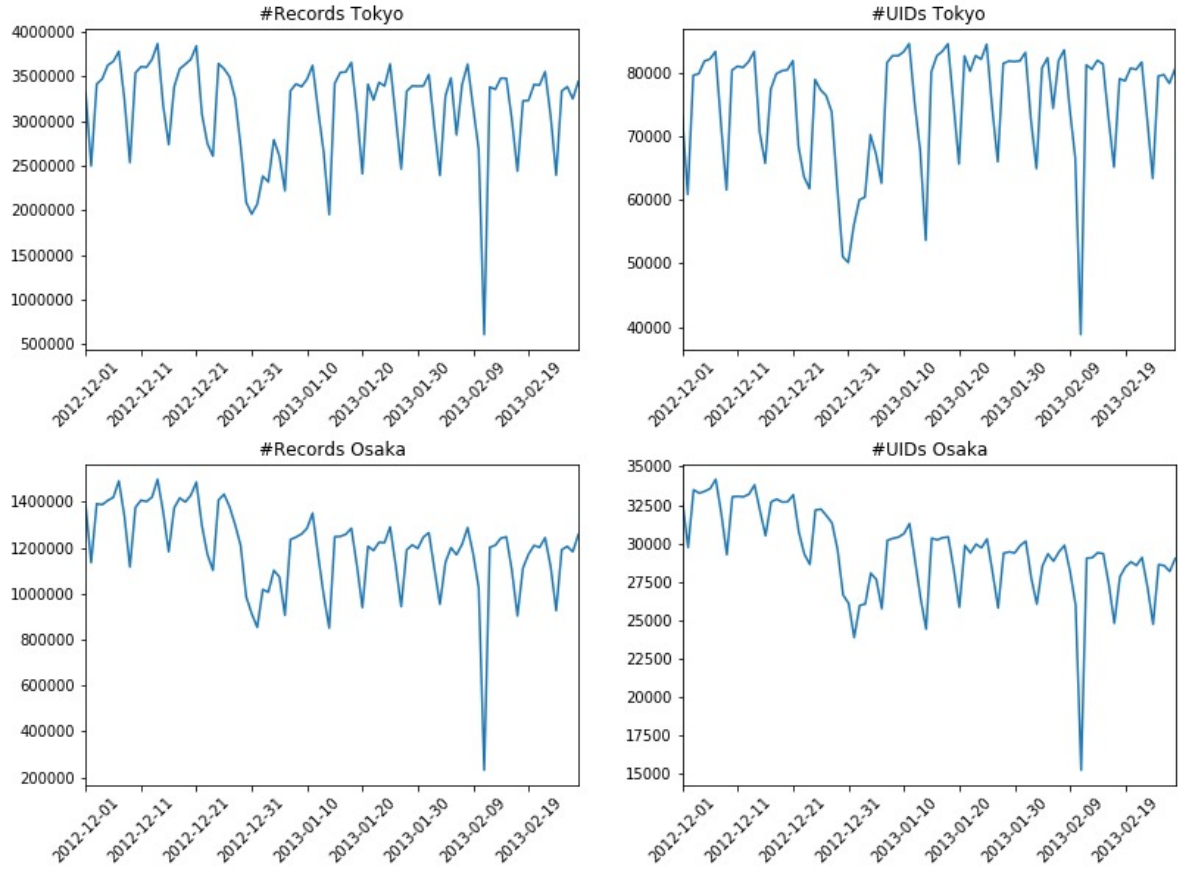

Fig. 9. Number of rows (#Records) and number of unique user IDs (#UIDs) in data of Tokyo and Osaka.

6.2 Experiment Setups

We deployed our algorithm on a deep learning workstation using Intel Xeon E5-2690v4 CPU (2.6GHz 14C 35M 9.60 GT/sec 135W), 2 x TitanX Pascal 12GB GDDR5X graphics card, 128GB (8x 16GB DDR4-2400 ECC RDIMM) and a 1.2TB Intel NVMe DC P3600 Series SSD. The algorithm was implemented in Python, and deep learning models are constructed and inferred using PyTorch 1.0.0 [30].

Experimentally, we set the dimension of the cluster ID embedding to 128, and the time embedding to 32. The GRU has a single layer with a hidden size of 256. For the multi-layer full connected layer of Key or Val, we set the latent dimensions to be 256. The model size of our propose predictor is 9.41 MB without compression.

6.3 Metric

We utilized four metrics to quantitatively evaluate our methods. Cross entropy (CE) is used for measuring the similarities (the lower the better) of the predicted distribution and real movement, and three metrics that are

widely used in the field of ranking and information retrieval:

$$MRR = \frac{1}{|N|}\sum_{i=1}^{|N|}\frac{1}{rank_i}, \qquad Acc@K = \frac{1}{|N|}\sum_{i=1}^{|N|} rank_i \leq K, \qquad Avg.Rank = \frac{1}{|N|}\sum_{i=1}^{|N|} rank_i \tag{11}$$

where $|N|$ is the number of samples, and $rank_i$ is the ranking of the real movement .

These three metrics are widely used in evaluating the performance of the ranking algorithm from different aspects: The mean reciprocal rank (MRR) is focused more on the top-ranked results, which is essential to many recommendation system applications, owing to the decaying effect with respect to the rank. In other words, the difference between ranks 1 and 2 will contribute far more than ranks 100 and 100000 to the metric. Meanwhile, Avg. Rank, with no decaying effect on the ranks, is more suitable for evaluating the performance of the robustness of hard samples. Acc@K shows a more straightforward way to demonstrate the accuracy of including the correct movement in the top-K rankings from our prediction. A higher MRR and Acc@K indicate a better prediction on the top-ranked results, whereas a lower Avg. Rank indicates better general prediction results.

For the applications that we focus more on top-ranked results, such as destination prediction and future-location based advertisement, MRR and Acc@5 are the key indicators of the service performance. For the applications that we pay relatively equal attentions on both the top-ranked and long-tail results, such as pushing traffic information and early warnings (we wish the warnings should reach each user that likely to be effected), CE and Avg. Rank are better indicators because a lower CE and Avg. Rank guarantee a high recall rate if we tolerate a low precision.

6.4 Comparison with Baseline Models

We implement four baseline methods:

(1) Markov: We implement a Markov chain model that predicts the future movement based on the first order Markov chain assumption. This baseline exploits the short-term effects of human mobility movement but does not model the short-term sequential pattern of the most-recent trajectories.
(2) PMM [7]: The periodical mobility model (PMM) is powerful in modeling the periodical patterns of human mobility, particularly the home/work patterns. However, the most-recent trajectories are not well applied in the model. For a comparison with other methods in our experiments, we write a discrete version that analyze the Dirichlet spatial distribution of each user with respect to each time slot.
(3) GRU [8]: The implementation is the same as the short-term predictor introduced in the subsection 5.1. This baseline is more powerful in modeling the most-recent trajectory patterns only. No personalized prediction is taken into consideration.
(4) DeepMove [16]: We implemented a DeepMove model, which is closest to our model, but with a different attention mechanism and user embedding. We used the default setting of DeepMove, which is the average sampling layer in the embedding encode module. We removed the design of user embedding, which includes a large volume of parameters and makes it difficult for decentralized learning and prediction.

As can be seen from Figure 10 and Table 2, in general, our proposed methods "Ours_centralized" (our proposed predictor trained in a centralized way) "Ours_fl" (our proposed predictor trained using federated learning without pre-training,) and "Ours_fl_pretrain" (our proposed predictor trained in a federated learning manner with pre-training) achieve the best performance (almost overlapping). "Ours_fl" does not perform as well as "Ours_centralized" and "Ours_fl_pretrain" in Tokyo whereas for Osaka it is slightly better. As the main reason for this, the predictor in Tokyo is more difficult to train and a federated learning from scratch is difficult to converge within a reasonable number of data communication rounds. In this case, a pre-training strategy is helpful to facilitate the training procedure. This is further discussed in the following subsection.

Interestingly, considering the performance of each model on weekdays and weekends, two opposite patterns are shown. PMM, which relies heavily on the long-term dependency, and our proposed method, which integrates

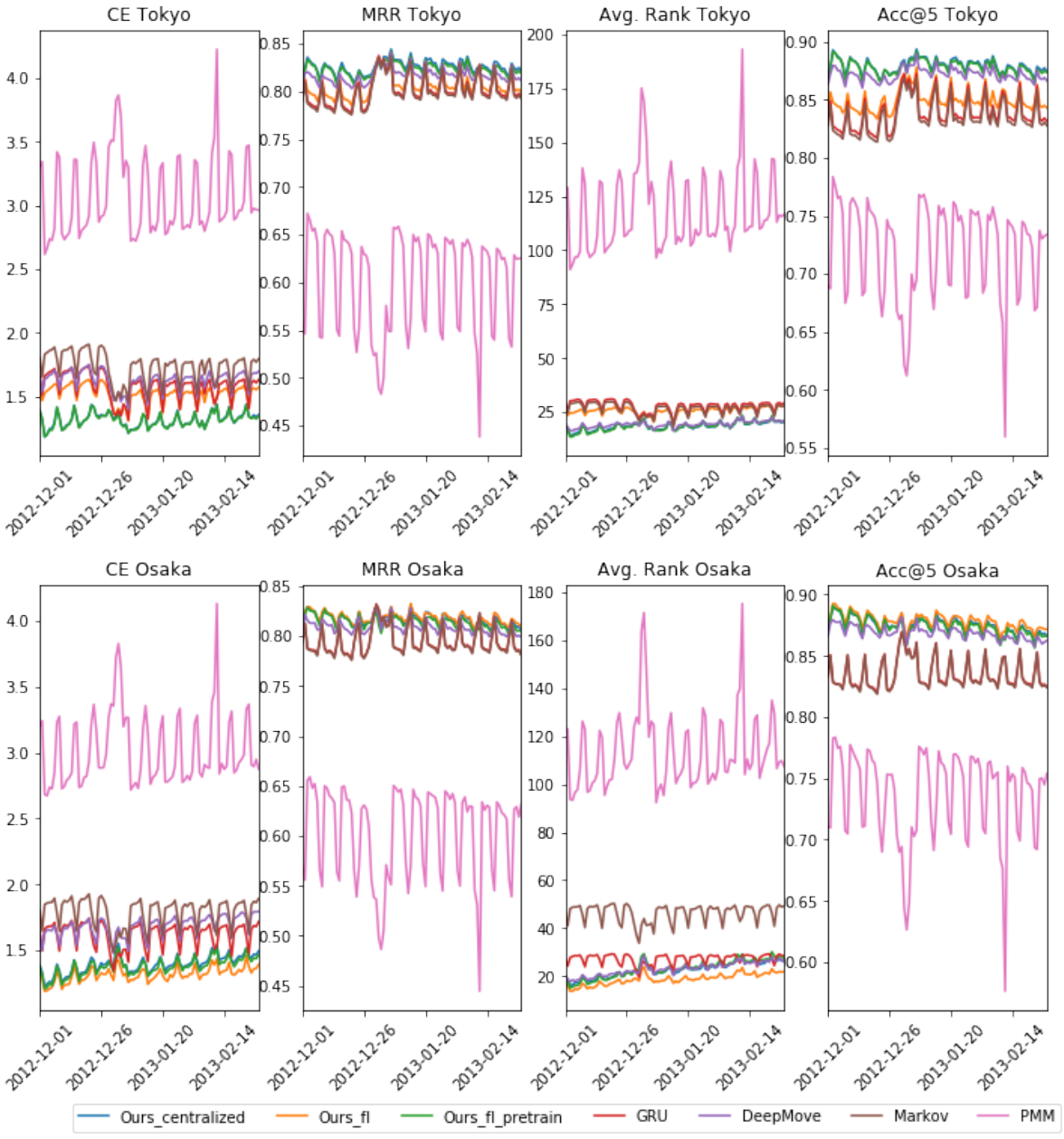

Fig. 10. Comparison of incremental and non-incremental past trajectory database.

both short- and long-term dependencies, achieve a better performance on weekdays because human mobility on weekdays is more periodical and thus predictable, whereas other methods such as the Markov and GRU methods are better at modeling short-term dependency and are thus relatively robust to irregular movements

Table 2. Quantitative evaluation

	Tokyo				Osaka			
	CE	MRR	Avg. Rank	Acc@5	CE	MRR	Avg. Rank	Acc@5
Ours_centralized	1.3149	**0.8258**	18.54	**0.8794**	1.3934	0.8168	22.81	0.8748
Ours_fl	1.5181	0.8064	25.63	0.8497	**1.3174**	**0.8185**	**18.91**	**0.8782**
Ours_fl_pretrain	**1.3106**	0.8232	**18.49**	0.8780	1.3746	0.8141	23.22	0.8737
GRU	1.5755	0.8019	27.79	0.8389	1.6179	0.7956	27.32	0.8352
DeepMove	1.6253	0.8169	19.34	0.8719	1.6800	0.8095	23.18	0.8688
Markov	1.7313	0.7998	26.65	0.8351	1.7875	0.7947	46.35	0.8344
PMM	3.0736	0.6012	118.30	0.7201	3.0233	0.6024	113.89	0.7354

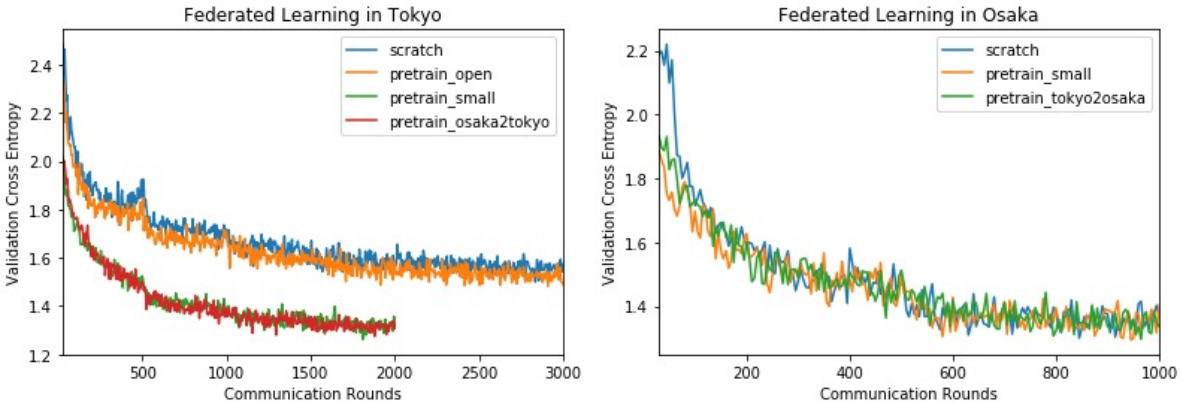

Fig. 11. Validation loss curves for federated learning using different pre-training strategies. (we omit the first 25 communication rounds for the federated learning). The learning rate is halved every 500 communication rounds.

but more sensitive to the movement intensity. The DeepMove method also integrates the short- and long-term dependencies, but the average sampling loses a significant amount of structural information from past trajectories; thus, DeepMove also shows a better performance on weekends. As shown in Figure 10, "Ours_fl" in Tokyo does not work as well as "Ours_centralized" and fails to model the long-term dependency, in particular, which is critical during the weekdays.

6.5 Federated Learning

We follow the basic framework of federated learning introduced in [27], and implement our federated learning strategy shown in Table 8. Training a predictor from scratch requires numerous data communication rounds, which leads to a longer time for model convergence and a higher cost of data communication. We implement the *FedAvg* algorithm introduced in [27], with a one-step update at the device side for each communication round. During our experiments, we found that a single-step update at the device side achieves the best performance for federated learning because the user trajectory data are highly non-IID at the device side, and thus an average of the multi-step updates from the candidate models have difficulty achieving an significant model improvement.

To facilitate our federated learning procedure. we conducted experiments on 3 different pre-training strategies:

(1) **Open** uses an open dataset such as Foursquare check-in data [41]. We use a dataset scraped from the Tokyo area, and process the checkin data to obtain the same data structure introduced in Section 4 for pre-training. This dataset has a completely different method of data collection from our dataset, and the studied area is not exactly the same.
(2) **Small** is pre-trained from a smaller dataset, which could be obtained through crowd sourcing or special agreements with users. In our experiment, we used a smaller dataset of 4096 user IDs for pre-training (approximately 3% of the total number of users in Tokyo and 6% of the total number of users in Osaka) to pre-train the model in a centralized way and use the entire model for federated learning.
(3) **City2City** uses the well-trained model from another city as the pre-trained model to facilitate the federated learning procedure. Recall that, in subsection 4, we assign the cluster IDs through the ranking of the visiting frequency of the grid cells in the city. Thus, although the layouts of the two cities are quite different, the same IDs for both may roughly represent similar locations of the two cities.

The validation loss curve in the federated training phase is plotted in Figure 11. In the case of Tokyo (left), the predictor is difficult to train and learning from scratch takes a undesirably long time for convergence. The "Open" strategy is the easiest way to obtain some pre-training data, and can slightly accelerate the training procedure by lowering the validation by approximately 0.05, or by saving a few hundred of communication rounds. The "Small" strategy has a much more significant effect of accelerating the learning procedure, namely an approximately 10-20 times faster rate than training from scratch. Although a small sample of the entire dataset is insufficient for representing the distribution of all data, and our pre-trained model is easily overfitted to a small sample of data (the validation loss starts from over 2.2), the attention mechanism can be trained to some degree during this pre-training and the learning procedure can be significantly accelerated. "City2City" pre-training strategy shows quite similar acceleration effects. The main reason for this is the attention mechanism, which is the difficult part of training the model and is relatively independent of the city layout, can also be pre-trained to a certain degree even from Osaka. In the case of Osaka (right), because the city is a relatively simpler than Tokyo, even learning from scratch can be converged within a few hundred communication rounds. Our pre-training strategies slightly accelerate the convergence speed, particularly during the early stage of learning.

6.6 Incremental Learning

Under ideal condition, the more past trajectories we obtain, the more personal information we can receive and the more accurate the prediction we can achieve. In this sense, incremental learning is critical to online human mobility. In Figure 12, we compare the results of the incremental and non-incremental versions of our proposed method. For the non-incremental version, we use a fixed past trajectory database from November 2012. For the incremental version, we simply add new trajectories from the previous day to the past trajectory database without updating the model. As we can see, an incremental method significantly outperforms the non-incremental method, particularly when the test date is far from the dates of the past trajectories. The reason for this is two-fold: 1) There are more new users for the non-incremental predictor than for the incremental predictor. In addition, because new users can only use a short-term predictor, a larger portion of new users may lead to a lower prediction accuracy, and 2) more past trajectories can be leveraged for each user to extract richer long-term information.

Note that, although an incremental version works better, in other parts of this paper, we only use the non-incremental version by default because the non-incremental predictor only uses the information from the training set, and thus it is considered fair with other baseline models.

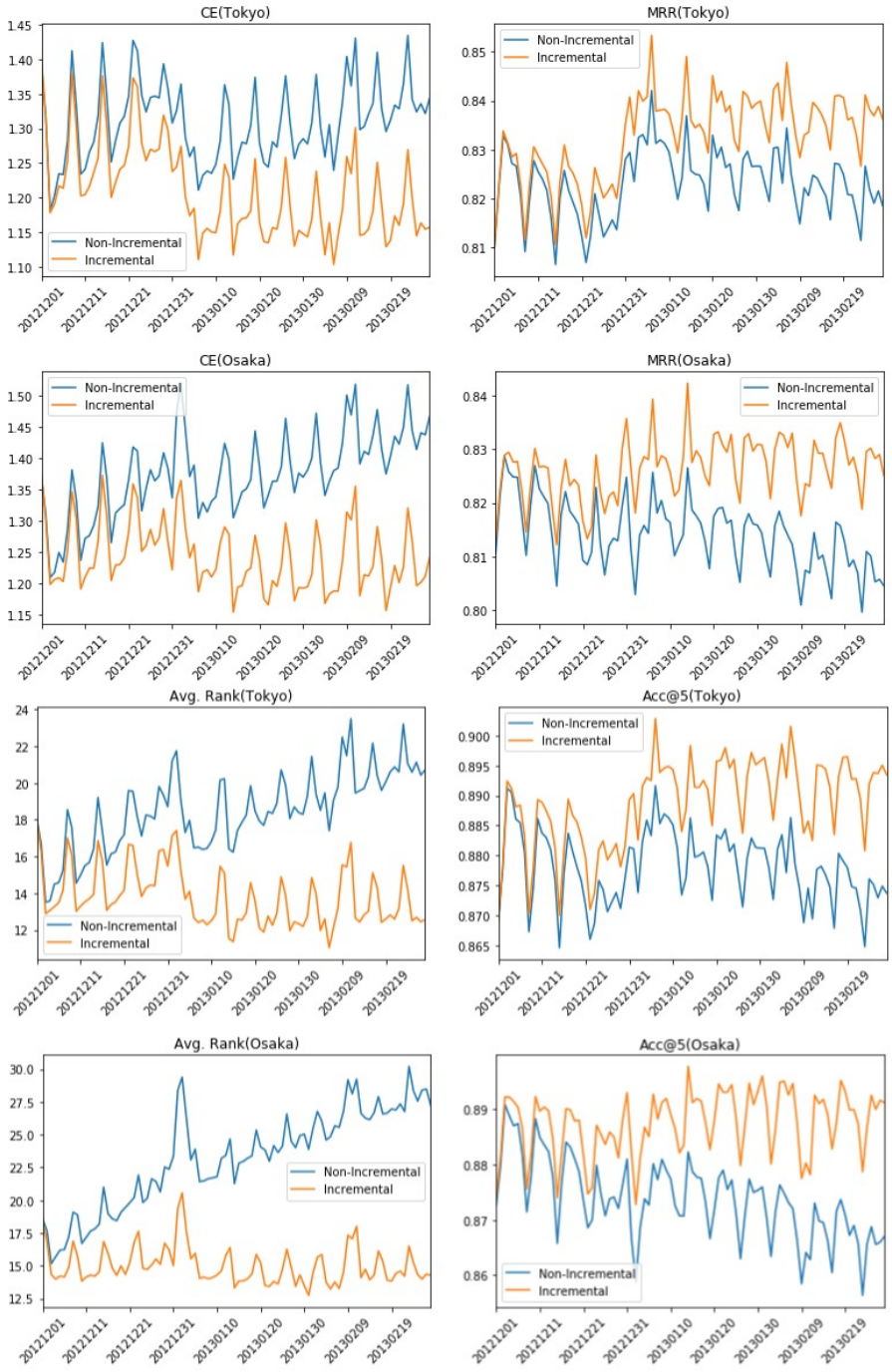

Fig. 12. Comparison of incremental and non-incremental past trajectory database of users.

6.7 Document Trajectory

In our standard method, document trajectories $traj^{doc}_{u,t:t+2\Delta t}$ only include the trajectories at the same time of day in the past trajectories. This is a simplification for reducing the number of document trajectories. However, we also note that other time slots can also provide useful information to improve the prediction accuracy. For example, one user may set off earlier than usual. If we include more document trajectories $\left\{traj^{doc}_{u,t+\tau:t+2\Delta t+\tau} \mid \tau \in [-\tau_{max}, \tau_{max}]\right\}$. We compare the performance of our method with more document trajectories $\tau_{max} > 0$. In Table 3, "Ours-1" represents the standard method and "Ours-3" represents the document trajectory set of $\tau_{max} = 1$. In general, adding more document trajectories can slightly improve our prediction accuracy, but at the expense of extra memory usage and computations.

Table 3. The effects of the time span in the document trajectories

	Tokyo				Osaka			
	CE	MRR	Avg. Rank	Acc@5	CE	MRR	Avg. Rank	Acc@5
Ours-1	1.3149	0.8258	18.54	0.8794	1.3933	0.8168	22.81	0.8748
Ours-3	**1.2886**	**0.8278**	**17.41**	**0.8819**	**1.3633**	**0.8195**	**21.54**	**0.8774**

6.8 Training Data Size & Sampling Rate

We notice that the size of training data and sampling rate will also heavily effect the prediction results. Most of the existing open dataset (e.g. Brightkite [7], Foursquare [41]) do not have either a large number of users in one city or a dense sampling rate of human trajectories. Comparing with our dataset (200,420 users in Tokyo and 80,379 in Osaka in Oct 2012), existing open datasets are far less (below 3% comparing with our Osaka data) number of users concentrated in one city. The prediction accuracy with respect to the size of training set is shown in Figure 13.

A lower sampling rate will over-simplify the real trajectory and lower the utility of real-world applications. The sampling rate of our data is about 18.16 (records per user per day) in Tokyo and 18.61 in Osaka, which is much higher than existing open data (0.83 in [41] and 0.086 in [7]). The predictive cross entropy will rise from 1.3149 to 1.9449 in Tokyo and from 1.3933 to 1.9679 in Osaka (we evaluate on our original data) when we lower down the sampling rate to 10%.

6.9 Day-of-week & Time-of-day

We summarize the cross entropy of our proposed method and GRU (short-term predictor) with respect to different time of day and day of week in Figure 14. Our proposed method can significantly reduce the prediction loss during the morning and evening rush hour. This is because the movements during these times have a more salient periodical pattern, which can be predicted through an attention-based long-term predictor. Similar results can also be found on the right side of Figure 14. Human movements on weekdays are more intense (people are moving more on weekdays) but more periodical, and vice versa. Thus, we can observe that the losses of our proposed method are high on weekends and low on weekdays, whereas the short-term predictor is high on weekdays and low on weekends.

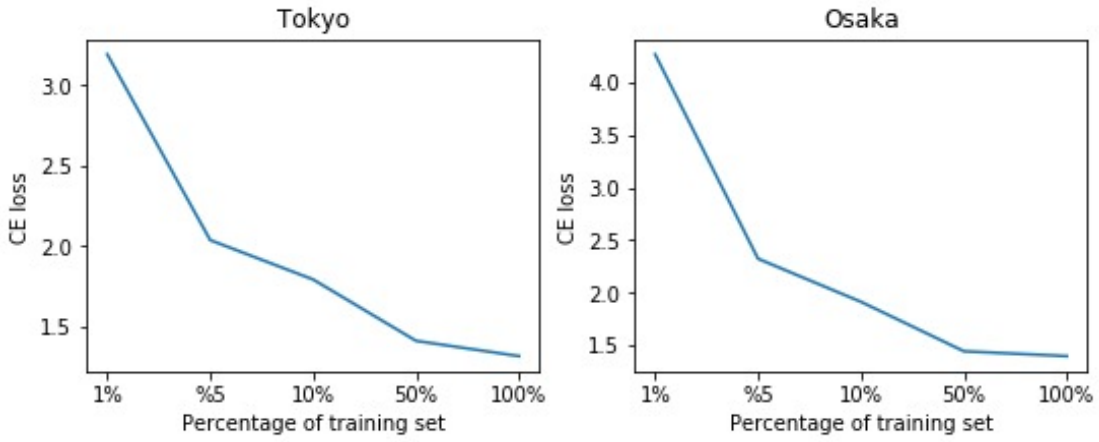

Fig. 13. Cross entropy with respect to training data size.

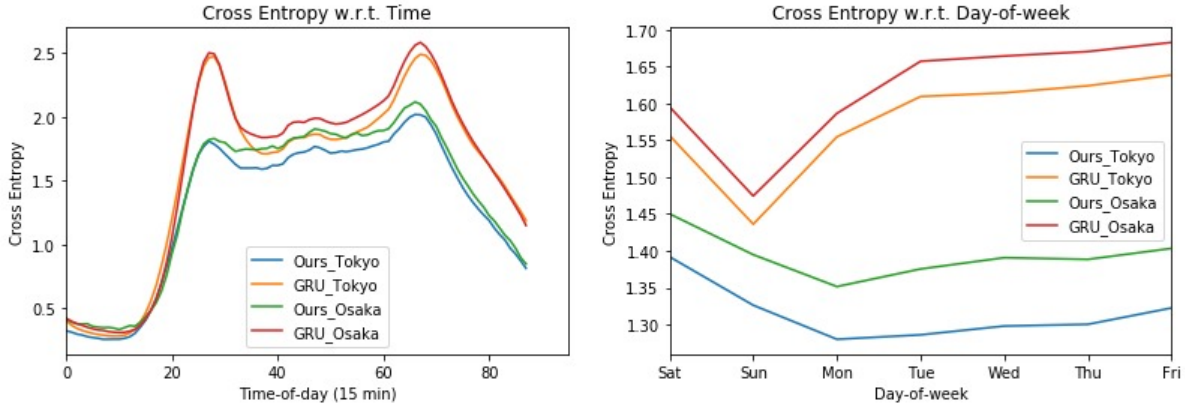

Fig. 14. Cross entropy with respect to time of day and day of week.

6.10 Mobile-device testing

6.10.1 Devices. We test the inference part of our method on two Android devices:

- Huawei MediaPad T3 K0B-W09, with Snapdragon 425 CPU, 2GB memory, 4800 mAh battery, Android 7.0 (API level 24).
- Samsung Galaxy S8 SM-G950FD, with Octa-Core (2 processors: 2.3Ghz Quad-Core Exynos M2 Mongoose, 1.7Ghz Quad-Core ARM Cortex-A53), 4GB memory, 3000mAh battery, Android 9.0 (API level 28)

6.10.2 Implementation Details. As we mentioned in related work, most of existing on-device deep learning libraries have a relatively incomplete support for recurrent neural network and dynamic network structure. Thus in this work, we implement our inference based on a light-weighted Java linear algebra library: Efficient Java Matrix Library (EJML). We export the weights from PyTorch model to text files, and load as matrices and vectors

Table 4. Performance on Android Devices

	Discharging	Discharging (Active)	Memory Usage	Storage Usage	Latency
Huawei MediaPad T3	$0.97 \pm 0.32mA$	$163.1 \pm 28.3mA$	76MB(3.8%)	56.41MB	390ms
Samsung Galaxy S8	$0.73 \pm 0.22mA$	$145.4 \pm 15.7mA$	78MB(2.0%)	53.96MB	248ms

in Android, and breakdown the recurrent neural network and dynamic network computation tasks into basic linear algebra operation and non-linear activation functions.

The predictor is scheduled with an interval of 15 minutes using AlarmManager and BroadcastReceiver in the standard Android SDK, which could still keep running periodically (the time interval will be inexact due to power optimization) even during doze mode.

6.10.3 Performance. The performance of our individual mobility predictor is given in Table 4. Our predictor consumes a negligible battery power and a reasonable amount of memory. The memory usage can be further optimized if we 1) remove the graphical user interface and only provide background service, 2) using adaptive SoftMax layer to reduce the number of parameters, or 3) using the float16 data type rather than float32 and applying more advanced model compression methods such as [23, 49]. Note that, we do not optimize on latency because it is sufficient in many application scenarios. This could be further optimized if we simply pre-load the **Key** and **Val** of each document trajectory, and an acceleration of more than 30%.

7 CONCLUSION & FUTURE WORK

In this study, we proposed a novel human mobility predictor that can achieve both a personalized prediction and training in a decentralized way. We modeled the personalized human mobility prediction as a few-shot learning problem and proposed an attention-based human mobility prediction method to make it possible to share model weights among the prediction task of personalized prediction for each user. In this way, we conducted a federated learning paradigm and applied pre-training strategies to conduct decentralized training for a personalized predictor; our experiments show that a pre-training strategy can significantly boost the speed of training for relatively difficult task (training the predictor in Tokyo).

As a limitation of this study, our current predictor is not sufficiently robust for irregular human mobility, particularly during a large event. A more sophisticated method of communicating and integrating irregularities at the city scale in an online way could be an interesting future direction. Another future direction of this study could be further reducing the data communication and time cost for the federated learning procedure. Even a pre-training process, we still require over 1,000 data communication rounds to reach optimality. Recent research on meta-learning, which studies how to use machine learning techniques to facilitate the learning process, could also be a promising direction for accelerating the federated learning procedure.

ACKNOWLEDGMENTS

This work was partially supported by Leading Initiative for Excellent Young Researchers (LEADER) Program and Grant in-Aid for Scientific Research B (17H01784) of Japan's Ministry of Education, Culture, Sports, Science, and Technology (MEXT); and JST, Strategic International Collaborative Research Program (SICORP).

REFERENCES

[1] Yuan Ai, Mugen Peng, and Kecheng Zhang. 2018. Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks* 4, 2 (2018), 77–86.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 961–971.

[3] B. Altaf, L. Yu, and X. Zhang. 2018. Spatio-Temporal Attention based Recurrent Neural Network for Next Location Prediction. In *2018 IEEE International Conference on Big Data (Big Data)*. 937–942. https://doi.org/10.1109/BigData.2018.8622218

[4] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. http://arxiv.org/abs/1902.01046 cite arxiv:1902.01046.

[5] Longbiao Chen, Dingqi Yang, Daqing Zhang, Cheng Wang, Jonathan Li, and Thi-Mai-Trang Nguyen. 2018. Deep mobile traffic forecast and complementary base station clustering for C-RAN optimization. *Journal of Network and Computer Applications* 121 (2018), 59 – 69. https://doi.org/10.1016/j.jnca.2018.07.015

[6] Longbiao Chen, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Kostadin Kushlev, Wangsheng Zhang, and Shijian Li. 2015. Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 571–575. https://doi.org/10.1145/2750858.2804291

[7] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 1082–1090. https://doi.org/10.1145/2020408.2020579

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. http://arxiv.org/abs/1412.3555 cite arxiv:1412.3555Comment: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop.

[9] Japanese Industrial Standards Committee. 1976. Grid Square Code. http://www.jisc.go.jp/app/jis/general/GnrJISNumberNameSearchList?toGnrJISStandardDetailList

[10] Trevor Darrell, Marius Kloft, Massimiliano Pontil, Gunnar Rätsch, and Erik Rodner. 2015. Machine Learning with Interdependent and Non-identically Distributed Data (Dagstuhl Seminar 15152). In *Dagstuhl Reports*, Vol. 5. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[11] Cynthia Dwork. 2008. Differential privacy: A survey of results. In *In Theory and Applications of Models of Computation*. Springer, 1–19.

[12] EU. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation). https://eur-lex.europa.eu/legal-content/EN/TXT

[13] Zipei Fan, Xuan Song, Ryosuke Shibasaki, Tao Li, and Hodaka Kaneda. 2016. CityCoupling: bridging intercity human mobility. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 718–728.

[14] Zipei Fan, Xuan Song, Tianqi Xia, Renhe Jiang, Ryosuke Shibasaki, and Ritsu Sakuramachi. 2018. Online Deep Ensemble Learning for Predicting Citywide Human Mobility. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 105 (Sept. 2018), 21 pages. https://doi.org/10.1145/3264915

[15] Biyi Fang, Xiao Zeng, and Mi Zhang. 2018. Nestdnn: Resource-aware multi-tenant on-device deep learning for continuous mobile vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 115–127.

[16] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1459–1468. https://doi.org/10.1145/3178876.3186058

[17] Jie Feng, Mingyang Zhang, Huandong Wang, Zeyu Yang, Chao Zhang, Yong Li, and Depeng Jin. 2019. DPLink: User Identity Linkage via Deep Neural Network From Heterogeneous Mobility Data. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 459–469. https://doi.org/10.1145/3308558.3313424

[18] Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. *CoRR* abs/1712.07557 (2017). arXiv:1712.07557 http://arxiv.org/abs/1712.07557

[19] Aris Gkoulalas-Divanis, Panos Kalnis, and Vassilios S. Verykios. 2010. Providing K-Anonymity in Location Based Services. *SIGKDD Explor. Newsl.* 12, 1 (Nov. 2010), 3–10. https://doi.org/10.1145/1882471.1882473

[20] Google. 2019. Android NNAPI. https://developer.android.com/ndk/guides/neuralnetworks

[21] Google. 2019. TensorFlow Lite. https://www.tensorflow.org/lite

[22] Marco Gramaglia and Marco Fiore. 2015. Hiding Mobile Traffic Fingerprints with GLOVE. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT '15)*. ACM, New York, NY, USA, Article 26, 13 pages. https://doi.org/10.1145/2716281.2836111

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[24] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *CoRR* abs/1704.04861 (2017). arXiv:1704.04861 http://arxiv.org/abs/1704.04861

[25] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the 22Nd ACM International Conference on*

Multimedia (MM '14). ACM, New York, NY, USA, 675–678. https://doi.org/10.1145/2647868.2654889

[26] Tom Kenter, Alexey Borisov, Christophe Van Gysel, Mostafa Dehghani, Maarten de Rijke, and Bhaskar Mitra. 2017. Neural Networks for Information Retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 1403–1406. https://doi.org/10.1145/3077136.3082062

[27] Jakub KoneÄŀnÃ¡, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*. https://arxiv.org/abs/1610.05492

[28] Tatsuya Konishi, Mikiya Maruyama, Kota Tsubouchi, and Masamichi Shimosaka. 2016. CityProphet: City-scale Irregularity Prediction Using Transit App Logs. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 752–757. https://doi.org/10.1145/2971648.2971718

[29] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML'10)*. Omnipress, USA, 807–814. http://dl.acm.org/citation.cfm?id=3104322.3104425

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[31] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. 2018. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78 (2018), 680–698.

[32] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. 2011. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*. Springer, 152–169.

[33] Xuan Song, Xiaowei Shao, Huijing Zhao, Jinshi Cui, Ryosuke Shibasaki, and Hongbin Zha. 2010. An online approach: Learning-semantic-scene-by-tracking and tracking-by-learning-semantic-scene. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 739–746.

[34] Latanya Sweeney. 2002. Achieving K-anonymity Privacy Protection Using Generalization and Suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10, 5 (Oct. 2002), 571–588. https://doi.org/10.1142/S021848850200165X

[35] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin. 2017. Beyond K-Anonymity: Protect Your Trajectory from Semantic Attack. In *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2017.7964921

[36] Liang Wang, Zhiwen Yu, Bin Guo, Tao Ku, and Fei Yi. 2017. Moving Destination Prediction Using Sparse Dataset: A Mobility Gradient Descent Approach. *ACM Trans. Knowl. Discov. Data* 11, 3, Article 37 (April 2017), 33 pages. https://doi.org/10.1145/3051128

[37] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. 2015. Regularity and conformity: Location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1275–1284.

[38] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1241–1250. https://doi.org/10.1145/3038912.3052620

[39] W Kang Z Li FY Wang Y Lv, Y Duan. 2015. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2):865-873* (2015).

[40] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 2147–2157. https://doi.org/10.1145/3308558.3313635

[41] Dingqi Yang, Daqing Zhang, Vincent. W. Zheng, and Zhiyong Yu. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (Jan. 2019), 19 pages. https://doi.org/10.1145/3298981

[43] Zimo Yang, Defu Lian, Nicholas Jing Yuan, Xing Xie, Yong Rui, and Tao Zhou. 2017. Indigenization of urban mobility. *Physica A: Statistical Mechanics and its Applications* 469 (2017), 232–243.

[44] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from Multiple Cities: A Meta-Learning Approach for Spatial-Temporal Prediction. In *The World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 2181–2191. https://doi.org/10.1145/3308558.3313577

[45] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Li Zhenhui. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *The Thirty-Second AAAI Conference on Artificial Intelligence*.

[46] Shanhe Yi, Zhengrui Qin, and Qun Li. 2015. Security and privacy issues of fog computing: A survey. In *International conference on wireless algorithms, systems, and applications*. Springer, 685–695.

[47] Jiafan Zhang, Bin Guo, Qi Han, Yi Ouyang, and Zhiwen Yu. 2016. CrowdStory: Multi-layered Event Storyline Generation with Mobile Crowdsourced Data. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*

(UbiComp '16). ACM, New York, NY, USA, 237–240. https://doi.org/10.1145/2968219.2971406

[48] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[49] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6848–6856.

[50] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

[51] Jingbo Zhou, Hongbin Pei, and Haishan Wu. 2016. Early Warning of Human Crowds Based on Query Data from Baidu Map: Analysis Based on Shanghai Stampede. *CoRR* abs/1603.06780 (2016). http://arxiv.org/abs/1603.06780

[52] Xiao Zhou, Anastasios Noulas, Cecilia Mascoloo, and Zhongxiang Zhao. 2018. Discovering Latent Patterns of Urban Cultural Interactions in WeChat for Modern City Planning. *arXiv preprint arXiv:1806.05694* (2018).

[53] Gergely ÃĄcs and Claude Castelluccia. 2014. A case study: privacy preserving release of spatio-temporal density in paris. In *KDD*. ACM, 1679–1688. https://doi.org/10.1145/2623330.2623361