This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3132056, IEEE Internet of Things Journal

Towards Crowdsourced Transportation Mode Identification: A Semi-Supervised Federated Learning Approach

Chenhan Zhang, *Student Member, IEEE,* Yuanshao Zhu, Christos Markos, Shui Yu, *Senior Member, IEEE*, and James J.Q. Yu, *Senior Member, IEEE*

***Abstract*—Privacy-preserving Transportation Mode Identification (TMI) is among the key challenges towards future intelligent transportation systems. With recent developments in federated learning (FL), crowdsourcing has emerged as a promising cost-effective data source for training powerful TMI classifiers without compromising users' data privacy. However, existing TMI approaches have relied heavily on the availability of transportation mode labels, which is often limited in real-world applications. While recent semi-supervised studies have partially addressed this issue by assigning pseudo-labels to unlabeled data, such practice often degrades classification performance as more unlabeled data are incorporated. In response to this issue, we present a semi-supervised FL scheme for TMI termed Mean Teacher Semi-Supervised Federated Learning (MTSSFL). MTSSFL trains a deep neural network ensemble under a novel semi-supervised FL framework, achieving highly accurate and privacy-protected crowdsourced TMI without depending on the availability of massive labeled data. MTSSFL introduces consistency-updating to insert the global model in the gradient updates of the local models that only have unlabeled data to improve their training. We also devise *mean-teacher-averaging*, a secure parameter aggregation mechanism that further boosts the global model's TMI performance without requiring additional training. Our extensive case studies on a real-world dataset demonstrate that MTSSFL's classification accuracy is merely 1.1% lower than the state-of-the-art semi-supervised TMI approach while being the only one to satisfy FL's privacy-preserving constraints. In addition, MTSSFL can achieve high accuracy with less training overhead due to the proposed semi-supervised learning design.**

***Index Terms*—Federated Learning, Semi-supervised Learning, Crowdsourcing, Transportation Mode Identification, Intelligent Transportation Systems**

I. Introduction

Transportation Mode Identification (TMI) aims to infer transportation modes from users' mobility data. As a core application of Intelligent Transportation Systems (ITSs), accurate TMI can help address a variety of transportation-related problems, such as public transportation planning, route recommendations, and traffic signal optimization [1]. With recent developments in data-driven analysis, particularly in machine and deep learning algorithms, Internet-of-Things (IoT)-enabled ITSs have attracted significant research interest due to their capacity for capturing massive quantities of data [2]. The widespread presence of Global Positioning System (GPS) sensors in modern smartphones and wearable devices have streamlined the acquisition of diverse user trajectories, paving the way for IoT-enabled TMI [3].

Training highly accurate deep learning models primarily requires immense amounts of data. Data crowdsourcing seeks to alleviate this issue by allowing mutiple users to collectively generate large datasets in a distributed manner [4]. A centralized server is always set to receive the crowdsourced data and train them. These data are usually transmitted to the centralized server in unprocessed forms, however, information implied by the data may have a strong connection to the users' privacy, which raises a privacy concern; for instance, GPS trajectories are tied to users' location and transportation history. There exist techniques for addressing such privacy concerns, including encrypted evaluation, data anonymization, and noise injection [5]. Nevertheless, the computational complexity of such methods often prohibits their use in real-world applications [6], with noise injection possibly degrading the performance of deep learning models over time.

Federated Learning (FL) provides a privacy-preserving framework for deep learning to realize the benefits of crowdsourcing [7]. It allows users to share their data without jeopardizing their privacy, provided that the data are only stored and used locally. Users only need to intermittently share their local model gradient updates with the centralized server, which in turn manages the collective training process. Concretely, the centralized server trains the global model by aggregating the local models' gradients and subsequently broadcasts the updated model parameters to all users. Each user uploads their local model to the server and then downloads the global model to perform offline inference with the cloud-distributed model. FL has rapidly sparked tremendous interest within the IoT-enabled ITS community due to its privacy-preserving nature and the efficient use of computing power provided by edge devices. Successful applications can be seen in urban traffic forecasting [8], [9], rail traffic control [10], vehicular networking [11], and others. Following this trend, we envision that FL

Corresponding author: James J.Q. Yu (email: yujq3@sustech.edu.cn)

This work is supported by the Stable Support Plan Program of Shenzhen Natural Science Fund No. 20200925155105002, by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032, by the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation No. 2020B121201001, and by Australia ARC DP200101374 and LP190100676.

Chenhan Zhang, Yuanshao Zhu, Christos Markos, and James J.Q. Yu are with the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Chenhan Zhang and Christos Markos are also with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. Shui Yu is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Australia.

will also be successful for crowdsourced TMI; however, some practical concerns need to be addressed first.

Existing TMI research typically employs machine or deep learning models in fully supervised settings, thereby assuming the availability of sufficient GPS trajectories labeled by transportation mode at training time [12]–[15]. This assumption, however, can not hold for real-world crowdsourced TMI applications. Even though GPS sensors can readily capture user movement without requiring human intervention, they have no knowledge of the corresponding transportation mode labels: these would have to be provided by the users themselves to ensure correctness. Nonetheless, manual annotation is both time-consuming and labor-intensive for users. While FL offers a variety of incentive and reward mechanisms, they can impose a significant financial or computational burden on task publishers in real crowdsourcing systems [16], [17]. Therefore, large volumes of unlabeled trajectories are often left unused, despite being no less useful than their labeled counterparts.

To close the research gap in existing FL-based approaches for crowdsourced TMI, we propose a novel semi-supervised federated learning scheme towards FL-based crowdsourcing TMI tasks. Particularly, we devise a semi-supervised FL framework, Mean Teacher Semi-Supervised Federated Learning (MTSSFL), to address the issues caused by the lack of labeled data in crowdsourced TMI. Concretely, MTSSFL leverages *consistency-updating*, which we introduce to include the global model in the gradient updates of local models that only possess unlabeled data. Such a teacher-student learning scheme, in conjunction with existing pseudo-labeling approaches, can considerably improve the local models' training on their own unlabeled data. Furthermore, we propose *mean-teacher-averaging* to replace conventional secure parameter aggregation mechanisms, in order to form a better global model without requiring additional training. Additionally, specific to the TMI task, we devise an ensemble of spatial-temporal deep neural networks for feature extraction from GPS trajectory data and achieve transportation mode identification. The main contributions of this paper are summarized as follows:

- We propose MTSSFL, a new FL-based semi-supervised learning scheme that incorporates an ensemble-learning-based TMI model. MTSSFL can effectively address the model training issues caused by the few labeled data typically available for crowdsourced, privacy-preserving TMI. The involved semi-supervised federated learning approach of MTSSFL can also serve as a generic solution to other proper applications.
- We introduce consistency-updating, a novel approach for training local models that only possess unlabeled data guided by the centralized model trained on few labeled data.
- We design an Exponential Moving Average (EMA)-based secure parameter aggregation mechanism termed *mean-teacher-averaging* to improve the global model without additional training.
- We conduct extensive case studies to assess the performance of MTSSFL on IID and non-IID data. We also investigate its robustness to different hyperparameter configurations, and provide guidelines into hyperparameter selection.

The remainder of this paper is organized as follows. Section II summarizes related work in TMI and semi-supervised learning. Section III defines the problems of GPS-based TMI and semi-supervised FL for TMI. Section IV details our data pre-processing techniques applied to raw GPS trajectory data, as well as the proposed TMI model and MTSSFL framework. Section V conducts a comprehensive series of case studies to demonstrate the effectiveness of the proposed approach and investigate its sensitivity to hyperparameter variations. Section VI discusses the traits regarding security and privacy, and generalization ability of MTSSFL. Finally, Section VII concludes this paper.

II. Related Work

A. Transportation Mode Identification

Contemporary TMI approaches can be broadly categorized into machine- and deep-learning-based. Their input data sources include but are not limited to GPS, accelerometer, and gyroscope sensors, often combined with Geographic Information System (GIS) information such as proximity to bus or metro stations [18]. Among the above, GPS sensors are arguably the most popular sources due to their rich spatial-temporal information, easy acquisition, and lower communication costs [19]. As such, we focus on GPS-based works in the remainder of this section.

Feature extraction and classification are two main sub-tasks in the GPS-trajectory-based TMI approaches. The design of feature extraction can be regarded as a distinguishing factor among different TMI approaches. In [12], the authors proposed a two-step identification scheme that has been widely adopted in utilizing GPS trajectory for TMI [20], [21]. In this scheme, trajectories are first partitioned into single-transportation-mode segments based on domain knowledge. Subsequently, a set of hand-crafted features are extracted for each segment and fed to machine learning models for classification.

More recently, the success of deep neural networks in research areas such as computer vision and natural language processing has led to their adoption for TMI. In this direction, Dabiri *et al.* [13] trained Convolutional Neural Network (CNN) ensembles, while Jeyakumar *et al.* [22] employed recurrent neural networks with the Long Short-Term Memory (LSTM) module to exploit the temporal dependencies within GPS trajectories and improve identification accuracy. Yu [19] leveraged a deep LSTM ensemble to cope with the few-shot data problem.

B. Privacy-preserving Transportation Mode Identification

Since GPS trajectory analysis may reveal individuals' personal information, the issue of privacy preservation in TMI has gradually attracted increasing attention [23]. Traditional approaches to privacy-preserving TMI have typically been based on cryptography [24], [25]. However, the required data encryption technology can be computationally expensive, especially on large-scale data. On the other hand, with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3132056, IEEE Internet of Things Journal

ever-growing restrictions brought forth by data privacy regulations (e.g., General Data Protection Regulation[1]), third-party organizations such as mobile communications operators are not allowed to collect or share users' personal data [26]. Since performing machine learning at scale usually requires sharing massive amounts of data among multiple public organizations or private companies, such regulations challenge the feasibility of existing approaches.

The emerging (FL) paradigm greatly breaks through this dilemma, whose decentralized learning strategy enables data can be trained locally at different organizations without exchange [27]. FL also achieves the trade-off between model performance and privacy-preserving, and great successes have been witnessed in various studies [28], [29]. Liu *et al.* [30] proposed a FL framework for traffic flow prediction, who first introduce FL to ITS research. Zhu *et al.* [31] devised a FL-based approach for TMI considering the non-IID data of GPS trajectories.

C. Semi-Supervised Deep Learning

Recent years have witnessed a wide adoption of semi-supervised deep learning approaches aiming to jointly train neural networks on few labeled and massive unlabeled data, spanning research areas such as computer vision and natural language processing problems. These approaches can be divided into roughly three categories [32].

The first category combines unsupervised pre-training with supervised fine-tuning [33]. Concretely, the pre-training phase trains the classifier on unlabeled data in an unsupervised manner; fine-tuning then trains the supervised component of the model on the available labeled data. The second category indirectly constructs semi-supervised algorithms based on latent features extracted from the model [34]. After the classifier is trained on the available labeled data in a supervised manner, the inference is performed on the unlabeled data. The predicted labels are then manually assessed, and the correctly classified ones are added to the training set for model training. While labeled and unlabeled data are both used in the previous two approaches, the model is ultimately trained in a supervised manner. The third category of semi-supervised approaches involves training models in a truly semi-supervised fashion. As one of the representatives of this category, Lee [35] devised "pseudo-labels" for unlabeled data by selecting the class having the maximum predicted probability as pseudo-ground-truth. Laine *et al.* [36] proposed an approach based on the moving average of the predicted labels in each training iteration, aiming to construct a better target. The target is then used to estimate the unsupervised loss and update the model. Following [36], the authors in [37] proposed a promising approach termed *mean teacher*, which averages model weights instead of label predictions.

In the field of TMI, Yazdizadeh *et al.* [38], and Dabiri *et al.* [21] recently proposed two semi-supervised deep learning approaches based on Generative Adversarial Networks (GANs) and convolutional autoencoders. Yu [19] instead trained a semi-supervised LSTM ensemble on different views of the data. While these semi-supervised approaches have been shown to be effective, they may not be able to satisfy several FL requirements [39]. First, in FL systems, the optimization objectives among the local models differ due to the relatively different distributions of their respective trained data. The averaging aggregation at the central server can contribute to develop a generic model from these local models effectively, mitigating these isolated training processes' negative effect to a certain extent. However, in semi-supervised learning contexts especially for those involved with dummy labels [35], the data distribution differences among different clients are more significant and intricate. The conventional updating and averaging approaches of FL are incapable of capturing these information timely, which makes it difficult to maintain the consistency between the client models and global models; this is a big challenge for semi-supervised federated learning. Second, the assumption of imbalanced label distributions for semi-supervised learning has not been well-studied in the federated learning context where most of them only considered mild conditions. Particularly, extreme conditions (e.g., only a small number of labeled data can be utilized or the labeled data only exist at the central server) were not involved in the previous researches. Last but not least, some of the state-of-the-art semi-supervised approaches were only designed and evaluated on simple classification tasks (e.g., [40]–[42]), which might not be applicable to practical and complex ones such as the investigated crowdsourcing TMI tasks in this work. Considering the traits of federated learning under few labeled data and the requirement of crowdsourcing tasks, in this paper we propose a novel semi-supervised FL framework for crowdsourced TMI based on the *mean teacher* approach [37] to fill the research gap.

III. Preliminaries

In this section, we first formulate the task of GPS-based transportation mode identification. We then delve into the problems of crowdsourced federated learning and learning from non-IID data.

A. GPS-based Transportation Mode Identification

In this work, transportation mode identification is performed on features extracted from GPS trajectories. The latter are represented as chronologically ordered sequences of discrete GPS records, with each record (or point) being defined by the following four attributes: *latitude* (lat), *longitude* (lng), *timestamp*, and *label*.

Following established TMI research [12], [13], [19], we aim to identify the following five transportation modes: *walk*, *bus*, *bike*, *driving*, and *train*. All GPS trajectories are first partitioned into segments such that each corresponds to exactly one transportation mode. Since raw GPS trajectories are ill-suited for training machine or deep learning models [12], we then preprocess them into motion and other features following standard TMI practice [12], [13], [19], as will be detailed in Section IV-A.

Let $\mathcal{F}_i$ denote the calculated features of the i-th GPS point in a segment. We can then reform GPS segment k of length

[1] https://gdpr-info.eu/

T as $\mathrm{GPS}_k = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T, \mathrm{label}_k\}$, where label_k is the corresponding transportation mode of the segment. Our aim is to train a deep learning classifier to classify GPS_k, $\forall k$, by transportation mode. This is formulated as

$$\mathrm{GPS}_k[\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_T] \xrightarrow{f(\cdot)} \mathrm{GPS}_k[\mathrm{label}_k], \tag{1}$$

where $f(\cdot)$ denotes the classification function learned by the classifier model.

B. Semi-supervised Federated Learning

1) ***Crowdsourced Federated Learning Framework****:* In this work, we propose a federated learning framework for crowdsourced TMI. We use the term "publisher" to refer to the initiator of the crowdsourcing task, who also owns a central server. We refer to the local (distributed) entities as "workers". Let q denote the total number of workers; we then have the set of workers $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$. Each worker C_i uses their respective database D_i to store their sensed GPS trajectories, resulting in the database set $\mathcal{D} = \{D_1, D_2, \ldots, D_q\}$. In FL, worker C_i uses their locally-stored data to train their local model $M_i \in \mathcal{M} = \{M_1, M_2, \ldots, M_q\}$, where the learned parameters of M_i are denoted by ϕ_i.

This work assumes that both the publisher and the workers are reliable and low-latency communicators. Both are also considered to be honest, which means that they will strictly execute FL protocols and will not try to infer other entities' data from the shared model parameters. Additionally, this work assumes that the security level of uploading channels is higher than that of broadcasting channels [5].

2) ***IID and non-IID Data****:* Each worker is likely to have more data corresponding to some transportation modes than others. This will inevitably lead to a skewed local class distribution, i.e., non-Independent and Identically Distributed (non-IID) data. Previous research indicated the deviating FL performance on IID and non-IID data, where the latter usually renders the FL systems to develop inferior training performance due to the data imbalance among different clients [43]. To explore how non-IID data affect the classification performance of our approach, we introduce the metric R to measure the level of non-IID. Specifically, the class distribution of database D_i belonging to worker C_i is defined as $P_i = [p_1, p_2, \ldots, p_c] \in \mathbb{R}^c$, where p_j is the fraction of the j-th class in D_i and c denotes the total number of classes. R can be defined as:

$$R = \frac{1}{2} \sum_{1 \leqslant i < o \leqslant q} \|P_i - P_o\|_1 \frac{1}{q(q-1)/2}, \tag{2}$$

where $\|\cdot\|_1$ denotes the L_1 norm, $\|P_i - P_o\|_1$ is the variation distance, $q(q-1)/2$ corresponds to the total number of worker pairs. Please note that the first factor $\frac{1}{2}$ is merely used to ensure $R \in [0, 1]$. We obtain $R = 0$ when each worker has a uniform class distribution, i.e., $P_i = [\frac{1}{c}, \frac{1}{c}, \ldots, \frac{1}{c}]$, while $R = 1$ means that each worker only has trajectories belonging to one transportation mode class.

IV. Methodology

In this section, we introduce the proposed scheme for crowdsourced TMI. We start by presenting the data pre-processing approach adopted to handle raw GPS data. Next, we detail the employed TMI model. Finally, we elaborate on the proposed semi-supervised federated learning framework.

A. Representation of TMI Data Features

Raw GPS data pre-processing is the prerequisite procedure of GPS-trajectory-based TMI, which consists of two steps, namely, GPS trajectory pre-processing and data feature representation.

1) ***GPS Trajectory Pre-processing****:* Raw GPS trajectories are usually denoted as chronologically ordered series of GPS records. To segment each trajectory by transportation mode, we follow the trajectory segmentation algorithm introduced in [12], which has since been widely used in the transportation literature [19], [21]. This segmentation method is based on the intuition that the transportation mode separating any other two has to be *walk*. For instance, if a user traveling by train intends to board a bus, they are bound to walk from the former to the latter. As such, this method first classifies each GPS point as *walk* or *non-walk* according to velocity and acceleration thresholds before forming single-transportation-mode segments by aggregating adjacent points with the same predicted label. Using the same thresholds defined by [12], we split all trajectories into a total of T^* single-transportation-mode segments $\{\mathrm{GPS}_1, \mathrm{GPS}_2, \ldots, \mathrm{GPS}_{T^*}\}$.

2) ***Motion Feature Extraction****:* We follow established TMI research [12], [13], [19] in extracting three pointwise motion-related features, i.e., speed, acceleration, and jerk. To do so, we first calculate the relative distance between every two consecutive GPS records using the Vincenty Formula [44], which can be denoted as:

$$d_i = \mathrm{Vincenty}\left(\mathrm{lat}_i, \mathrm{lng}_i; \mathrm{lat}_{i+1}, \mathrm{lng}_{i+1}\right). \tag{3}$$

Based on d_i, we can then estimate speed s_i, acceleration a_i, and jerk j_i for the i-th GPS point as follows:

$$s_i = \frac{d_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad s_T = s_{T-1}, \tag{4a}$$

$$a_i = \frac{s_{i+1} - s_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad a_T = 0, \tag{4b}$$

$$j_i = \frac{a_{i+1} - a_i}{\Delta t_i}, \quad 1 \leq i \leq T, \quad j_T = 0, \tag{4c}$$

where T is the number of GPS records in a GPS segment. In this way, a motion feature vector $x_i \equiv \mathcal{F}_i = (d_i, s_i, a_i, j_i)$ can be extracted for the i-th GPS point. These 4-channel feature vectors are then stacked into a tensor for each trajectory segment and used as input to the TMI model detailed in the sequel.

B. TMI Model

In this work, we build our TMI model based on DNNs and ensemble learning techniques. Ensemble learning technique can deal with the bias and variance developed by conventional single-model networks and develop high-accuracy prediction

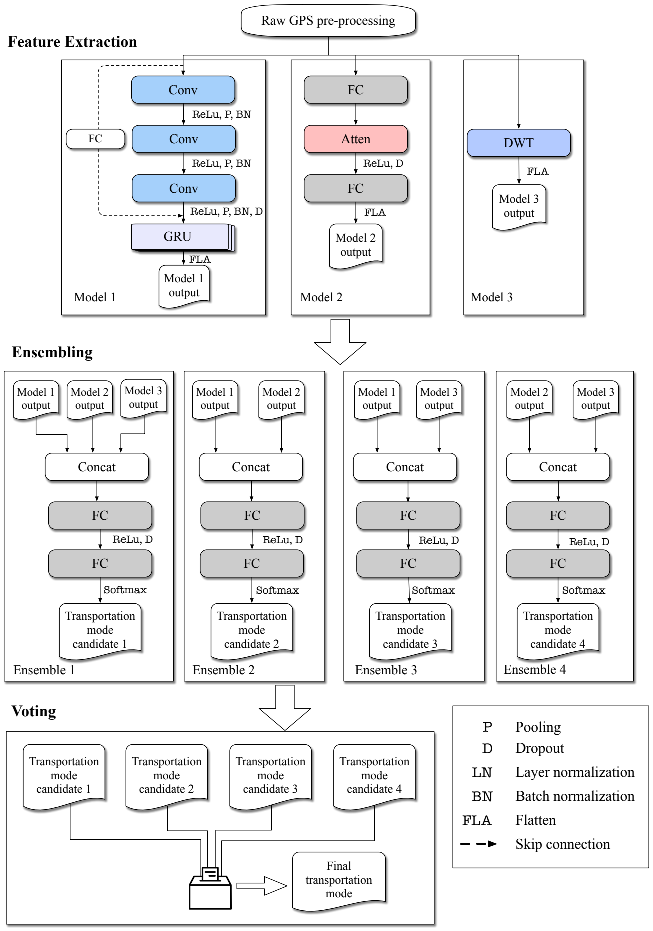


Fig. 1. The framework of TMI model.

by fusing multiple sub-models, while the training of an ensemble model might be relatively costly since more than one network are required to be trained on the same dataset [45]. The FL-based crowdsourcing can greatly improve the training efficiency on an ensemble model by distributed learning, which makes the best use of the advantages and circumvents the disadvantages of ensemble learning. Thus, we consider an ensemble model as the configured model for MTSSFL in the investigated crowdsourcing TMI task. The framework of the devised TMI model is shown in Fig. 1. It incorporates three procedures, namely, *feature extraction*, *ensembling*, and *voting*. Particularly, we employ three sub-models for feature extraction, i.e., Models 1 to 3. We elaborate on the three sub-models and the ensemble and voting procedures of our model in the sequel.

*1) **Model 1**:* This sub-model first employs three convolution layers with skip connections to capture the spatial features within the input motion features $\mathbf{X}$. The layers' output channels are set to 32, 64, and 128, respectively. The kernel sizes and strides for all three convolution layers are set to 3 and 1, respectively. Each convolution is followed by a max-pooling operation with size 2.

Next, we stack eight layers of Gated Recurrent Units (GRUs) with hidden size 16 to exploit the temporal dependencies within the input data. As a simplified variant of the LSTM module, the GRU adopts a stack of gating units and cell states to process and control the input information [8], [46]. There are two types of gating units, i.e., the update gate z and the reset gate r. The operations performed in each layer can be written as:

$$r_t = \sigma\left(W^{(r)}\mathbf{X}_t + U^{(r)}h_{t-1}\right), \tag{5}$$

$$z_t = \sigma\left(W^{(z)}\mathbf{X}_t + U^{(z)}h_{t-1}\right), \tag{6}$$

$$h'_t = \tanh\left(W\mathbf{X}_t + r_t \odot Uh_{t-1}\right), \tag{7}$$

$$h_t = z^t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t, \tag{8}$$

where $\mathbf{X}_t$ is the input of each GRU layer, $t \in T$, and weight matrices $W^{(z)}$, $W^{(r)}$, $U^{(z)}$, $U^{(r)}$ connect $\mathbf{X}_t$ and h_{t-1} to the two gates. Finally, h'_t is the intermediate activated output, while h_t denotes the final output.

*2) **Model 2**:* This sub-model incorporates an attention module between two linear layers, In this work, following the one proposed in [47], the attention module incorporates a stack of multi-head attentional layers, each of which computes the attention using scaled dot-product attention mechanism. The orchestra of multi-head enables the model to collectively involve knowledge learned from multiple representation subspaces. The processes can be formulated as

$$\mathbf{X}^A = \text{concat}(\text{hd}_1, \text{hd}_2, \ldots, \text{hd}_{n^*})W^O, \tag{9}$$

where

$$hd_i = \text{softmax}(\frac{QW_i^Q(KW_i^K)^T}{\sqrt{d_k}})VW_i^V, \tag{10}$$

where Q, K, V are the query, key, and value, which serve as the input of self-attention mechanism in [47], and the mechanism requires that they are all $\mathbf{X}$; $d_k = \frac{hidd_a}{n^*}$ denotes the dimension of keys where $hidd_a$ denotes the hidden size of the attention module, which is set to 128; softmax denotes the softmax activation function; hd_i denotes each attentional head and n^* is the number of heads; $\text{concat}(\cdot)$ represents the concatenation operation; W^O, W^Q, W^K, and W^V are the corresponding weight matrices. Considering the tradeoff between model performance and training overload, the number of heads and hidden size is set to 8 and 30, respectively.

*3) **Model 3**:* To better analyze the data characteristics, a wavelet representation-based feature extracting approach named Discrete Wavelet Transform (DWT) is adopted in Model 3 to further exploit the hidden time-domain feature from the feature vectors. Specifically, DWT employs discrete wavelets $\psi_{a,b}(t)$ to convolve the input, which can be defined as

$$\psi_{a,b}(t) = \frac{1}{2^a}\psi(\frac{t}{2^a} - b), \;\; a, b \in \mathbb{Z}, \tag{11}$$

where $\psi(t)$ is a pre-defined mother wavelets, a denotes the oscillatory level and b denotes the shifted position of DWT. Given a time sequence signal $x(t)$, DWT transforms the input by $\psi_{a,b}(t)$ into the following signal, which can be formulated as

$$d_{a,b}(x(t), \psi(t)) = \int_{-\infty}^{+\infty} x(t)\psi^*_{a,b}(t)dt = \langle x(t), \psi_{a,b}(t)\rangle, \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3132056, IEEE Internet of Things Journal

where $\psi^*_{a,b}(t)$ is the complex conjugate of $\psi_{a,b}(t)$. Additionally, DWT can be interpreted regarding a multi-resolution decomposition of the input signal. Specifically, following the inference in [48], given a decompsition level M, a hierarchical framework can be built as

$$\begin{aligned} s(t) =& \sum_a^M \sum_b d_{a,b}(x(t), \psi(t)) 2^{-a/2} \psi(\frac{t}{2^a} - b) \\ &+ \sum_b A_{M,b} 2^{-M/2} \varphi(\frac{t}{2^M} - b) \\ \triangleq& \sum_a^M D_a(t) + A_M(t), \end{aligned} \tag{13}$$

where $A_{M,b} = \langle x(t), \varphi_{M,b}(t) \rangle$ denotes the approximation coefficient at level M, $\varphi(t)$ represents a companion scaling function. We can utilize (13) to decompose the signal $x(t)$ into a detailed signal $D_a(t)$ and an approximation signal $A_M(t)$.

In this work, since we emphasize more on the general trend of GPS trajectory characteristics, the detailed signals are omitted, and only the approximation signal $A_M(t)$ of the pre-processed quarternary feature (i.e., $x_i = (d_i, s_i, a_i, j_i)$) is reserved. The *daubechies* mother wavelets are used to decompose the feature, following [48]. Finally, we can obtain the tensor of quarternary feature $\mathbf{X} \in \mathbb{R}^{T \times n \times 4}$ and the tensor after DWT $\mathbf{X}^{dwt} \in \mathbb{R}^{T \times n \times 1}$, with the latter forming the output of Model 3.

4) ***Ensemble and Voting:*** Multi-view Ensemble Learning (MEL) is a type of semi-supervised learning aiming to train different learning models with different views of the original data [49]. In the feature extraction stage, we constructed three sub-models to learn different latent representations of the input. Following the concept of MEL, we create four ensembles by concatenating the outputs of the three sub-models as:

$$\mathbf{X}^{E1} = \text{concat}(\mathbf{X}^{M1}, \mathbf{X}^{M2}, \mathbf{X}^{M3}), \tag{14}$$

$$\mathbf{X}^{E2} = \text{concat}(\mathbf{X}^{M1}, \mathbf{X}^{M2}), \tag{15}$$

$$\mathbf{X}^{E3} = \text{concat}(\mathbf{X}^{M1}, \mathbf{X}^{M3}), \tag{16}$$

$$\mathbf{X}^{E4} = \text{concat}(\mathbf{X}^{M2}, \mathbf{X}^{M3}), \tag{17}$$

where $\mathbf{X}^{M1}$, $\mathbf{X}^{M2}$, and $\mathbf{X}^{M3}$ are the outputs of the three sub-models; $\mathbf{X}^{E1}$, $\mathbf{X}^{E2}$, $\mathbf{X}^{E3}$, and $\mathbf{X}^{E4}$ denote the four ensemble tensors for corresponding ensembles. After a series of linear transformation of the ensemble tensor, we use a softmax function to estimate the probability that a trajectory segment pertain to a certain transportation mode for each ensemble, and further obtain the transportation mode by:

$$\text{mode}_{\text{pred}} \leftarrow \arg\max \text{softmax}(\mathbf{X}^{o,i}), \tag{18}$$

where $\mathbf{X}^{o,i}$ denotes the linear transformed tensor of each ensemble.

Finally, we use a voting strategy to decide the final transportation mode classification from the predictions of the four ensembles. Specifically, we regard the inferred transportation mode of each ensemble as a *candidate*. A hard voting procedure [50] is adopted, in which we select the final class as the one with the largest sum of votes among the classes predicted by the four candidates. In the case of a tie, we consider the transportation mode classification provided by Ensemble 1 as the final classification, since Ensemble 1 is trained on more inputs and is thus expected to be more reliable than the others.

As shown in Fig. 1, we adopt batch normalization followed by dropout with rate 0.5 to ensure training stability. We use the Rectified Linear Unit (ReLU) function to activate all hidden layers. Please note that all model hyperparameters are selected following standard deep learning practice, i.e., via grid search based on model performance on the validation set.

5) ***Sub-model Design Principles:*** We construct the above three sub-models based on the following intuition. Model 1 incorporates convolution layers and recurrent GRU modules to extract both spatial and temporal correlations of the transformed trajectory data. The combination of CNN and RNN has been demonstrated to be effective in exploiting the spatial-temporal correlations in traffic data [22], [51]. Models 2 and 3 are instead tailored to time series feature learning, each producing different latent representations due to the nature of the attention mechanism and DWT. Particularly, for Model 2, the adopted attention mechanism enables to “attend” to the parts that are relevant to the current part of the data since arbitrary parts of time-series can be more important at different time steps, which overcomes the limitation of GRU adopted in Model 1 that encode everything into one or more hidden layers of fixed size. For Model 3, the adopted DWT, as a signal decomposition technique, can extract frequency-domain data features of the input motion feature vectors. Compared with other frequency-domain processing techniques, DWT is also capable of developing a relatively stable spectrum when handling temporally non-stationary signals [52], which is applicable to the input motion feature vectors of our model. Given that each of the three sub-models has strengths that the others do not, and although it is uncertain which one is the most discriminative for transportation mode identification, ensembling them aggregates their predictions and ensures that their collective knowledge is utilized towards the final decision making.

C. Mean Teacher Semi-Supervised Federated Learning (MTSSFL)

In this paper, we propose a semi-supervised FL framework, MTSSFL, to solve the massive unlabeled data problems in data collaboration of federated learning for crowdsourced TMI. In this subsection, we introduce the proposed framework by firstly presenting the architecture, participants, and communication protocol. Then, we elaborate on the proposed model optimization and aggregating algorithms of MTSSFL. Lastly, we describe the designs for the privacy protection and trustworthiness of MTSSFL.

1) ***Architecture, Participants, and Communication Protocol:*** As illustrated in Fig. 2, MTSSFL is a framework designed for privacy-protected FL data collaboration. There are three main entities in MTSSFL, namely, publisher, workers, and third-party evaluator. The publisher intends to train a powerful model; however, it only possesses a small amount of labeled GPS trajectory data in its own central server. Therefore,

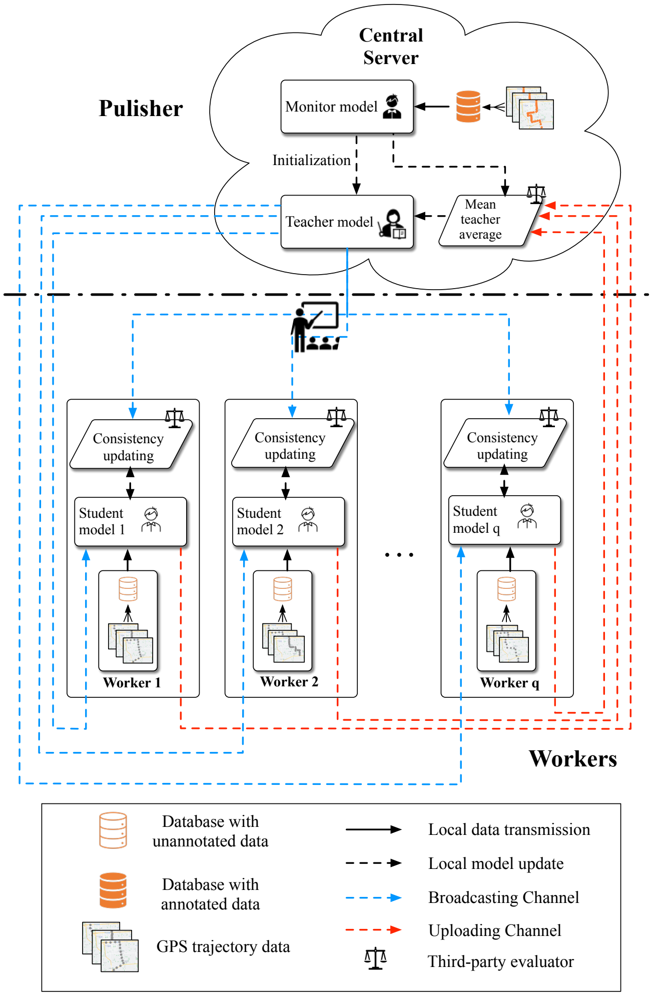


Fig. 2. Framework overview of the proposed MTSSFL.

Algorithm 1: MTSSFL Communication Protocol

```
Initialization:
1 Central server pre-trains the monitor model with
    labeled data, and update the pre-trained model
    parameter to the teacher model.
Iteration:
2 The central server distributes the copies of the teacher
    model to all workers, and each worker trains its
    copies with unlabeled data using the **pseudo-labeling**,
    and ***consistency-updating*** approaches.
3 Each worker uploads the updated model parameters to
    the central server. The central server aggregates the
    model parameters from the teacher model, student
    models, and the monitor model using the
    **mean-teacher-averaging** approach and updates the
    teacher model's parameters with the aggregation
    result.
```

the publisher publishes a crowdsourcing TMI task. Workers receive the task and process the local model training to serve the data collaboration with the publisher and other workers; however, each of them only possesses a set number of GPS trajectory data which is unlabeled. The third-party evaluator is independent of other participants, which is assumed to be trustworthy. The security of training and data is critical in FL-based crowdsourcing task [53], [54]; therefore, the independently trained TMI model from different participants and the data for training will be firstly assessed by the third-party evaluator before any aggregation or collaboration[2]. In this work, since all the data and participants are assumed to be benign, we consider all the evaluations are passed in our simulations. To successfully achieve the training goal under such biased data distribution, MTSSFL introduces a novel semi-supervised learning scheme. Specifically, MTSSFL regards the local models owned by workers as **student models**, and the global model on the central server as **teacher model**. We additionally introduce the **monitor model** on the central server to train with the labeled data. Terming it "monitor" is because trained with the labeled data can guarantee itself obtain the best training effect compared to the student models of workers, like a student who has the best learning ability in a class. The teacher model is not trained directly but updated by the aggregation of the monitor model and student models. Therefore, while the teacher and monitor models are both on the central server, they are of different functionality and cannot be conflated. Furthermore, since the communication and computation overhead by aggregation is minuscule compared with the training and data transmission process [55], a benefit of this design is that the development of the teacher model will not introduce extra computation burden.

The communication protocol between the entities is demonstrated in Algorithm 1. In the following subsections, we will give details to the involved approaches, i.e., *consistency-updating* and *mean-teacher-averaging*.

*2) **Consistency-updating**:* Before presenting the proposed *consistency-updating* approach, we first briefly introduce the pseudo-labeling approach. Pseudo-labeling is a popular approach in the semi-supervised learning community, which is a process of using the model trained on the labeled data to make predictions on the unlabeled data, filtering the samples based on the classification results, and re-inputting them into the model for training [35]. Particularly, the pseudo-labeling approach considers the training balance between labeled data and unlabeled data, which defines a loss function as

$$J = \frac{1}{B}\sum_{m=1}^{B}\sum_{i=1}^{c} J(y_i^m, \hat{y}_i^m) + \alpha(t)\frac{1}{B'}\sum_{m=1}^{B'}\sum_{i=1}^{c} J(y_i'^m, \hat{y}_i'^m), \tag{19}$$

where B and B' denote the size of mini-batch in labeled and unlabeled data, respectively; $\hat{y}_i^m$ and $\hat{y}_i'^m$ are the output of m samples in labeled and unlabeled data, respectively; y_i^m and $y_i'^m$ are the labels of m samples in labeled and unlabeled data, respectively; $\alpha(t)$ is a balancing weight.

In the condition of data collaboration defined in this paper, it is hard to directly adopt the pseudo-labeling approach due

[2] The technical detail of the operation of the third-party evaluator is out of the scope of this paper since it is not an influential factor of the investigated semi-supervised federated learning performance. Related investigation will be included in future work.

to that local student models cannot access the labeled data for pre-training. In this context, we proposed *consistency-updating* approach. Specifically, we follow the pseudo-label generation as proposed in [35] to develop the pseudo labels for unlabeled data[3]. Moreover, to fine-tune the student models in the data collaboration condition, we incorporate the loss developed by the teacher model when updating the local student models. A consistency cost J_{con} is introduced to measure the distance between the prediction of student model and teacher model on unlabeled data using L2 loss, which can be formulated as:

$$J_{con} = \frac{1}{c}\sum_{i=1}^{c}\left(\dot{f}(x_i^{'m}) - f(x_i^{'m})\right)^2, \tag{20}$$

where $\dot{f}$ and f denote the teacher and student TMI model, respectively; $x_i^{'m}$ is the unlabeled data. Further, we can define a final loss function J_s for the student model's update by combining Eq. (20) and the second half of the Eq. (19), which can be formulated as

$$J_s = \frac{1}{B'}\sum_{m=1}^{B'} J_{con} + \frac{1}{B'}\sum_{m=1}^{B'}\sum_{i=1}^{c} J(y_i^{'m}, f(x_i^{'m})), \tag{21}$$

*3) **Mean-teacher-averaging**:* In this work, we proposed a mean-teacher-averaging as a secure parameter aggregation mechanism for the TMI models based on the approach proposed in [37]. In [37], the authors proposed an EMA-based approach to update the model parameter of the teacher model at communication round t, which can be defined as

$$\dot{\phi}_t = \delta\dot{\phi}_{t-1} + (1-\delta)\phi_t, \tag{22}$$

where $\dot{\phi}$ and ϕ are the parameters of teacher model and student model, respectively; δ is a smoothing coefficient. The adoption of EMA make $\dot{\phi}_t$ react more significantly to the most recent learned parameters, which can guarantee a satiating training effect [37].

We extend this method into a model parameter aggregation method applicable to FL. Specifically, we first use the naive FedAvg [56] to aggregate model updates of student and monitor models, which can be formulated as

$$\phi_{\Sigma,t} = \frac{1}{1+q}\left(\sum_{i=1}^{q}\phi_{i,t} + \ddot{\phi}_t\right), \tag{23}$$

where q denotes the number of student models (i.e., workers); $\ddot{\phi}$ is the model parameter of monitor model; ϕ_Σ represents the aggregated parameters from student and monitor models by FedAvg. Subsequently, we incorprate the ϕ_Σ by Eq. (23) into Eq. (22), and finally obtain the aggregation mechanism for MTSSFL as

$$\dot{\phi}_t = \delta\dot{\phi}_{t-1} + (1-\delta)\phi_{\Sigma,t}. \tag{24}$$

Note that appropriate selection of the smoothing coefficient is vital when training the teacher model. We empirically found that a default value of $\delta = 0.1$ works well in practice; Section V-C3 includes a hyperparameter sensitivity test for δ and provides guidelines into setting its value based on application requirements.

To summarize, the entire working process of MTSSFL is shown as Algorithm 2.

V. Case Studies

To fully assess the performance of the proposed MTSSFL framework in identifying the transportation modes with massive unanotated data, a series of comprehensive case studies are carried out with a real-world data set. Firstly, we compare the proposed scheme with previous transportation mode models in the literature. Subsequently, the performance of the proposed scheme on non-IID data is investigated. Then, we investigate the performance sensitivity of the proposed scheme to the hyperparameters of MTSSFL. Lastly, a study on the performance of the proposed scheme with fewer local iterations is conducted.

A. Experimental Setup

*1) **Dataset Description**:* MTSSFL is evaluated on Geolife [12], [57], an open dataset of real-world GPS trajectories by Microsoft Research Asia. It contains a total of 17,621 trajectories collected by 182 users over 2,090 days, with a total traveled distance of 1,292,951 kilometers. Among these users, only 69 have labeled parts of their trajectories by transportation mode. The labeled trajectories are considered as ground truth and pre-processed as per Section IV-A. Even though a total of eleven transportation modes are labeled, not all of them are sufficiently represented in the dataset. As such, we follow the dataset authors' recommendations in only considering the five most prominent transportation modes (see Section III-A). We also perform data augmentation by flipping the input motion features along the temporal dimension as in [58]. Finally, a total of $T^* = 16,370$ single-transportation-mode GPS trajectory segments are obtained based on the available ground-truth labels. Note that, although evaluation on at least one more dataset would be ideal, we are not aware of any other dataset of similar or larger size that contains densely sampled GPS trajectories labeled by transportation mode.

To train the TMI models involved in the proposed scheme in a semi-supervised manner, we first partition the original dataset into two datasets using a 17:3 ratio. The second dataset is used as the testing set and serves the purpose of evaluating the identification accuracy of the monitor model when assessing MTSSFL. We further partition the first of the two datasets into a percentage of unlabeled data[5] and labeled data, with the proportion of unlabeled and labeled data being denoted by γ and $1-\gamma$, respectively; in other words, we use hyperparameter γ to control the percentage of unlabeled data in the training set. In MTSSFL, the labeled data are assigned to the monitor model, while the unlabeled data are uniformly distributed among the student models. Please note that $\gamma = 0$ effectively corresponds to using all labels

[3] Interested readers can refer the detailed process of pseudo-labeling in the reference.

[4] The numbers of epochs for the monitor model and student models are set the same (i.e., E_l) to reduce macroscopic communication latency between publisher and workers.

[5] The transportation mode labels of this data subset are discarded.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3132056, IEEE Internet of Things Journal

Algorithm 2: MTSSFL Training

```
Input: workers $\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$; number of
       communication rounds (global epochs) $E$;
       cloud mini-batch size (for monitor model) $B$;
       local mini-batch size (for student models) $B'$;
       number of local epochs $E_l$[4]; learning rate $\eta$;
       gradient optimizer $\mathcal{L}(\cdot)$ for TMI model;
       gradient optimizer $\mathcal{L}_{con}(\cdot, \cdot)$ for TMI model
       used for consistency-updating as per Eq. (21);
       smoothing coefficient $\delta$; volunteer ratio $\mu$.
Output: Teacher model parameters $\dot{\phi}$.

CentralServer:
    Initialization:
        initialize and pre-train monitor model
        parameters $\ddot{\phi}$
        update teacher model parameters $\dot{\phi}$ with $\ddot{\phi}$
    Iteration:
        for round $t = 1, 2, \ldots, t \in E$ do
            broadcast $\dot{\phi}_t$ to workers
            for epoch $t = 1, 2, \ldots, t \in E_l$ do
                $\ddot{\phi}_t \leftarrow$ MU ($\ddot{\phi}_{t-1}$, $B$)
            wait until all $\phi_{i,t}, i \in q$ are received
            randomly select a ratio ($\mu$) of volunteers
            to participate in this round of aggregation
            $\dot{\phi}_{t+1} \leftarrow$ MTA ($\phi_{i,t}$, $\ddot{\phi}_t$, $\dot{\phi}_t$, $\delta$, $q$)

Worker:
    Initialization:
        initialize student model parameters $\phi_i$
    Iteration:
        for round $t = 1, 2, \ldots, t \in E$ do
            foreach worker $C \in \mathcal{C}$ in parallel do
                receive $\dot{\phi}_t$ from central server
                for epoch $t = 1, 2, \ldots, t \in E_l$ do
                    $\phi_t \leftarrow$ CU ($\phi_{t-1}$, $\dot{\phi}_t$, $B'$)
                upload $\phi_t$ to central server

Approach MU ($\phi_t$,$B$):
    // Model update
    foreach batch $b = 1, 2, \ldots, b \in B$ do
        $\phi_{t+1} \leftarrow \phi_t - \eta \cdot \mathcal{L}(\phi_t)$

Approach CU ($\phi_{t-1}$,$\dot{\phi}_t$, $B$):
    // Consistency update
    foreach batch $b = 1, 2, \ldots, b \in B$ do
        $\phi_t \leftarrow \phi_{t-1} - \eta \cdot \mathcal{L}_{con}(\phi_{t-1}, \dot{\phi}_t)$

Approach MTA ($\phi_{i,t}$,$\ddot{\phi}_t$, $\dot{\phi}_{t-1}$, $\delta$,$q$):
    // Mean-teacher average
    $\dot{\phi}_t = \delta\dot{\phi}_{t-1} + \frac{1-\delta}{1+q}(\sum_{i=1}^{q} \phi_{i,t} + \ddot{\phi}_t)$
```

in fully supervised learning. In this case, we only report the identification accuracy of the monitor model when evaluating MTSSFL, since no data are distributed to the student models.

TABLE I
Comparison of Transportation Mode Identification Methods

Method	Accuracy	Semi-supervised	Secure crowdsourcing
MLP	33.1%	–	–
SVM	47.0%	–	–
KNN	54.9%	–	–
CNN	83.6%	–	–
LSTM	81.7%	–	–
SPL	72.5%	✓	–
SGAN	83.1%	✓	–
SECA	73.2%	✓	–
STS	59.1%	✓	–
ELSTM	90.3%	✓	–
MTSSFL	89.2%	✓	✓

2) ***Experimental Settings:*** Our simulations were conducted on a server equipped with eight NVIDIA GeForce RTX 2080 GPUs and an Intel Xeon E5-2620 v4 CPU. All neural networks were developed using PyTorch v1.6.

3) ***MTSSFL Configuration:*** Unless otherwise stated, we set the number of workers $q = 20$, the number of communication rounds (i.e., global epochs) $E = 100$, and the number of local epochs $E_l = 5$. We also select the mini-batch size for the monitor and student models as $B = 256$ and $B' = 50$, respectively. All models in MTSSFL are trained using the Adam optimizer [59] with learning rate $\eta = 0.0005$. We also set the smoothing coefficient for mean-teacher-averaging $\delta = 0.2$, the volunteer ratio $\mu = 0.5$, and the proportion of unlabeled data $\gamma = 0.5$. Please note that all baselines' hyperparameters are selected according to their corresponding literature.

4) ***Baselines:*** We evaluate MTSSFL against a series of established TMI baselines. Specifically, we consider the following fully supervised machine and deep learning models: (1) Multi-Layer Perceptron (MLP) [19], (2) k-Nearest Neighbors (KNN) [19], (3) Support Vector Machine (SVM) [19], (4) CNN [13], and (5) LSTM [60]. We also include the following state-of-the-art semi-supervised TMI approaches: (6) SEmi-supervised Convolutional Autoencoder (SECA) [21], (7) Semi-Pseudo-Label (SPL) [21], (8) Semi-supervised Generative Adversarial Network (SGAN) [38], (9) Ensemble-based LSTM (ELSTM) [19], and (10) Semi-two-steps (STS) [21]. It is worth mentioning that, since models (1)–(5) are not designed for semi-supervised learning, we only use labeled data to train them.

B. Identification Accuracy

1) ***Results:*** The comparison of identification accuracy is shown in Table I. It is evident the proposed scheme outperformed the traditional machine and deep learning baselines while performing comparably to the state-of-the-art semi-supervised frameworks. The inferior performance of the conventional machine learning approaches, MLP, SVM, and KNN can be attributed to their shortage in handling complex nonlinearity of data features. Nevertheless, we can observe a satisfying result obtained by CNN and LSTM, which implies their capacity of capturing spatial correlation information

TABLE II
SENSITIVITY OF TRANSPORTATION MODE IDENTIFICATION ACCURACY TO PERCENTAGE OF UNLABELED DATA

Method	Accuracy (%)					
	$\gamma = 0.99$	$\gamma = 0.95$	$\gamma = 0.90$	$\gamma = 0.80$	$\gamma = 0.50$	$\gamma = 0$ (Supervised)
SPL	50.9	56.0	61.8	68.6	72.5	75.4
SGAN	68.4	77.7	80.5	82.1	83.1	83.8
SECA	52.0	56.1	62.9	69.3	73.2	76.8
STS	50.7	53	50.6	54.4	57.7	59.1
ELSTM	84.8	86.5	89.0	90.0	90.8	91.5
MTSSFL	82.4	83.1	85.7	87.3	89.2	91.4
MTSSFL-non-IID	82.3	82.5	85.7	86.9	89.0	91.3

and long-term temporal correlation information from the data respectively regardless of user data privacy. Among the semi-supervised *state-of-the-arts* approaches, the proposed scheme scores the second. Compared to STS, SGAN, and SECA, ELSTM and the proposed scheme achieve higher accuracy thanks to the learning capability of the ensemble design. The difference between the proposed scheme and the first-performing ELSTM method is due to the difference in network size that the number of our employed neurons is only approximately a third of the latter's. Meanwhile, considering the distributed configuration of the proposed scheme, the actual performance gap could be even smaller.

*2) **Accuracy for Different Percentages of Unlabeled Data**:* To evaluate MTSSFL against the selected baselines when different percentages of unlabeled data are available at training time, we vary $\gamma \in \{0.99, 0.95, 0.9, 0.8, 0.5, 0\}$. According to the results reported in Table II, we observe that MTSSFL achieved 82.4% accuracy when only 1% of training data are labeled, i.e., $\gamma = 0.99$. Accuracy only declined by 6.8% compared to when $\gamma = 0.5$. We note that, although ELSTM attained the highest classification accuracy for all percentages of unlabeled data, it was not designed with federated learning in mind and thus cannot provide secure crowdsourcing. Achieving the third best overall results, SGAN scored an accuracy of 83.1% for $\gamma = 0.50$; yet it only achieved 68.4% accuracy when $\gamma = 0.99$, demonstrating significant performance degradation when labeled data were scarce.

*3) **Accuracy on non-IID Data**:* As discussed in Section III-B2, non-IID data are commonly encountered in crowdsourcing applications. Therefore, we additionally study the proposed scheme's performance on non-IID data. We construct a non-IID dataset based on Eq. (2), where $R = 1$. Our simulation results are shown in Table II *MTSSFL-non-IID*. We find that, regardless of the percentage of unlabeled data γ, the proposed scheme performed similarly on both IID and non-IID data. This indicates that MTSSFL is indeed robust to non-IID data.

C. Performance with Different MTSSFL Hyperparameters

*1) **Number of Workers** q:* The default number of workers in our simulations is set to $q = 20$. However, in real-world crowdsourced TMI, the number of workers is expected to be very large. To examine the performance of the proposed scheme with a larger number of workers, we conduct a series of simulations with $q \in \{20, 30, 40, 50, 100\}$.[6]

From the results shown in Fig. 4(a), it is evident that the number of workers has a negative correlation with the accuracy of the proposed scheme on both IID and non-IID data. This is not surprising, as more workers mean more model parameters to be learned and more unlabeled data to be used in training; this makes performing the required aggregation algorithms more challenging for the centralized server. We can draw the same conclusion from the convergence curves shown in Fig. 3(a) and 3(d), where larger numbers of workers resulted in slower training convergence. Nonetheless, it is important to note that the accuracy degradation was not significant. In the case of 100 workers, only a $0.9 - 1\%$ accuracy reduction was observed.

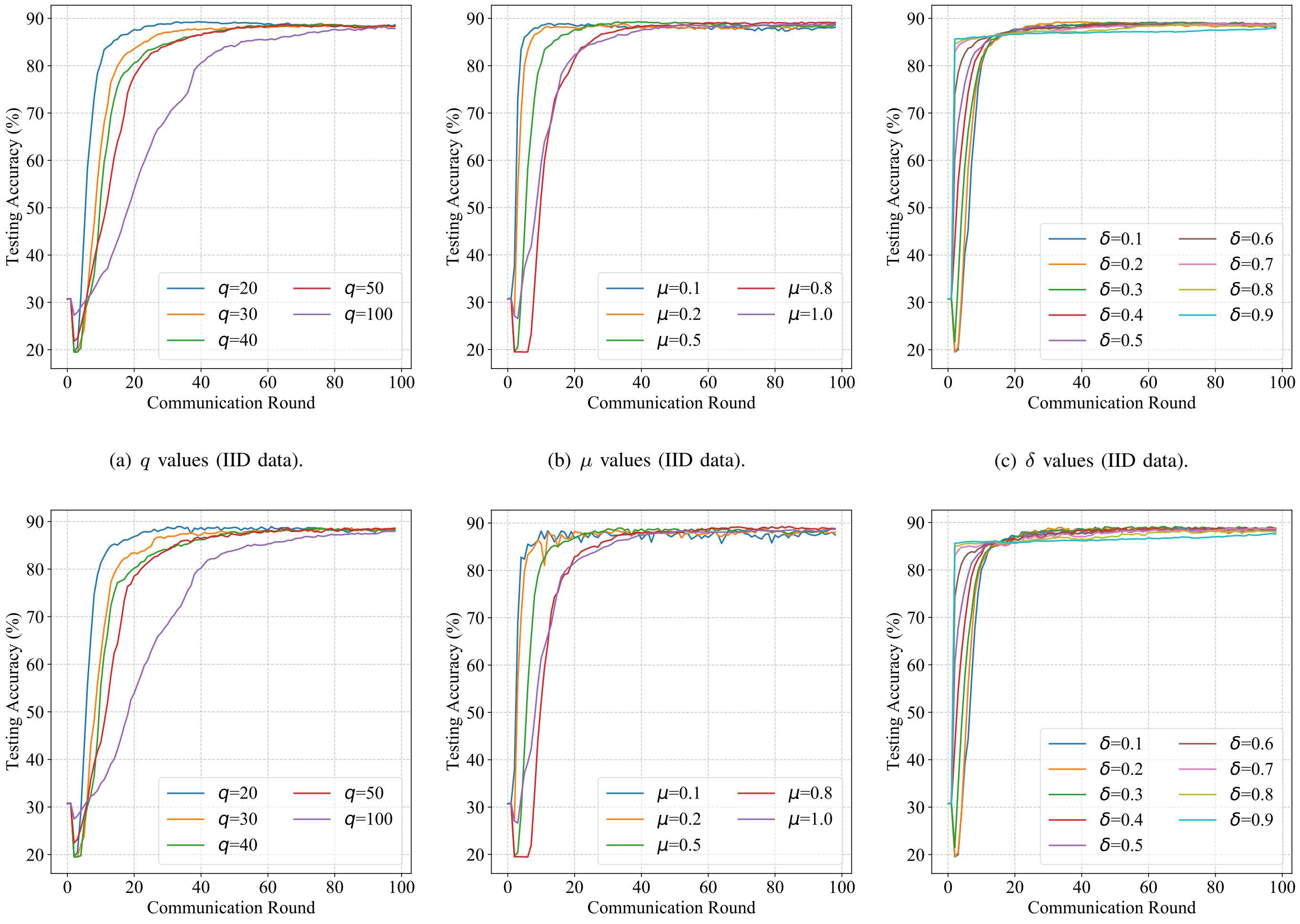
*2) **Fraction of Volunteers** μ:* In the above tests, the fraction of volunteers participating in a communication epoch was empirically set to $\mu = 0.5$. It is interesting to investigate the impact of different μ on the proposed scheme's performance. In this experiment, we evaluate classification accuracy by varying $\mu \in \{0.1, 0.2, 0.5, 0.8, 1\}$.

The simulation results are shown in Fig. 3(b), 3(e) and 4(b). It appears that higher volunteer fractions negatively affect the convergence speed of the global model, yet the impact on the final classification accuracy is negligible: the difference between the highest accuracy and the lowest accuracy is merely 0.3% (IID) and 0.8% (non-IID), respectively. This implies that a properly selected fraction of volunteers can somewhat improve model performance, while too many may slow down convergence.

*3) **Smoothing Coefficient** δ:* The proposed secure parameter aggregation mechanism of MTSSFL, termed emphmean-teacher-averaging, uses smoothing coefficient δ to control the exponential moving average when training the global model. Here, we examine the training impact of different values for δ by varying $\delta \in \{0.1, 0.2, \ldots, 0.9\}$.

According to the learning curves shown in Fig. 3(c) and 3(f), it appears that the higher the value for δ, the faster training converges. However, Fig. 4(c) shows that the final accuracy decreased as δ increased, with the best value for δ on IID and non-IID data being 0.2 and 0.3, respectively. The above trade-off has a significant implication for MTSSFL's real-world applicability: system operators will be able to either increase δ to accelerate convergence, or use relatively smaller

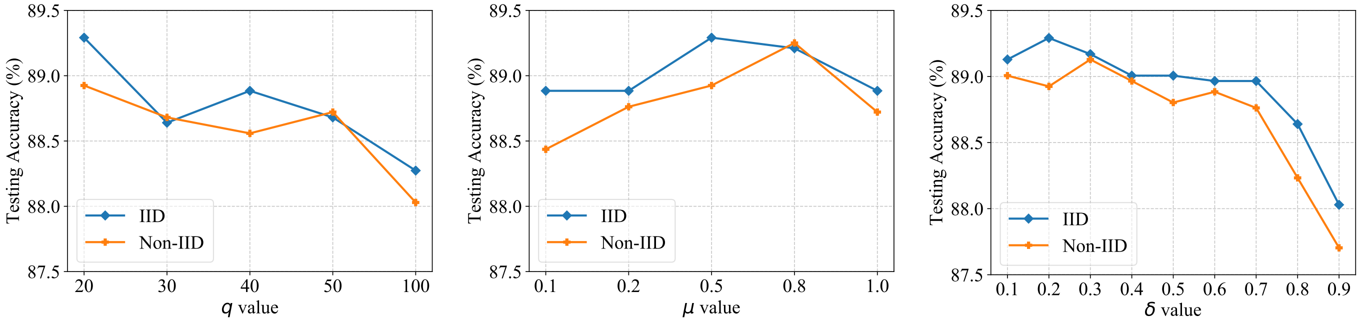
[6] Simulations are limited to $q \leq 100$ due to the dataset's moderate size.



(a) q values (IID data). (b) μ values (IID data). (c) δ values (IID data).

(d) q values (non-IID data). (e) μ values (non-IID data). (f) δ values (non-IID data).

Fig. 3. Learning curves of MTSSFL for different hyperparameter settings.



(a) Accuracy for different q values. (b) Accuracy for different μ values. (c) Accuracy for different δ values.

Fig. 4. Accuracy of MTSSFL for different hyperparameter settings.

values for δ when maximizing classification accuracy is the main target.

*4) **Local Iterations** E_l:* It is rather challenging to guarantee both high accuracy and low latency in the crowdsourced TMI. Latency may occur due to two factors: one is the latency of workers, which is excluded from the scope of this study based on our assumption in Section III-B1. The other is the workers' training time consumption. In this work, we empirically set the default number of local training epochs $E_l = 5$. To further investigate whether the proposed scheme can meet the requirements of crowdsourced TMI in fewer training iterations, we also evaluate its performance with fewer local training

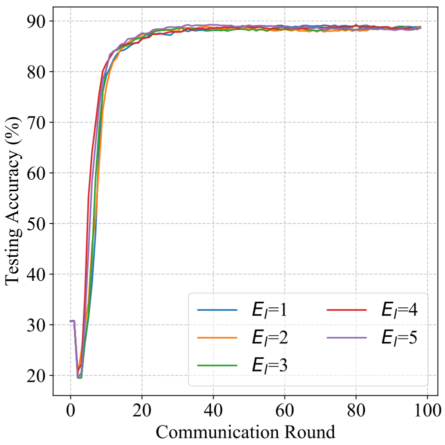


Fig. 5. Learning curves of MTSSFL for different numbers of local epochs.

TABLE III
ACCURACY AND TRAINING TIME OF MTSSFL FOR DIFFERENT NUMBERS OF LOCAL EPOCHS

E_l	Accuracy (%)	Training time (s)
1	89.1	750 ± 30
2	88.9	1480 ± 30
3	88.8	1920 ± 45
4	88.3	2620 ± 70
5	89.2	3110 ± 80

Note: Deviations are due to fluctuations in the performance of the computing device.

epochs, i.e., we test $E_l \in \{1, 2, 3, 4, 5\}$.

The corresponding simulation results are shown in Fig. 5 and Table III. It is evident that the performance of MTSSFL did not deteriorate when decreasing E_l. In fact, $E_l = 1$ resulted in nearly the same accuracy and convergence speed as $E_l = 5$. However, the former led to a 75.6% reduction in training time. This reduction can be attributed to the proposed consistency-updating scheme performing fewer iterations. Thus, for time-critical scenarios, training time can be reduced without significantly compromising TMI accuracy by employing fewer local training iterations.

In summary, the learning curves presented in Fig. 3 and 5 shows that MTSSFL can achieve the satisfying convergence performance under different conditions even though the convergence speed may differ, which demonstrates the convergence robustness of MTSSFL.

VI. SECURITY AND PRIVACY (S&P), AND GENERALIZATION DISCUSSION ON MTSSFL

A. *Security and Privacy*

The FL nature can guarantee MTSSFL with good TMI performance of the collaborative model without involving any data exchange among the workers. Furthermore, MTSSFL also allows workers only having unlabeled data to contribute to training the global model without accessing the central server's labeled data. While in-depth S&P issues such as defense against malicious attackers are out of the scope of this paper, MTSSFL incorporates the following advantages in terms of security and privacy:

- **Resiliance**. In MTSSFL, the amount of unlabeled data involved in training is equal for each worker regardless of how much data they actually have. For example, in real-world scenarios, different workers may generate different numbers of GPS trajectories due to factors such as frequency of travel or online connectivity. This leads to a mismatch in the amount of data they possess and can therefore be used for training. For workers who contribute significantly more data during training, if their data are adversarially mixed with malicious or low-quality data, the teacher model may easily be poisoned, and its TMI performance may suffer. To mitigate this issue, we set the constraint that the amount of data involved in training is equal for each worker. Furthermore, in Section III-B1, we assume that the security of uplink channels is higher than that of downlink broadcasting channels. In MTSSFL, the two channels (denoted in Fig. 2 by blue and red dotted lines, respectively) are set up to use different frequency bands or time slots, while the frequency bands also change dynamically to protect the model transmission from eavesdropping attacks.
- **Trustworthiness.** To preserve the contribution of each worker and enhance trustworthiness during crowdsourcing, an authorized third-party evaluator is involved in examining the model updating, performance evaluation, and aggregation processes of MTSSFL. The workers that are tested to be anomalous will be disqualified from the crowdsourcing. It is also worth mentioning that, as illustrated in Fig. 2, the third-party evaluator operates locally, thereby reducing additional communication overhead and avoiding possible malicious attacks during transmissions.

In this work, we focus on the semi-supervised federated learning performance of MTSSFL, and we will further investigate the above S&P factors in future work.

B. *Generalization*

In this work, MTSSFL is proposed to solve the lack of labeled data problems, which is a realistic problem in crowdsourcing TMI tasks. However, this problem is not peculiar to the crowdsourcing TMI tasks, which also exists in other tasks. Thus, the proposed framework can be used for a broader range of semi-supervised federated learning. For those applications which have, but not limited to, the following characteristics, MTSSFL are well-suited:

- **Expensive data labeling.** Unlike some simple classification tasks, the manual labeling of transportation mode to raw GPS data can be very expensive since it requires massive expert knowledge. However, ideally, supervised learning is more advised since massive data with exact labels usually develop more ideal training results [61]. MTSSFL can be considered in the cases where the manual labeling is indeed unattainable.
- **Stable and timely communication (data transmission).** As can be seen from the communication protocol of MTSSFL in Algorithm 2, the data transmission between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3132056, IEEE Internet of Things Journal

the central server and workers is relatively frequent due to the design of consistency update approach. For applications running in unstable or inferior network environments, the possible straggling effects may degenerate the framework's overall efficiency.

- **Semi-supervised classification task.** As a semi-supervised learning approach, MTSSFL is designed for TMI, which is a classification task. For these "semi-supervised classification" tasks, MTSSFL can be directly applied. However, for "semi-supervised regression" tasks where the output variable is real-valued (e.g., missing data imputation) [62], a specific variant of MTSSFL is required to handle them which is out of the scope of this paper.

VII. Summary and Future Work

In this paper, we proposed a novel semi-supervised FL approach for TMI to address a *de facto* problem that hinders the realization of crowdsourced TMI: the fact that only a small number of labeled data (owned by the central server) can be used in model training, while the distributed workers only possess their sensed (unlabeled) data. To this end, we first introduced a deep ensemble-learning-based TMI scheme to exploit the spatial-temporal relationships within GPS trajectory data. We then proposed a semi-supervised FL framework, MTSSFL, tailored to the aforementioned problem. To train the distributed models having only unlabeled data at the workers' end, we proposed an approach named consistency-updating to get the central model trained with labeled data involved in the worker models' training process, under a "teacher-monitor-student" triad. Furthermore, to improve classification performance without the need for further training, we proposed an EMA-based approach named *mean-teacher-averaging* for model aggregation. In addition, we introduced a series of privacy and security design of MTSSFL. Our extensive case studies on a real-world GPS trajectory dataset showed that the proposed scheme outperformed established TMI approaches while protecting data privacy, performing comparably to the state-of-the-art with only marginally lower classification accuracy. We also find the capacity of the proposed scheme for handling non-IID data. By studying the influence of different hyperparameters on the model performance, we demonstrated the outstanding training efficiency of the proposed scheme, which can benefit the crowdsourced task.

In future work, we will further investigate in-depth privacy and security mechanisms for crowdsourced TMI. We will also extend MTSSFL to other domains to consolidate its universality and address other potential issues related to semi-supervised FL.

References

[1] H. Mäenpää, A. Lobov, and J. L. M. Lastra, "Travel mode estimation for multi-modal journey planner," *Transportation Research Part C: Emerging Technologies*, vol. 82, pp. 273–289, 2017.

[2] C. Zhang, S. Zhang, J. J. Yu, and S. Yu, "FASTGNN: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Transactions on Industrial Informatics*, 2021.

[3] B. Wang, L. Gao, and Z. Juan, "Travel mode detection using GPS data and socioeconomic attributes based on a random forest classifier," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1547–1558, 2017.

[4] M. Li, J. Wu, W. Wang, and J. Zhang, "Towards privacy-preserving task assignment for fully distributed spatial crowdsourcing," *IEEE Internet of Things Journal*, 2021.

[5] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[6] B. S. Ciftler, A. Albaseer, N. Lasla, and M. Abdallah, "Federated learning for RSS fingerprint-based localization: A privacy-preserving crowdsourcing method," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 2112–2117, IEEE, 2020.

[7] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," *arXiv preprint arXiv:1906.10893*, 2019.

[8] Y. Liu, S. Zhang, C. Zhang, and J. J. Yu, "FedGRU: Privacy-preserving traffic flow prediction via federated learning," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[9] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.

[10] G. Hua, L. Zhu, J. Wu, C. Shen, L. Zhou, and Q. Lin, "Blockchain-based federated learning for intelligent control in heavy haul railway," *IEEE Access*, vol. 8, pp. 176830–176839, 2020.

[11] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[12] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of the 17th international conference on World Wide Web*, pp. 247–256, 2008.

[13] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transportation research part C: Emerging technologies*, vol. 86, pp. 360–371, 2018.

[14] J. J. Yu, "Travel mode identification with GPS trajectories using wavelet transform and deep learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[15] C. Wang, H. Luo, F. Zhao, and Y. Qin, "Combining residual and LSTM recurrent networks for transportation mode detection using multimodal sensors integrated in smartphones," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[16] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10700–10714, 2019.

[17] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[18] S.-H. Fang, Y.-X. Fei, Z. Xu, and Y. Tsao, "Learning transportation modes from smartphone sensors based on deep neural network," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, 2017.

[19] J. J. Yu, "Semi-supervised deep ensemble learning for travel mode identification," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 120–135, 2020.

[20] F. Yang, Z. Yao, Y. Cheng, B. Ran, and D. Yang, "Multimode trip information detection using personal trajectory data," *Journal of Intelligent Transportation Systems*, vol. 20, no. 5, pp. 449–460, 2016.

[21] S. Dabiri, C.-T. Lu, K. Heaslip, and C. K. Reddy, "Semi-supervised deep learning approach for transportation mode identification using GPS trajectory data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 1010–1023, 2019.

[22] J. V. Jeyakumar, E. S. Lee, Z. Xia, S. S. Sandha, N. Tausik, and M. Srivastava, "Deep convolutional bidirectional LSTM based transportation mode recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1606–1615, 2018.

[23] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in GPS traces via uncertainty-aware path cloaking," in *Proceedings of the 14th ACM conference on Computer and communications security*, pp. 161–171, 2007.

[24] Z. Huo, X. Meng, H. Hu, and Y. Huang, "You can walk alone: Trajectory privacy-preserving through significant stays protection," in *International conference on database systems for advanced applications*, pp. 351–366, Springer, 2012.

[25] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, "Privacy-preserving transportation traffic measurement in intelligent cyber-physical road systems," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3749–3759, 2015.

[26] C. Tikkinen-Piri, A. Rohunen, and J. Markkula, "Eu general data protection regulation: Changes and implications for personal data collecting companies," *Computer Law & Security Review*, vol. 34, no. 1, pp. 134–153, 2018.

[27] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[28] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

[29] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

[30] Y. Liu, J. J. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.

[31] Y. Zhu, S. Zhang, Y. Liu, D. Niyato, and J. James, "Robust federated learning approach for travel mode identification from Non-IID GPS trajectories," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 585–592, IEEE, 2020.

[32] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[33] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 201–208, JMLR Workshop and Conference Proceedings, 2010.

[34] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature extraction*, pp. 1–25, Springer, 2006.

[35] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, 2013.

[36] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[37] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, pp. 1195–1204, 2017.

[38] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Semi-supervised GANs to infer travel modes in GPS trajectories," *arXiv preprint arXiv:1902.10768*, 2019.

[39] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "A survey towards federated semi-supervised learning," *arXiv preprint arXiv:2002.11545*, 2020.

[40] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[41] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[42] U. K. Dutta, M. Harandi, and C. C. Shekhar, "Semi-supervised metric learning: A deep resurrection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 7279–7287, 2021.

[43] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *arXiv preprint arXiv:2104.02151*, 2021.

[44] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975.

[45] M. Ganaie, M. Hu, *et al.*, "Ensemble deep learning: A review," *arXiv preprint arXiv:2104.02395*, 2021.

[46] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, Curran Associates, Inc., 2017.

[48] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[49] V. Kumar and S. Minz, "Multi-view ensemble learning: A supervised feature set partitioning for high dimensional data classification," in *Proceedings of the Third International Symposium on Women in Computing and Informatics*, pp. 31–37, 2015.

[50] R. Atallah and A. Al-Mousa, "Heart disease detection using machine learning majority voting ensemble method," in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1–6, IEEE, 2019.

[51] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.

[52] Z. Sun, L. Zhou, and W. Wang, "Learning time-frequency analysis in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3388–3396, 2018.

[53] Y. Guo, H. Xie, Y. Miao, C. Wang, and X. Jia, "Fedcrowd: A federated and privacy-preserving crowdsourcing platform on blockchain," *IEEE Transactions on Services Computing*, 2020.

[54] C. Zhang, Y. Guo, X. Jia, C. Wang, and H. Du, "Enabling proxy-free privacy-preserving and federated crowdsourcing by using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6624–6636, 2021.

[55] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2276–2288, 2020.

[56] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[57] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 312–321, 2008.

[58] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2019.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[60] G. Asci and M. A. Guvensan, "A novel input set for lstm-based transport mode detection," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 107–112, IEEE, 2019.

[61] R. Gilyazev and D. Y. Turdakov, "Active learning and crowdsourcing: A survey of optimization methods for data labeling," *Programming and Computer Software*, vol. 44, no. 6, pp. 476–491, 2018.

[62] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, "Semi-supervised regression: A recent review," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 2, pp. 1483–1500, 2018.