A Semisupervised End-to-End Framework for Transportation Mode Detection by Using GPS-Enabled Sensing Devices

Zhishuai Li, Gang Xiong, *Senior Member, IEEE*, Zebing Wei, *Graduate Student Member, IEEE*, Yisheng Lv, *Senior Member, IEEE*, Noreen Anwar, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—As an essential component of Internet of Things, GPS-enabled devices record tremendous digital traces, which provide a great convenience for understanding human mobility. How to discover transportation modes efficiently from such valuable sources has come into the spotlight. In this article, the transportation mode detection is treated as a dense classification task, and a similarity entropy-based encoder-decoder (SEED) model is proposed. We first design an encoder-decoder backbone for end-to-end mode detection. Then, a semi-supervised learning module based on similarity entropy is proposed to exploit numerous unlabeled data. Specifically, we stack several convolutional layers as an encoder to capture hierarchical features from fixed-length trajectories, and then adopt transposed convolutional layers as a decoder. For a semi-supervised module, inspired by entropy regularization, we use the *K*-Means algorithm to cluster prototype vectors from the encoder's predictions. We then fine-tune the encoder by sharpening the similarity distribution between unlabeled predictions and prototypes, aiming to make the former close to one prototype only while staying away from others. A majority-voting post-processing method is used to alleviate jitter impact when inferring. The Experimental results show that SEED significantly outperforms segmentation-then-inference methods. Furthermore, the similarity entropy-based module can improve the generalization performance of the model, and the metrics such as intersection over union can be increased by 5% over baselines. All of these verify the superiority of our method.**

***Index Terms*—GPS trajectory, human mobility, semi-supervised learning, transportation mode detection (TMD).**

Manuscript received March 23, 2021; revised August 23, 2021; accepted September 15, 2021. Date of publication September 24, 2021; date of current version May 9, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2104001; in part by the National Natural Science Foundation of China under Grant U1909204, Grant 61773381, Grant U1811463, Grant 61872365, and Grant 61773382; and in part by the Chinese Guangdong's S&T Project under Grant 2019B1515120030 and Grant 2020B0909050001. *(Zhishuai Li and Gang Xiong are co-first authors.) (Corresponding author: Yisheng Lv.)*

Zhishuai Li, Zebing Wei, and Yisheng Lv are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: yisheng.lv@ia.ac.cn).

Gang Xiong is with the Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Guangdong Engineering Research Center of 3-D Printing and Intelligent Manufacturing, The Cloud Computing Center, Chinese Academy of Sciences, Dongguan 523808, China.

Noreen Anwar and Fei-Yue Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Digital Object Identifier 10.1109/JIOT.2021.3115239

I. Introduction

WEARABLE devices have become a ubiquitous part of daily life for many. Thanks to the penetration of Internet of Things (IoT) devices, context-awareness computing has greatly attracted the attention of researchers [1]–[3]. As an application of IoT, transportation mode detection (TMD) for crowds is important [4]. By recognizing the travel mode from digital traces, TMD is instrumental in understanding travel behaviors and personal context awareness, serving as an utmost key component for urban planning and intelligent transportation systems [5]–[9]. Users can benefit from the rich knowledge about the recorded trajectories. They can not only know where others have been but also how people reach their locations, thus allowing service providers to make customized route recommendations.

Nowadays, despite multisensor fusion methods have already been studied [10]–[13], the majority of the infrastructures in transportation systems is equipped with GPS devices only, which means that the TMD research on single sensor remains general and necessary. If the inherent patterns of GPS traces are fully exploited, fewer or even no auxiliary devices need to be deployed, thus reducing management costs.

As earlier work, Zheng *et al.* [14] released the Geolife data set-based solely on GPS trajectory, and proposed a two-stage identification technique. It follows a segmentation-then-inference pipeline, in which the raw trajectory is first partitioned into several single-mode segments by heuristic segmentation algorithms, and then the mode of each segment is inferred via the trained machine learning models such as a decision tree (DT).

Nevertheless, it is extremely challenging to distinguish individual transportation modes, which can be evidenced by: 1) the data source only records partial motion features, resulting in unstable inference results. For instance, traffic patterns generated from driving are similar to those from walking when traffic congestion occurs; 2) the widely used two-stage approach requires manual intervention and heuristic design, which leads to inefficiency; 3) and unlabeled trajectory data are more than the labeled one, while the latter is rarely used to strengthen the model.

In response to these issues, in this article, we reframe TMD as a dense classification problem, in which the features of

fixed-length consecutive points in GPS records are regarded as input, and the mode regarding each GPS point is identified. We propose a similarity entropy-based encoder-decoder (SEED) model to recognize transportation mode in an end-to-end manner. Note that due to the noise caused by travel uncertainty (such as traffic jams), such point-level inference may cause a jitter problem, which means that the prediction for a particular point is different from its neighbors.

To achieve such a fine-grained identification, shallow and deep features from the consecutive points are both required. We thus use an encoder-decoder in SEED as a backbone to capture the hierarchical information among GPS points. This context-awareness structure cannot only extract the features of an isolated point but also handle the correlations of its neighbors. Moreover, a similarity entropy-based module is adopted to exploit the unlabeled trajectories to improve a model's generalization. Inspired by entropy regularization [15], [16], we use the K-Means algorithm to cluster several prototype vectors from the hidden predictions of an encoder, and then fine-tune the network by sharpening the similarity distribution between unlabeled predictions and prototypes, aiming to make the former close to one prototype only while staying away from others. For alleviating the jitter problem, we conduct a post-processing operation via a majority voting [17]. Concretely, we devise a fixed-length window that slides on the model's output. For the GPS point located in the center of the sliding window, its transportation mode is determined by the majority category in that window. In application, the trained semi-supervised model can be directly applied to infer the transportation modes for each trajectory point.

Our contributions are threefold.

1) We reframe the TMD task as an end-to-end dense classification instead of the commonly used two-stage detection. The experimental results demonstrate that our approach significantly outperforms typical segmentation-then-inference methods.
2) For the robust inference, we propose an encoder-decoder backbone to handle the hierarchical features among GPS points and then adopt a majority-voting method to post-process the output.
3) We propose a similarity entropy-based module to exploit massive unlabeled data for semi-supervised learning. The experimental results show that it is helpful to the generalization of our model, with up to 5% improvement over the model without this module.

The remainder of this article is organized as follows. Section II provides the literature review on related TMD and advanced semi-supervised methods. Section III states TMD formulation and data description. Section IV details SEED, including an encoder-decoder structure, proposed similarity entropy-based module, and training and inference procedures. Numerical experiments are performed in Section V. The concluding remarks and future directions are discussed in Section VI.

II. Related Work

Recently, with the proliferation of IoT devices, numerous digital traces can be easily obtained and used to provide emerging possibilities for mining human activity. This section first introduces the related literature and IoT-related data sets on a TMD problem, and then presents the advanced semi-supervised learning methods to solve it. We also survey recent work in Table I in terms of data sets, data sources, modes, used methods, adopted features, and reported accuracy.

A. Data Sources and Models of TMD

In earlier research, scholars had to collect the mobility data themselves for the study [29]–[31]. Muller and Ian recognized users' activities which include still, walking, and driving via global system for mobile communications (GSM) data in 2006. They calibrated the context of GSM signal strength by a hidden Markov model (HMM) and K-means algorithms, and achieved 80% classification accuracy [27]. Assemi *et al.* [28] used a nested logit model to infer the mode from smartphone travel surveys in New Zealand and Australia and reported high accuracy without any preprocessing operation. Wang *et al.* recruited 312 participants in Shanghai, and collected their GPS trajectories through a smartphone application. Then, they constructed a random forest model, which achieved the classification accuracy of 93% [32].

With the widespread concern of human mobility, many scholars, institutions, and companies provide open-source trajectories to support the replication of experiments. The data availability fosters a strand of literature. As pioneering work, Zheng *et al.* released Geolife, which includes the GPS tracks of 182 individuals from April 2007 to August 2012. They proposed a two-stage (i.e., segmentation-then-inference) framework that used a change point-based approach for segmentation and a DT algorithm for inference [14]. They also performed additional meaningful work based on the data set, such as exploring human mobility [33], social networking service [34], and interesting locations mining [35]. Subsequent work focused more on the inference stage, namely, predicting the transportation mode for the specified segments. To improve the accuracy of inference, Endo *et al.* proposed a novel feature extraction method. They mapped each segment into a 2-D image, and trained a convolutional neural network (CNN) to recognize the corresponding travel mode, achieving 67% accuracy [18]. Dabiri and Heaslip [19] adjusted all the segments to a fixed length, and utilized a 1-D CNN to extract the features from the segments. Then, an ensemble learning method is adopted to improve classification accuracy up to 84.8% from 79.8%. Jiang *et al.* [20] proposed a multiscale model that incorporated local and global features, then fed the concatenated features into a random forest for prediction.

Later, the HTC data set was published in 2012. It has the data of 224 volunteers and includes ten transportation modes, such as still, walk, run, bike, and vehicles. The trajectories are collected every 8.5 s by three motion sensors including accelerometers, magnetometers, and gyroscopes. Yu *et al.* [23] employed a support vector machine (SVM) to classify the modes, and obtained about 91% classification accuracy. With this data set, Vu *et al.* [24] proposed a novel control gated-based recurrent neural network (RNN) that used accelerometers only, obtaining slight improvement over other RNN variants. Asci and Guvensan [25] extended a series

TABLE I
SUMMARY OF RELATED WORK

Datasets	Data sources	Transportation modes	Methods	Adopted features	Reported accuarcy
GeoLife	GPS	Walk, Bus, Bike, Car/Taxi, Subway/Train	Decision tree [14]	Distance, speed, acceleration, etc of segments	72%
			Image-CNN [18]	Trajectories drawn as images	67%
			Ensemble-CNN [19]	Distance, speed, acceleration, and orientation of GPS points	85%
			Random forest [20]	Local and global attributes (speed, heading, etc) of trajectory	84%
			SECA [21]	Distance, speed, scceleration, and orientation of GPS points	77%
			Proxy-labels [22]	Discrete Fourier/wavelet transform of features for segments	92%
HTC	5 mobile sensors, e.g. GPS, WiFi	Still, Walk, Run, Bike, Motorcycle, Car, Bus, Metro, Train, High-speed rail	SVM [23]	Sensors' signals from time or spectrum domain	91%
			RNN [24]	The sensors' signals in sampled sliding windows	93%
			LSTM [25]	Features from time domain or spectrum domain after FFT	97%
US-TM	9 mobile sensors	Walk, Car, Still, Train, Bus	Random forest [26]	The sensors' signals in sampled sliding windows	93%
SHL	15 mobile sensors	Still, Walking, Run, Bike, Bus, Car, Train/Subway	CNN [12]	Hand-craft spectrum features in sensors sequence	94%
Others	GSM	Walking, Driving, Still	HMM [27]	Patterns of signal strength fuctuations and cells	80%
	GPS	Walk/Run, Bicycle, Car, Bus	Random forest [28]	Speed, Acceleration, orientation, distance, etc	93%

of hand-crafted spectrum-domain features by the fast Fourier transform (FFT) as the input for long short-term memory (LSTM), thus improving the accuracy by 2%.

The U.S.-Transportation data set[1] has nearly 32 h of total records for 13 participants during their daily activities. The data set is collected by six sensors in phones with a frequency of 20 Hz, and distinguishes five transportation modes: 1) walking; 2) car; 3) still; 4) train; and 5) bus. With different classification algorithms (DT, random forest, SVM, and neural networks), the maximum accuracy reported in [26] is 93%.

Sussex–Huawei locomotion (SHL) data set is a generic data set that is suitable for a wide range of research in TMD, mobility pattern mining, localization, tracking, etc. 2812-h labeled data for three participants (four smartphones for each one) is recorded by multiple sensors simultaneously with high-sampling rate [36]. To recognize eight transportation modes (still, walk, run, bike, bus, car, train, and subway), Wang *et al.* [37] designed several hand-craft features and used classifiers, such as Naive Bayesian, DT, random forest, K-nearest neighbor, and deep learning-based algorithms. Finally, a CNN model using spectrum features and post-processing operation achieved the best performance with an accuracy of 93.3%.

To sum up, IoT-related devices have evolved by leaps and bounds, and machine learning approaches are becoming more and more convenient to mine the underlying patterns from trajectories. For the mode detection of GPS trajectory, the two-stage framework is commonly used, in which some heuristic rules should be defined to distinguish segments. Therefore, it is necessary to develop a point-wise end-to-end classification framework for TMD.

[1]http://cs.unibo.it/projects/us-tm2017

B. Semi-Supervised Methods

The essence of semi-supervised learning is to enhance a model's generalization [38], [39]. In practice, it is less expensive and easier to obtain the unlabeled data, while avoiding the challenges of annotating a large amount of data. In our task, there is both label and unlabeled data. Neither supervised nor unsupervised learning algorithms can make full use of them. So an opportune semi-supervised methods should be developed. The taxonomy of semi-supervised approaches can be categorized as *unsupervised learning with labeled data* and *supervised learning with unlabeled data*. As an instance of the former, Zhu *et al.* [40] constructed a graph according to the correlation among samples, and then used the Gaussian random field to handle unlabeled data, which is a typical clustering-then-labeling method. For the latter, diverse approaches are proposed under different hypotheses, such as pretraining and surrogate-task learning. To make the model more robust, researchers have implemented consistency regularization methods, including teacher-student architecture [41], mixup [42], and mixmatch [43]. Virtual

adversarial training [44] added noise where a model's output varied dramatically, to ensure that the prediction remained consistent. The low-density separation between classes assumes that the decision boundary is located in a low-density region. Self-training for unlabeled data [45] was proposed by using the class with the maximum predicted probability as pseudo labels. In co-training, two different classifiers were trained separately and provided pseudo labels for each other, which can be viewed as label propagation [46].

From the perspective of TMD, Yazdizadeh *et al.* [47] proposed a semi-supervised GAN model (DCGAN) which output the classes of transportation modes rather than "real" or "fake." Under the framework of self-training, James developed proxy labels for unlabeled data by a trained model, and both true and proxy labels were used to train a stronger model [22]. Dabiri *et al.* [21] proposed a semi-supervised framework semi-supervised convolutional autoencoder (SECA) based on a surrogate reconstruction task to enhance a model. Their surrogate-task model achieved 5% improvement over the pseudo-label learning approaches.

The minimizing prediction entropy has been exhibited to act as a strong regularizer in semi-supervised learning [15]. Aiming to penalize low-confidence predictions, entropy regularization can be used to effectively handle unlabeled data when training samples are limited [48]. Google brain team systematically evaluated a confidence penalty method based on entropy maximization on six benchmarks, such as machine translation and speech recognition, and found that the penalty could further enhance the advanced models [49]. Dean [50] explored the effectiveness of regularizing the entropy of prediction with an uncertain number of classes, and verified that the entropy regularization was a promising method to de-bias the weakly supervised learning system. The entropy regularization is also researched in computer vision, especially for domain-awareness image segmentation [51]–[53]. In this article, we harness the power of a generalized entropy module to implement a semi-supervised learning method for more accurate TMD from GPS trajectory.

III. Preliminaries

A. Problem Statements

In this section, we introduce notations in TMD, and more details can be found in [19].

GPS Point and Trajectory: GPS devices log subscribers' locations periodically. Each recorded point p is a triplet consisting of geographic location and timestamp, namely, $p = [\text{latitude}, \text{longitude}, \text{timestamp}]$. The trajectory T for each subscriber is a series of time-stamped GPS points $T = [p_1, p_2, \ldots, p_N]$, where N represents the length of the sequence.

Change Point and Segmentation: The change point is defined as the location where subscribers switch their transportation mode. There may be zero or several change points in a particular trajectory. The segmentation operation means partitioning a trajectory into multiple segments according to change points. Consequently, the transportation mode within a segment of GPS points is identical.

TMD Problem: The objective of the study is to detect transportation modes for each GPS trajectory. Typical two-stage methods first implement trip segmentation from raw GPS records and then classify modes for each segment [14], [21]. Such segmentation-then-classification pipeline requires experiential design for detecting the change point, thereby leading to inefficiency. By contrast, we treat the problem as an end-to-end task, i.e., detecting transportation mode for each point from raw GPS records directly, which is a dense classification problem. The features of GPS points are taken as input, and the corresponding transportation modes are output.

Since the neural networks can only handle a fixed-length sequence, the trajectories are divided into many fragments with length L for detection. Different from the term "segment," the modes of the points within each fragment can be different. A fixed-length fragment $[p_1, p_2, \ldots, p_L]$ with size $(L \times K)$ is set as input $\mathbf{x}$, where K is the dimension of motion features of GPS points. The prediction y is of size $(L \times C)$, and encoded by one-hot labeling. C is the number of classes. In this task, five transportation modes are concerned as the labels, including walk, bike, bus, drive, and railway, namely, $C = 5$.

B. Motion Characteristicsof GPS Points

Based on the geographic location and timestamp recorded by GPS devices, researchers have performed various hand-crafted features to represent a GPS trajectory. Zheng *et al.* [14] took length, mean velocity, covariance of velocity, top three velocities, and top three accelerations from each segment to classify a mode. Further, bearing rate, velocity change rate, and stop rate were adopted as enhanced features to improve the inference accuracy in their subsequent work [33]. However, these segment-level features are tailored for the mode detection of segments and are not applicable for the point-level mode classification in this article.

Recently, the relative distance R, speed S, acceleration A, and bearing rate B of consecutive GPS points have been evaluated as effective features for mode detection [19], [21]. Therefore, we adopt the above hand-crafted features for the point-wise classification. The first three terms can be calculated by

$$R_i = \mathcal{G}\big([\text{lat}_i, \text{lon}_i], \big[\text{lat}_{i+1}, \text{lon}_{i+1}\big]\big) \tag{1}$$

$$S_i = \frac{R_i}{t_i - t_{i+1}} \tag{2}$$

$$A_i = \frac{S_i}{t_i - t_{i+1}} \tag{3}$$

where R_i, lat_i, lon_i, t_i, S_i, and A_i represent the relative distance, latitude, longitude, timestamp, speed, and acceleration of point p_i, respectively. $\mathcal{G}([\text{lat}_i, \text{lon}_i], [\text{lat}_{i+1}, \text{lon}_{i+1}])$ stands for the geodesic distance between points p_i and p_j. As depicted in Fig. 1, bearing measures the degree between the line connecting two successive points and true north. The bearing rate for point p_i is the absolute difference between two consecutive bearings

$$B_i = |\alpha_{i+1} - \alpha_i| \tag{4}$$

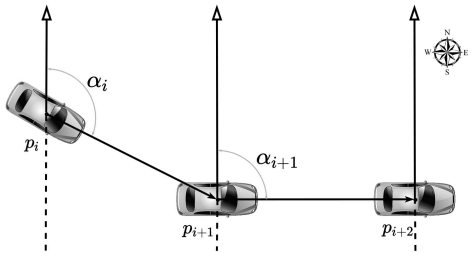

Fig. 1. Demonstration of the bearing in GPS trajectory.

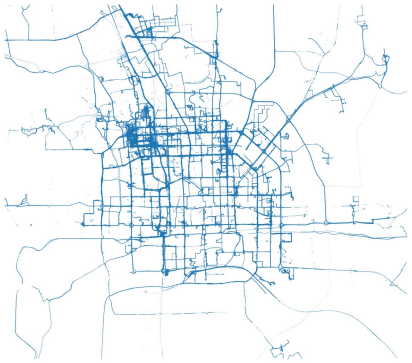

Fig. 2. Visualization of the adopted data set in Geolife.

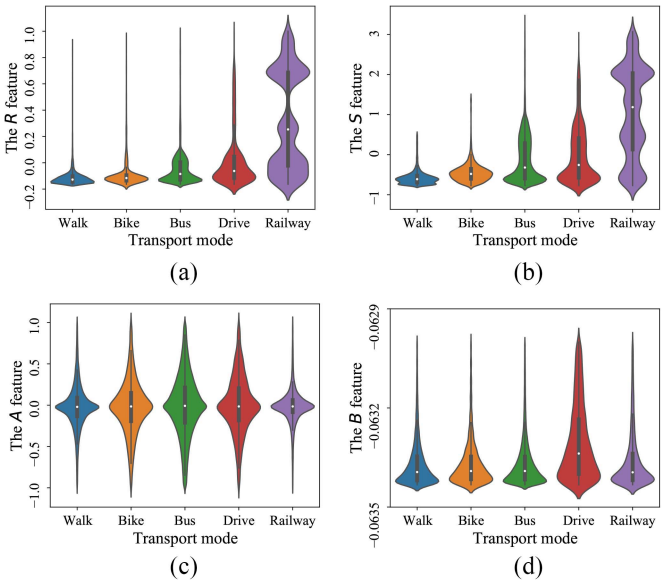

Fig. 3. Violin plot shows the probability distribution of the four motion features after Z-score transformation. There is a plain box plot inside each bar, and the white dot stands for the median. (a) Relative distance. (b) Speed. (c) Acceleration. (d) Bearing rate.

where α_i and α_{i+1} are the bearing of p_i and p_{i+1}, respectively, and computed as

$$\alpha_i = \arctan \frac{\mathcal{G}([\text{lat}_i, \text{lon}_i], [\text{lat}_{i+1}, \text{lon}_i])}{\mathcal{G}([\text{lat}_{i+1}, \text{lon}_i], [\text{lat}_{i+1}, \text{lon}_{i+1}])}. \tag{5}$$

C. Data Process

We adopt the Geolife[2] data set, which contains 182 individuals' daily trajectories recorded every 1–5 s with fine-grained resolution. Traces of 69 users are labeled with travel modes. The studied trajectories are visualized in Fig. 2, which locate in the urban area of Beijing, China.

We obtain individual trajectories and arrange the GPS points in a chronological order, respectively. The following three criteria are utilized to filter abnormal records and choose the aforementioned motion features.

1) Interval Δ_i between two consecutive GPS points p_i and p_{i+1} is examined, i.e., $\Delta_i = t_{i+1} - t_i$. If $\Delta_i \geq 20$ min, the features of point p_i are not calculated but replicated from its previous one p_{i-1} directly.
2) We check the proceeded features and filter out the points whose speed or acceleration exceeds the realistic range of their transportation mode. The GPS points with abnormal geographic location are also discarded.
3) For the labeled trajectory, the segments with less than ten GPS points are abandoned. The remaining points are concatenated as new trajectories in a chronological order.

Then, the treated trajectories are partitioned into fragments with length 2048 (i.e., $L = 2048$), which is the average length of three transportation modes. Each GPS point is denoted by a 4-D tuple (i.e., $K = 4$).

In Fig. 3, we demonstrate the distribution of chosen motion features with respect to five transportation modes after z-score transformation. For a particular motion feature, the transformation can be formulated as

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

where x is a raw value, z is its standard score, and μ and σ are the mean and standard deviation of the population, respectively.

IV. Proposed Methodology

In this section, we elaborate on the proposed approach of end-to-end TMD. The proposed SEED architecture is illustrated in Fig. 4. An encoder-decoder network is used as the backbone, and a similarity entropy-based module is utilized to exploit the unlabeled data. When inferring, we harness a heuristic post-processing method to overcome the jitter problem.

A. Encoder-Decoder Backbone

The proposed model adopts an encoder and a corresponding decoder as a backbone, followed by a point-wise classification layer. A shortcut operation is used to model the correlations between shallow and deep features when restoring input size. The batch normalization layers are applied to each feature map. First, the encoder module is used to extract hierarchical features. Then, the decoder module is established to predict a point-wise probabilistic map according to the context-awareness features.

The encoder network is composed of four convolutional layers to implement feature extraction from motion characteristics

[2]https://www.microsoft.com/en-us/research/project/geolife-building-social-networks-using-human-location-history/

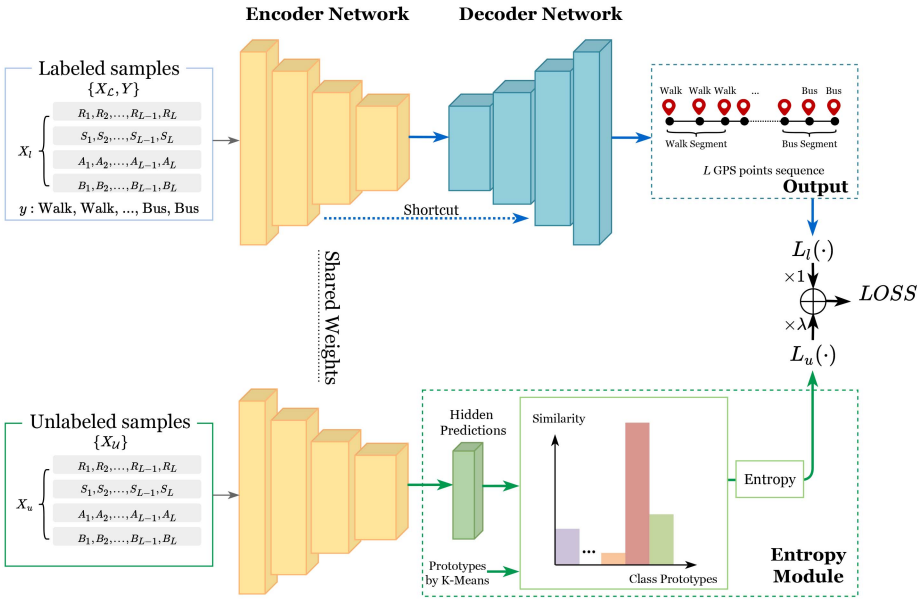

Fig. 4. Architecture of the proposed semi-supervised model, which is composed of a encoder-decoder backbone and a similarity entropy-based module. $\{X_{\mathcal{L}}, Y\}$ is the set of labeled samples and $\{X_{\mathcal{U}}\}$ is the unlabeled set. The term "unlabeled predictions" means the hidden predictions of the encoder with the unlabeled input. The entropy module uses the unlabeled samples to calculate point-wise similarity with the prototypes, and sharpens the similarity distribution between the unlabeled predictions and all prototypes, aiming to make the former close to one prototype only while keeping away from others.

of GPS points. For each convolutional layer, we apply the max-pooling operation to enlarge the receptive fields and adopt the exponential linear unit (ELU) as activation. The transposed convolutional layers are used as a decoder. Hence, there are also four layers that are responsible for restoring input size from the generated feature maps and yielding multichannel prediction. The last output layer is activated by softmax to calculate the probabilities of predicted labels.

The supervised loss L_l is a negative log-likelihood term among the predictions and corresponding point-wise ground truth for all labeled samples. Specifically, for a sample $\{\mathbf{x}_l, y\}$ from the labeled data set $\{X_{\mathcal{L}}, Y\}$, we can get the loss

$$L_l = -\sum_{i=1}^{C} y_i \log p(y_i|\mathbf{x}_l; \mathbf{w}_e, \mathbf{w}_d) \tag{7}$$

where $\mathbf{w}_e$ and $\mathbf{w}_d$ stand for the parameters of the encoder and decoder, respectively, y is the one-hot encoding for C categories, and $p(\cdot)$ is the predicted probability.

B. Similarity-Based Entropy for Semi-Supervised Learning

Entropy regularity was introduced to measure the overlap of categories in [15], which is related to the hypothesis of low-density separation in semi-supervised learning. The unlabeled trajectory can provide us with rich information regarding their similarity to labeled data. The entropy module is illustrated in Fig. 4. First, when inputting labeled data, the features $\mathbf{H} \in \mathbb{R}^{h \times d}$ in the last hidden layer of the encoder network are predicted, where h is the number of features and d is their dimension. Then, the K-Means algorithm is adopted to cluster prototypes for each class as the embedding representations. Hence, the dimension of each prototype is also of size d. In this stage, the ground truth is subsampled to match the size of h. When unlabeled samples are input, the score s_i is calculated to represent the similarity between the hidden predictions with the unlabeled input (termed as "unlabeled predictions") and the prototype $\mathbf{z}_i$, i.e.,

$$s_i = \frac{\langle \mathcal{F}(\mathbf{x}_u; \mathbf{w}_e), \mathbf{z}_i \rangle}{\|\mathcal{F}(\mathbf{x}_u; \mathbf{w}_e)\| \times \|\mathbf{z}_i\|} \tag{8}$$

where s_i stands for the cosine similarity between the unlabeled predictions of encoder and the ith prototype vector $\mathbf{z}_i$, and $\mathcal{F}(\mathbf{x}_u; \mathbf{w}_e)$ means the unlabeled predictions with input $\mathbf{x}_u$ from the unlabeled data set $\{X_{\mathcal{U}}\}$, $\langle , \rangle$ and $\|\cdot\|$ represent the inner product and the vector norm operations, respectively. The principle of entropy regularization for s is to sharpen the similarity distribution between the unlabeled predictions and all the prototypes, resulting in the unsupervised loss term. Namely, the regularizer penalizes the low-confidence predictions to make them close to one prototype only while staying away from others. The unsupervised loss L_u is

$$L_u = -p(s|\mathbf{x}_u; \mathbf{w}_e, \mathbf{z}) \log p(s|\mathbf{x}_u; \mathbf{w}_e, \mathbf{z}) \tag{9}$$

where $\mathbf{w}_e$ is the parameters of the encoder and $p(\cdot)$ represents the similarity distribution between the unlabeled predictions and all prototypes.

C. Training and Inference

Total loss LOSS is the summation of supervised loss L_l and unsupervised losses L_u, i.e.,

$$\text{LOSS} = L_l + \lambda \times L_u \tag{10}$$

where λ is the weight of the unsupervised loss term. The training process is illustrated in Algorithm 1. We first pretrain a model using the labeled data and obtain the hidden predictions of encoder $\mathbf{H}$. Then, the K-Means algorithm is conducted to cluster the point-wise features and gather K embedding representations (i.e., prototypes) for each transportation mode.

Algorithm 1: Simplified Process for the Proposed SEED Model

Inputs: The number of prototypes for each class: K; Weight: λ; Max training epochs: T_1, T_2; Learning rate: α_1, α_2; Training data: $\mathcal{D} = \{(\mathbf{x}_l, y)\}_{l\in\mathcal{L}} \cup \{(\mathbf{x}_u, -1)\}_{u\in\mathcal{U}}$.

Outputs: The weights $\mathbf{w}_e$ and $\mathbf{w}_d$

```
## Pre-training
Initialize encoder and decoder weights w_e, w_d arbitrarily;
for t = 1 to T_1 do
    p(y|x_l; w_e, w_d) ← softmax F(x_l; w_e, w_d);
    L ← Σ_{i=1}^{C} −y_i log p(y_i|x_l; w_e, w_d)   (Negetive Log-Likelihood loss);
    w_e ← w_e − α_1 ∇_{w_e} L;
    w_d ← w_d − α_1 ∇_{w_d} L;
## Group K × C prototypes
Output the hidden predictions for labeled data x_l assigned by encoder: H ← F(x_l; w_e);
Cluster K prototypes on each class from H: z ← [z_1, z_2, ..., z_{K×C}];
## Fine-tuning
for t = T_1 to T_1 + T_2 do
    p(y|x_l; w_e, w_d) ← softmax F(x_l; w_e, w_d);
    p(s|x_u; w_e, z) ← softmax ⟨F(x_u; w_e), z_i⟩ / (‖F(x_u; w_e)‖ × ‖z_i‖);
    L ← Σ_{i=1}^{C} −y_i log p(y_i|x_l; w_e, w_d)   (Negetive Log-Likelihood loss term) ) + λ ×
        Σ_{i=1}^{K×C} −p(s_i|x_u; w_e, z) log p(s_i|x_u; w_e, z)   (Similarity entropy-based loss term);
    w_e ← w_e − α_2 ∇_{w_e} L;
Return w_e, w_d
```

A total of $K \times C$ prototypes are obtained when there are C categories. Finally, we employ all the labeled and unlabeled data to optimize the LOSS with a smaller learning rate, aiming to make the unlabeled predictions from encoder $\mathcal{F}(\mathbf{x}_u; \mathbf{w}_e)$ close to one prototype only.

In the inference stage, to alleviate the jitter problem in which the prediction of a particular point is different from its neighbors, we leverage a majority-voting algorithm [17] for post-processing to smooth the model's predictions. It works by traversing all the output through a sliding window with size M, and the final prediction of point p_i is adjusted to follow the majority of predicted labels in the window $[p_{i-(M/2)}, p_{i-(M/2)+1}, \ldots, p_{i+(M/2)}]$, where M is the voting scope.

In practice, the trained semi-supervised model can be directly applied to infer the mode for each trajectory point. Specifically, when the GPS trajectories are obtained, the four features (i.e., distance, speed, acceleration, and bearing rate) can be calculated according to the relative distance and time interval in trajectories. Then every 2048 points are regarded as input into the trained model, and the model will automatically output the transportation mode regarding each point.

V. Experimental Results and Analysis

In this section, we provide the performance of our proposed method and analyze the numerical results compared with commonly used models.

A. Experimental Setup

In the experiment, 50% of a specific data set is used for training, 30% of data is used as unlabeled samples, and the remaining 20% is used for testing. The twofold cross validation is also carried out. The hyper-parameters are as follows.

1) The size of voting scope: $M = 100$.
2) The weight of the unsupervised term: $\lambda = 0.4$.
3) The number of pretraining epochs: $T_1 = 200$.
4) The number of fine-tuning epochs: $T_2 = 50$.
5) The number of prototypes for each class: $K = 20$.

The neural networks are all implemented using Pytorch and optimized by the Adam method. All experiments are conducted on a workstation with an Intel Core E5-2620 CPU and a 12-GB Nvidia GeForce GTX Titan Xp Graphics Card.

The performance metrics for evaluation are as follows.

1) Accuracy by length

$$A_L = \frac{\sum_{j=0}^{N} V_j}{\sum_{i=0}^{N} R_i}, \quad V_j = \begin{cases} R_j, & \text{if } \hat{y}_j = y_j \\ 0, & \text{else} \end{cases} \tag{11}$$

where N stands for the total number of GPS points, R_j represents the relative distance feature of the jth point p_j, and $\hat{y}_j$ and y_j are the predicted and true label of p_j, respectively.

2) Accuracy by duration

$$A_D = \frac{\sum_{j=0}^{N} W_j}{\sum_{i=0}^{N} \Delta_i}, \quad W_j = \begin{cases} \Delta_j, & \text{if } \hat{y}_j = y_j \\ 0, & \text{else} \end{cases} \tag{12}$$

where Δ_j is the time interval between p_j and its successive point p_{j+1}.

3) Intersection over Union (IoU) across each class

$$\text{IoU} = \frac{T_P}{T_P + F_P + F_N} \tag{13}$$

where T_P, F_P, and F_N are the true positive, false positive, and false negative points, respectively.

4) Precision

$$\text{Precision} = \frac{T_P}{T_P + T_N}. \tag{14}$$

5) Recall

$$\text{Recall} = \frac{T_P}{T_P + F_N}. \tag{15}$$

6) Mean IoU ($\overline{\text{IoU}}$)

$$\overline{\text{IoU}} = \frac{1}{C} \sum_{n=1}^{C} \text{IoU}_n \tag{16}$$

where C represents the number of classes.

TABLE II
PERFORMANCE OF SUPERVISED AND SEMI-SUPERVISED MODELS

Training ratio	Models	A_L	A_D	IoU				
				Walk	Bike	Bus	Drive	Railway
25%	Supervised-only	0.607	0.714	**0.660±0.011**	**0.673±0.03**	0.555±0.02	0.361±0.006	0.674±0.015
	Semi-supervised	**0.617**	**0.717**	0.651±0.022	0.66±0.043	**0.561±0.024**	**0.371±0.015**	**0.686±0.014**
50%	Supervised-only	0.653	0.740	0.671±0.012	0.694±0.022	**0.585±0.028**	0.429±0.007	0.729±0.019
	Semi-supervised	**0.670**	**0.743**	**0.672±0.008**	**0.697±0.019**	0.578±0.034	**0.456±0.013**	**0.742±0.026**
75%	Supervised-only	0.703	0.757	0.679±0.007	**0.710±0.026**	**0.597±0.019**	0.450±0.009	0.764±0.018
	Semi-supervised	**0.706**	**0.763**	**0.681±0.008**	0.709±0.027	0.596±0.011	**0.481±0.027**	**0.768±0.018**
100%	Supervised-only	0.717	0.777	0.688±0.011	0.724±0.023	0.640±0.022	0.509±0.007	**0.791±0.024**
	Semi-supervised	**0.723**	**0.780**	**0.691±0.012**	**0.730±0.023**	**0.641±0.017**	**0.531±0.028**	0.790±0.025
Average Improvement		↑ 1.4%	↑ 0.6%	↓ 0.2%	↓ 0.4%	↓ 0.1%	↑ 5.1%	↑ 1.1%

B. Benchmarks

To illustrate the advantages of our approach, we compare the proposed model with two-stage methods based on the DT algorithm, which are the same as that in [14].

1) *Same-Length-Based DT (SLDT):* First, GPS trajectories are partitioned into several segments, and each segment has the same length after being partitioned. Then, the C4.5 DT is used to infer the mode of each segment.
2) *Same-Duration-Based DT (SDDT):* The segments are of the same duration instead of the same length while other steps remain the same as model 1). Additionally, we also conduct four competitive end-to-end models:
3) *CNN:* The model has six CNN layer.
4) *SECA [21]:* The model has six CNN layers, and follows the autoencoder framework for semi-supervised learning;
5) *Pseudo Label [21]:* The model has six CNN layers, and uses the self-training strategy for semi-supervised learning;
6) *Ensemble CNN [17]:* The model ensembles six CNN submodels and uses majority voting strategy for supervised learning.
7) *BiLSTM CNN:* The model has one-layer bidirectional LSTM module and a CNN layer;
8) *ResNet With 8-Times Up Sampling [ResNet-Up(8×)]:* The 1-D "ResNet18" [54] is taken as a backbone to capture hierarchical correlations and the predictions are generated by a 8-times upsampling layer with the "nearest" interpolation.

It should be noted that the benchmarking models 4–6 are originally proposed for the segment-level inference, which can not process the point-level input. So we partition GPS trajectories into several segments. Each segment has the same length and is assigned with the majority class as a label after being partitioned.

C. Performance

First, we validate the superiority of our proposed semi-supervised model over its peers. With varying amounts of labeled data, we build the supervised and semi-supervised models separately, and then evaluate them on the same test set. Table II provides the evaluated results between the supervised-only and semi-supervised algorithms. The best performance is shown in bold. With 25% labeled data, our semi-supervised model demonstrates slight improvement. As the labeled data increases, the proposed method has a more significant enhancement, especially in the IoU of "Drive" class. The average improvement among all the performance metrics is calculated in the last row in Table II. Our semi-supervised model increases the IoU of "Drive" by +5.1% maximally and reduces 0.4% in "Bike." Table III exhibits the comprehensive performance between SEED and its competitive methods. It can be seen that the end-to-end manners outperform the two-stage models generally. The Ensemble-CNN outperforms all the benchmarks and achieves 7% improvement against the CNN model. Compared with the Ensemble-CNN, the improvement of our method is encouraging (+5.1%) on Recall metric. The remaining metrics of the proposed model are also increased by 1.5%–4.0%. These indicate that SEED works well for end-to-end point-wise mode detection.

TABLE III
PERFORMANCE OF DIFFERENT MODELS

Models	A_L	A_D	$\overline{\text{IoU}}$	Recall	Precision
SLDT	0.423	0.449	0.228	0.443	0.417
SDDT	0.427	0.455	0.249	0.449	0.425
FNN	0.630	0.573	0.410	0.613	0.570
CNN	0.660	0.670	0.514	0.694	0.678
SECA	0.661	0.676	0.541	0.722	0.683
Pseudo-label	0.653	0.686	0.542	0.711	0.674
BiLSTM-CNN	0.680	0.711	0.592	0.754	0.731
ResNet-Up(8×)	0.710	0.751	0.638	0.783	0.769
Ensemble-CNN	0.711	0.749	0.652	0.782	0.773
SEED(ours)	**0.723**	**0.780**	**0.677**	**0.822**	**0.793**

Fig. 5 illustrates the precision and recall of compared models under different proportion of labeled data, i.e., 25%, 50%, 75%, and 100%. Compared with the two semi-supervised methods

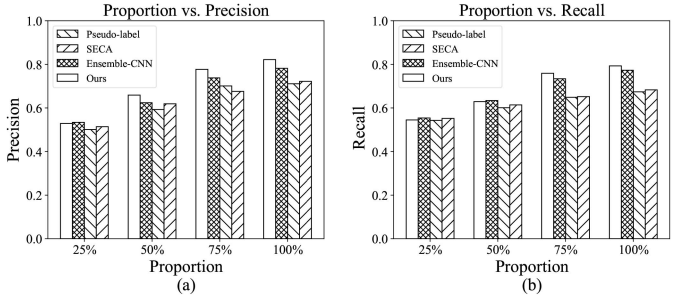

Fig. 5. Precision and recall of compared models under different proportion of labeled data. (a) Results of precision with 25%, 50%, 75%, and 100% of labeled data, respectively. (b) Results of recall with 25%, 50%, 75%, and 100% of labeled data, respectively.

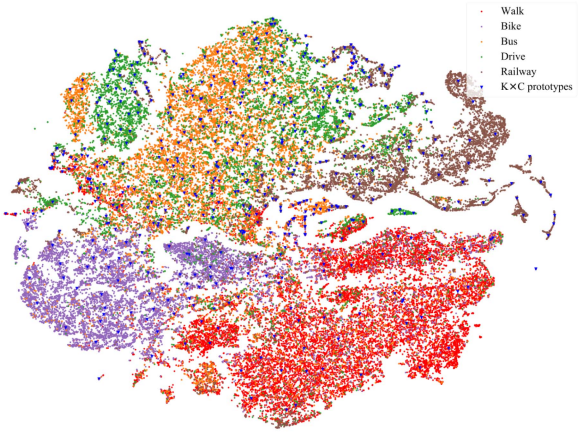

Fig. 6. 2-D t-SNE embedding representations of the hidden features predicted by the encoder.

SECA and pseudo-label models, our proposed model gains remarkable performance across all metrics. When the labeled data is with a small proportion (25%), the performance of our model does not surpass its three peers. The reason may be that there are so little labeled data that the clustered prototypes becomes biased, resulting in unimpressive performance.

Then, we illustrate the hidden predictions $\mathbf{H}$ of the encoder after dimension reduction by t-SNE [55] in Fig. 6, where the blue triangles represent the clustered class-wise prototypes. In the experiment, we find that the model works well when setting $K = 20$, namely, there are total 100 prototypes since $C = 5$. From the embedding representations, the hidden features for "Walk" (red dots), "Bike" (purple dots), and "Railway" (brown dots) are all with promising discrimination. However, the decision boundary between classes "Bus" (orange dots) and "Drive" (green dots) is ambiguous. This may be since they are both motorized and therefore the similar motion characteristics arise, leading to difficulties for the model to distinguish the two modes. Such a phenomenon also affects the follow-up analysis of the predicted results. Fig. 7 displays the percent stacked bars, in which the horizontal axis indicates the ground-truth transportation modes and the vertical axis represents the proportion of predicted classes in the ground truth. The color of each rectangle corresponds to the predicted labels, whose length is determined by the corresponding occupancy in the total result across the class. It can be seen that about 20% of "Drive" GPS points are predicted as "Bus," while about 10%

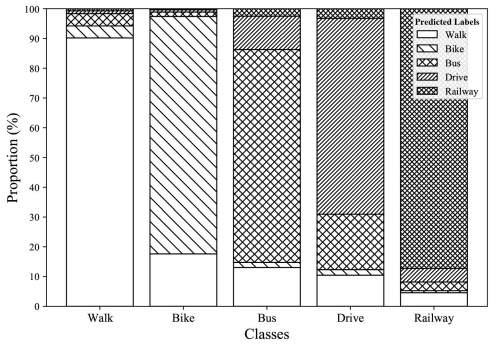

Fig. 7. Percent stacked bars for the point-wise prediction. The horizontal axis indicates the ground-truth transportation modes, while the vertical axis represents the proportion of predicted labels in the ground truth.

TABLE IV
ABLATION EXPERIMENTAL RESULTS ON THE POST-PROCESSING AND SIMILARTY ENTROPY-BASED COMPONENTS

Models	A_L	A_D	$\overline{\text{IoU}}$	Recall	Precision
w/o SE	0.717	0.770	0.670	0.821	0.785
with GE	0.713	0.775	0.668	0.820	0.785
w/o PP	0.701	0.757	0.644	0.793	0.773
SEED(ours)	**0.721**	**0.781**	**0.677**	**0.822**	**0.793**

and 15% of "Bus" GPS points are predicted as "Drive" and "Walk," respectively. "Walk" gets a higher precision (90%), but it also accounts for a large occupancy in other classes, indicating that the model prefers to classify the GPS points as "Walk" mode.

D. Ablation Study

Since SEED contains semi-supervised and post-processing modules, we perform the following ablation experiments to provide insights into these two components of the proposed objective function specifically.

1) *w/o SE:* A model without similarity entropy-based (SE) component, namely, the supervised learning-only model.
2) *with GE:* Replacing the similarity entropy-based component by a generalized-entropy (GE) loss.
3) *w/o PP:* A model without majority-voting post-processing component.

Their performance is summarized in Table IV. The model without post-processing (**w/o PP**) component causes the largest degradation ($-2.9\% \sim -5.1\%$), which proves that the post-processing method is most crucial. The model with the GE module (**with GE**) demonstrates worse results (about $-0.2\% \sim -1.3\%$ degradation) than the proposed method and even becomes inferior to the supervised-only model (**w/o SE**), which indicates that the GE component may hurt the model and the proposed similarity entropy-based module is effective.

VI. CONCLUSION

With the proliferation of IoT devices, extensive digital traces have been collected and become valuable sources for understanding human behavior. In this article, we propose a semi-supervised learning approach to identify transportation modes from individual GPS trajectories. Contrast to previous

studies, we reframe a two-stage detection pipeline to an end-to-end manner and exploit unlabeled data to improve the model's generalization ability. The proposed approach convincingly outperforms benchmarks which include typical two-stage and other end-to-end methods. Moreover, the effectiveness of the similarity entropy-based module is verified by ablation experiments. The semi-supervised learning model can also be used in other off-line sequential classification tasks, such as trip purpose identification or signal analysis.

The limitation of our method is that the proposed model can only handle fixed-length input. Additionally, the model tends to make more predictions about the "Walk" mode, and needs to be strengthened for the recognition between "Bus" and "Drive" modes. Therefore, efforts should be made to further employ the free-length input and take advantage of the road network for better discrimination during mode detection. Considering the motion features may vary from person to person, the difference among individuals is also worthy to be discovered. These may provide novel insights to tackle the issues.

REFERENCES

[1] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[2] Y. Gao *et al.*, "Parallel end-to-end autonomous mining: An IoT-oriented approach," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1011–1023, Feb. 2020.

[3] M. Ghahramani, M. Zhou, and G. Wang, "Urban sensing based on mobile phone data: Approaches, applications, and challenges," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 3, pp. 627–637, May 2020.

[4] T. Bantis and J. Haworth, "Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics," *Transp. Res. C, Emerg. Technol.*, vol. 80, pp. 286–309, Jul. 2017.

[5] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[7] Z. Li *et al.*, "A multi-stream feature fusion approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, Oct. 7, 2020, doi: 10.1109/TITS.2020.3026836.

[8] G. Xiong *et al.*, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA J. Automatica Sinica*, vol. 2, no. 3, pp. 320–333, Jul. 2015.

[9] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4063–4071, Oct. 2020.

[10] T. Feng and H. J. P. Timmermans, "Transportation mode recognition using GPS and accelerometer data," *Transp. Res. C, Emerg. Technol.*, vol. 37, pp. 118–130, Dec. 2013.

[11] A. Jahangiri and H. A. Rakha, "Applying learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2406–2417, Oct. 2015.

[12] 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 Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5473–5485, Sep. 2021.

[13] T. Zhou, M. Chen, and J. Zou, "Reinforcement learning based data fusion method for multi-sensors," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 6, pp. 1489–1497, Nov. 2020.

[14] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw gps data for geographic applications on the Web," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 247–256.

[15] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems*, vol. 17. Red Hook, NY, USA: Curran Assoc., 2004, pp. 529–536.

[16] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," in *Advances in Neural Information Processing Systems*, vol. 33. Red Hook, NY, USA: Curran Assoc., 2020.

[17] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Ensemble convolutional neural networks for mode inference in smartphone travel survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2232–2239, Jun. 2020.

[18] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, "Deep feature extraction from trajectories for transportation mode estimation," in *Proc. Pac. Asia Conf. Knowl. Disc. Data Min.*, 2016, pp. 54–66.

[19] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 360–371, Jan. 2018.

[20] G. Jiang, S.-K. Lam, P. He, C. Ou, and D. Ai, "A multi-scale attributes attention model for transport mode identification," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 22, 2020, doi: 10.1109/TITS.2020.3008469.

[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 Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 1010–1023, May 2020.

[22] J. James, "Semi-supervised deep ensemble learning for travel mode identification," *Transp. Res. C, Emerg. Technol.*, vol. 112, pp. 120–135, Mar. 2020.

[23] M. C. Yu, T. Yu, S. C. Wang, C. J. Lin, and E. Y. Chang, "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[24] T. H. Vu, L. Dung, and J.-C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 392–396.

[25] G. Asci and M. A. Guvensan, "A novel input set for LSTM-based transport mode detection," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2019, pp. 107–112.

[26] C. Carpineti, V. Lomonaco, L. Bedogni, M. D. Felice, and L. Bononi, "Custom dual transportation mode detection by smartphone devices exploiting sensor diversity," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2018, pp. 367–372.

[27] I. Anderson and H. Muller, "Practical activity recognition using GSM data," in *Proc. 5th Int. Semantic Web Conf. (ISWC)*, vol. 1, 2006, pp. 1–8.

[28] B. Assemi, H. Safi, M. Mesbah, and L. Ferreira, "Developing and validating a statistical model for travel mode identification on smartphones," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1920–1931, Jul. 2016.

[29] M. Thejaswini, P. Rajalakshmi, and U. B. Desai, "Novel sampling algorithm for human mobility-based mobile phone sensing," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 210–220, Jun. 2015.

[30] S. A. Hoseinitabatabaei, Y. Fathy, P. Barnaghi, C. Wang, and R. Tafazolli, "A novel indexing method for scalable IoT source lookup," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2037–2054, Jun. 2018.

[31] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.

[32] B. Wang, L. Gao, and Z. Juan, "Travel mode detection using GPS data and socioeconomic attributes based on a random forest classifier," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1547–1558, May 2018.

[33] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 312–321.

[34] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, Jun. 2010.

[35] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web (WWW)*, 2009, pp. 791–800.

[36] H. Gjoreski *et al.*, "The university of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42592–42604, 2018.

[37] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen, "Benchmarking the SHL recognition challenge with classical and deep-learning pipelines," in *Proc. ACM Int. Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, 2018, pp. 1626–1635.

[38] H. Han, W. Ma, M. Zhou, Q. Guo, and A. Abusorrah, "A novel semi-supervised learning approach to pedestrian reidentification," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 3042–3052, Feb. 2021.

[39] M. Zhou, Y. Tang, Z. Tian, L. Xie, and W. Nie, "Robust neighborhood graphing for semi-supervised indoor localization with light-loaded location fingerprinting," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3378–3387, Oct. 2018.

[40] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[41] 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*. Red Hook, NY, USA: Curran Assoc., 2017, pp. 1195–1204.
[42] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017. [Online]. Available: arXiv:1710.09412.
[43] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2019, pp. 5049–5059.
[44] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.
[45] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn.*, vol. 3, 2013, pp. 1–6.
[46] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 1135–1142.
[47] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Semi-supervised GANs to infer travel modes in GPS trajectories," *J. Big Data Anal. Transp.*, vol. 2021, pp. 1–11, Jul. 2021, doi: 10.1007/s42421-021-00047-y.
[48] R. Zhang, X. Li, H. Zhang, and F. Nie, "Deep fuzzy k-means with adaptive loss and entropy regularization," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2814–2824, Nov. 2020.
[49] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017. [Online]. Available: arXiv:1701.06548.
[50] W. Dean, "De-biasing weakly supervised learning by regularizing prediction entropy," in *Proc. Int. Conf. Learn. Represent. Workshops*, 2019, pp. 1–5.
[51] T. H. Vu, H. Jain, M. Bucher, M. Cord, and P. Perez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2517–2526.
[52] T. Kalluri, G. Varma, M. Chandraker, and C. Jawahar, "Universal semi-supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 5259–5270.
[53] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8049–8057.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
[55] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

Zhishuai Li received the B.E. degree in automation from China University of Petroleum, Qingdao, China, in 2017. He is currently pursuing the Ph.D. degree with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing.

His research interests include intelligent optimization algorithms and intelligent transportation systems.

Gang Xiong (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 1996.

He is a Professor with the Beijing Engineering Research Center of Intelligent Systems and Technology, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is also with the Guangdong Engineering Research Center of 3-D Printing and Intelligent Manufacturing, The Cloud Computing Center, Chinese Academy of Sciences, Dongguan, China. His research interests include parallel control and management, cloud computing and big data, intelligent manufacturing, and intelligent transportation systems.

Zebing Wei (Graduate Student Member, IEEE) received the B.E. degree in automation from Dalian University of Technology, Dalian, China, in 2019. He is currently pursuing the MA.Eng. degree with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China, and the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

His research interest includes intelligent transportation and big data mining.

Yisheng Lv (Senior Member, IEEE) received the B.E. and M.E. degrees in transportation engineering from Harbin Institute of Technology, Harbin, China, in 2005 and 2007, respectively, and the Ph.D. degree in control theory and control engineering from the Chinese Academy of Sciences, Beijing, China, in 2010.

He is an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. He is also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing. His research interests include traffic data analysis, dynamic traffic modeling, and parallel traffic management and control systems.

Noreen Anwar received the bachelor's degree from the University of Central Punjab, Lahore, Pakistan, in 2016, and the master's degree from Southwest Jiaotong University, Chengdu, China, in 2021.

Since July 2007, she has been worked as an external trainee with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include artificial intelligence, machine learning, intelligent transportation systems, and intelligent robotics.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the University of Arizona, Tucson, AZ, USA, in 1990, and became a Professor and the Director of the Robotics and Automation Lab (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, where he was appointed as the Director of the Key Lab of Complex Systems and Intelligence Science. From 2006 to 2010, he was the Vice President for Research, Education, and Academic Exchanges, Institute of Automation, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. His current research focuses on methods and applications for parallel systems, social computing, parallel intelligence and knowledge automation.

He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, and the *IEEE Intelligent Transportation Systems Magazine* from 2006 to 2007, and an EiC of IEEE INTELLIGENT SYSTEMS from 2009 to 2012, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016. He is currently an EiC of IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, the Founding EiC of IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and *Chinese Journal of Command and Control*. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, and ASME conferences. He was the President of IEEE ITS Society (2005–2007), Chinese Association for Science and Technology (USA) in 2005, the American Zhu Kezhen Education Foundation (2007–2008), the Vice President of the ACM China Council (2010–2011), and the Vice President and Secretary General of Chinese Association of Automation (CAA, 2008–2018). He is currently the President of CAA Supervision Council and the IEEE Council on RFID, and the Vice President of IEEE SMC Society.