Semi-supervised Deep Ensemble Learning for Travel Mode Identification

James J.Q. Yu[a,*]

[a] *Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China*

Abstract

Travel mode identification is among the key problems in transportation research. With the gradual and rapid adoption of GPS-enabled smart devices in modern society, this task greatly benefits from the massive volume of GPS trajectories generated. However, existing identification approaches heavily rely on manual annotation of these trajectories with their accurate travel mode information, which is both economically inefficient and error-prone. In this work, we propose a novel semi-supervised deep ensemble learning approach for travel mode identification to use a minimal number of annotated data for the task. The proposed approach accepts GPS trajectories of arbitrary lengths and extracts their latent information with a tailor-made feature engineering process. We devise a new deep neural network architecture to establish the mapping from this latent information domain to the final travel mode domain. An ensemble is accordingly constructed to develop proxy labels for unannotated data based on the rare annotated ones so that both types of data contribute to the learning process. Comprehensive case studies are conducted to assess the performance of the proposed approach, which notably outperforms existing ones with partially-labeled training data. Furthermore, we investigate its robustness to noisy data and the effectiveness of its constituting components.

1. Introduction

Travel mode information is critical to human mobility analytics, which is a key component of intelligent transportation research [1, 2]. Accurate travel mode data associated with the corresponding trajectory information serve as the foundation to solve many transport-related problems, e.g., transportation planning, operation, and control, etc [3–6]. Conventional travel mode information was obtained through rigorously designed travel surveys, which were costly and can only develop small sample sizes (mostly in thousands) [7]. In the past few decades, advanced data collection methods, especially global positioning system (GPS)-enabled civilian equipment and smartphone devices, provide system operators with a massive volume of user trajectory data comprised of discrete GPS records. These trajectories grant the possibility of new transportation applications such as online traffic accident detection and speed estimation [8]. Travel mode data are also enriched by analyzing GPS trajectories via mode identification techniques, and public domain data sets have been published to facilitate related research [9, 10].

Generally speaking, GPS-enabled devices record the positional information of trips and do not have explicit knowledge on the current travel mode. Travel mode identification techniques aim to infer travel mode information based on these GPS trajectories. As firstly proposed by Zheng *et al.* [9], typical approaches adopt a two-step data analytic protocol to perform the identification: 1) feature engineering to create data features with domain knowledge, and 2) supervised learning to construct a machine learning model for the task. Existing approaches employ hand-crafted data features computed by descriptive statistics of motion and displacement characteristics such as velocity, and frequency domain characteristics such as spectral spread [5, 9, 11]. With the constructed feature data, a wide variety of machine learning algorithms can be utilized for mode classification, including but not limited to decision tree, random forest, support vector

[*] Corresponding author
Email address: yujq3@sustech.edu.cn (James J.Q. Yu)

machine, and deep neural networks [12, 13]. Despite the limitations of deep learning methods such as lack of explanability and requires a huge volume of GPS data to function well, the research community is witnessing increasing effort in adopting deep learning techniques in travel mode identification [7, 11].

However, there is a significant research gap in contemporary travel mode identification research. A majority of the current work relies on end-to-end fully-supervised training techniques to handle the task, see [5, 9, 13, 14] for some examples. Such learning paradigm requests the training data to provide ground truth travel mode information together with respective GPS trajectories. While the research community has witnessed public domain data sets with such information, e.g., GeoLife [9], the volume of travel mode-annotated data is minuscule compared with unlabeled GPS trajectories of trips [7]. This is due to the nature of these trajectories: GPS devices can automatically generate GPS records without human assistance, but the corresponding travel mode information can only be provided by the device user. Hence, acquiring labeled data is more expensive and labor-intensive than auto-generated data.

In the meantime, the massive volume of unlabeled data that are easily accessible can potentially improve the identification accuracy with their additional latent information if the fully-supervised training limitation can be relaxed, which makes semi-supervised learning a promising solution to enhance existing travel model identification techniques [15]. By semi-supervised learning, only a small portion of trajectories in a data set needs to be annotated, and the learning algorithm can also make use of the other unlabeled data for training. Recently, a first semi-supervised travel mode identifier based on convolution neural network and auto-encoder is proposed in [7]. The approach, however, relies heavily on the sample size and density of labeled data to train the encoder as a supervised classifier. This characteristic hinders the identifier from developing accurate travel mode information when labeled data is scarce, e.g., 62.9% accuracy given 10% of all trajectories annotated [7]. A new semi-supervised travel mode identifier is required to fully utilize the unlabeled data.

To bridge the research gap, we propose a novel semi-supervised deep ensemble learning-based travel mode identification approach. The proposed identifier focuses on producing proxy labels for unlabeled data, which can be used as training targets together with the original annotated data. While these proxy labels do not reflect the respective ground truth travel modes, they do provide sample population distribution information for learning. The identifier is formulated based on an ensemble of four long short-term memory (LSTM) empowered deep neural networks (DNN), which take time domain trajectory attributes and frequency domain statistics developed by discrete Fourier transform (DFT) and wavelet transform (DWT). The main contributions of this paper are summarized as follows:

- We propose a new DNN architecture for travel mode identification. The model can be trained end-to-end with supervised learning.
- We propose a proxy-label-based semi-supervised learning algorithm. It utilizes a DNN ensemble to leverage the unlabeled GPS trajectories.
- We conduct a series of comprehensive case studies to illustrate the performance and investigate the influence of GPS position accuracy. Empirical studies also reveal the significance of the constituting components in the identifier.

The rest of this paper is organized as follows. Section 2 introduces the background of travel mode identification research. Section 3 presents the new DNN architecture and its associated feature engineering techniques. Section 4 elaborates on the proposed semi-supervised learning scheme with proxy label and ensemble learning. Section 5 demonstrates a series of case studies to reveal the efficacy of the proposed identifier. Finally, Section 6 concludes this paper.

2. Background

Travel mode identification is a significant task in human activity recognition and has attracted much research effort [5]. The task has been accomplished using various approaches in the literature, which can be generally classified into three categories: rule-based classifier, machine learning techniques, and discrete choice models [16]. Additionally, multiple data sources are utilized in the identification – GPS trajectory, smartphone sensor data, geographic information system, socioeconomic attributes to name a few. Wu *et al.* [17] summarizes identifiers based on GPS data and Elhoushi *et al.* [18] presents a comprehensive survey on travel mode identification approaches with different data sources. Nonetheless, there are two notable issues with mode identification with data other than GPS trajectories, i.e., data communication cost and data acquisition difficulty. According to the survey by Elhoushi *et al.* [18], mode identification can achieve the best performance when 20 Hz accelerometer data, 100 Hz gyroscope or barometer data are available. While these data can be easily obtained by contemporary smartphones, transmitting them to the control center incurs a considerable communication burden. Granted that well trained identifiers can be distributed to the end-user, executing the identification process may induce significant power consumption. On the other hand, the construction of accurate geographic information system and socioeconomic attribute database is sometimes difficult due to regulation reasons or economic concerns, and maintaining them also requires a long-term support. Therefore, this work focuses on GPS-based travel mode identifiers, and we present recent work empowered by machine learning techniques in this section.

A fundamental requirement of using GPS data for travel mode identification is that the data shall cover a majority of the investigated region and population, so that the latent characteristics of trajectories can be captured by learning techniques. Additionally, travel modes cannot be overly biased, otherwise a skewed model may be produced that renders inferior performance on data with a different mode distribution. In this area of research, Zheng *et al.* [9] is among the trailblazers of GPS trajectory-based travel mode identification. Together with later work Zheng *et al.* [19], the authors established the well-recognized segmentation-identification two-step framework for the task. With domain-specific commonsense information, a trajectory is segmented into multiple triplegs, each of which is considered to have a unique travel mode. Statistics of each tripleg, e.g., mean and variance of speed, are employed in several machine learning techniques for mode classification. The result demonstrates the importance of feature engineering over machine learning choice and advocates the use of both fundamental and advanced data features for better performance.

Following this line of work, researchers devoted much effort in improving both aspect of the identification task, i.e., feature engineering and machine learning. To list some recent results on the new data features developed, Dabiri and Heaslip [20] included jerk and rate of change in the heading direction of entities – which were shown to be effective through empirical studies – into the statistical feature set. Reference Mäenpää *et al.* [14] discovered that spectral features of entity speed and acceleration greatly boost the system performance based on statistical tests. Güvensan and Asci [21] further summarized a list of well-performing frequency-domain features developed by Fourier transform for the task. Wang *et al.* [5] emphasized that other sources of related information, e.g., socioeconomic attributes, can also improve the identification accuracy.

On the other hand, contributed by the recent breakthrough in machine learning techniques, especially deep learning approaches, travel mode identification is notably enhanced due to their excellent latent information extraction capability with sufficient data [22]. A wide range of canonical and new machine learning techniques demonstrated their efficacy in handling this classification task, e.g., random forest [5], multi-layer perceptron [23], convolution neural network [20], recurrent neural network [24], etc. Nonetheless, all approaches above fall into the supervised learning category that all training data need to be properly annotated with travel mode information, which is not practical in the contemporary big data era.

To handle this problem, Dabiri *et al.* [7] and Yazdizadeh *et al.* [25] independently presented two semi-supervised travel mode identification approaches based on auto-encoder and generative adversarial network, respectively. In Dabiri *et al.* [7], the proposed model tried to establish a trajectory-to-latent-information mapping so that each trajectory can be expressed with a low-dimensional latent representation, which is considered as the input data for classification. In this process, both the mapping and the subsequent classification are learned from the small volume of annotated data. However, this model does not perform well when annotated data are scarce, as the accuracy of mode classifier cannot be guaranteed by merely training with the available information. Furthermore, the fixed-length latent representation prohibits the model from analyzing trajectories with varying length, which are common in general data sets. In Yazdizadeh *et al.* [25], both annotated data and synthesized trajectories from random noise are employed to train a generative model, in which the discriminator is considered as the final classifier. Nonetheless, necessary

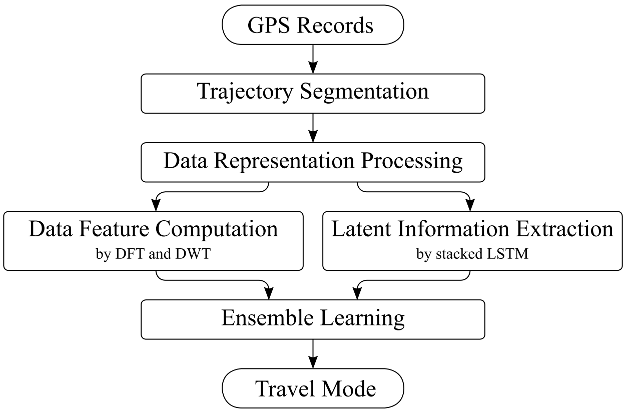


Figure 1: Data processing flow of the proposed identifier.

technical details are missing from the presentation so that assessing its validity is difficult. The inclusion of an "unsupervised learning loss" in the training process may also imply a possible data leakage issue. To handle these drawbacks, we propose a new semi-supervised travel mode identification based on proxy labels and ensemble learning in the following sections.

3. Travel Mode Identification with Deep Neural Network

In this section, we introduce a new deep neural network-based travel mode identifier. We first present the detailed architecture of the proposed identifier and then elaborate on the data pre-processing techniques employed in handling GPS records in raw trajectories. Subsequently, the data feature computation and latent information extraction schemes are devised with brief introductions to related preliminaries.

3.1. System Architecture

The data processing flow of the proposed travel mode identifier is depicted in Fig. 1. The identifier takes raw GPS records as data input and develops corresponding travel modes by a series of processing steps. Specifically, input GPS records are first segmented into multiple GPS trajectories, each of which belongs to an arbitrary travel mode, and two adjacent ones fall into different modes. While the time-series GPS records in trajectories already contain sufficient information for the classification, high-level data representations can be further established with GPS record processing algorithms to expose more relevant data properties. This data processing step greatly benefits subsequent calculations, i.e., data feature computation by DFT and DWT, and latent information extraction by stacked LSTM that aim to provide classification basis for the final ensemble learning identifier. The output is considered as a travel mode inference of the input GPS records, and this completes the identification process.

In this design, the five "blocks" shown in Fig. 1 are the constituting components of the identifier, whose efficacy greatly influences the system performance. In this work, we follow the previous literature in designing the GPS record processing algorithm that includes both the segmentation and representation. Subsequently, DFR and DWT are adopted to investigate the global and local frequency-domain features in the input data, which are shown to be discriminating ones in general signal processing tasks. Additionally, a stacked LSTM is constructed to further extract the time-domain data-correlation (latent information) thanks to its outstanding feature-extraction capability. Finally, all intermediate data are post-processed by an ensemble of three deep neural networks, which serves as the travel mode identifier.

3.2. GPS Record Processing

The GPS record processing step comprises trajectory segmentation and data representation processing[1] sub-steps, which are executed sequentially. In particular, GPS records are typically presented in the form of a sequence of GPS points $\{P_1, P_2, \cdots, P_M\}$ of length M, each of which is defined by $P_i = \langle \text{lat}_i, \text{lng}_i, t_i \rangle$ – a 3-tuple of latitude, longitude, and timestamp. Since each point corresponds to a geographical location, connecting these points constructs a trajectory, and segmentation refers to dividing the trajectory into multiple triplegs according to various travel modes [12]. Each tripleg contains only GPS points correspond to the same travel mode, and identifications are performed on the tripleg-level.

In this work, we follow the previous change-point-based segmentation approach in Zheng *et al.* [9] to create triplegs. The approach is established on assumptions of real-world scenarios, i.e., `walk` is the transition mode between other modes, and people must stop when they change travel modes. Consequently, trajectory segmentation is conducted as follows [9]:

1. Use a loose maximum velocity and acceleration to identify GPS points in `walk` mode from others, and construct triplegs for each group of `walk`/non-`walk` mode points.
2. If the number of consecutive GPS points in a tripleg does not exceed a threshold, it is called "uncertain" [9]. If the number of consecutive uncertain triplegs exceeds another threshold, they are merged as a new one. Otherwise, these triplegs are merged into the tripleg immediately ahead of them.
3. The start and end points of triplegs with `walk` mode points are considered as mode changing points. They are used to segment the trajectory.

Additionally, we also employ an interval based segmentation scheme in step 1) above. If the time interval between two consecutive GPS points exceeds 1 min, the tripleg is further segmented between them.

The second step in GPS record processing is data representation processing. While learning directly from GPS point series is possible, the potential heterogeneous sample interval hinders statistical learning mechanisms from extracting data characteristics easily [13, 19]. In addition, the domain knowledge incorporated in advanced data representation can better help the learning models to focus on distinguishing properties of travel modes instead of common statistical feature extraction tasks [13]. A commonly adopted pre-processing technique is to combine the geographical and temporal information to develop speed-related data features. Specifically, we employ the speed, acceleration, jerk, and turn rate time-series of each length-N tripleg as the first set of data representation as follows:

$$s_i = d_i/\Delta t_i,\ 1 \le i < N; \qquad s_N = s_{N-1}; \tag{1a}$$

$$a_i = (s_{i+1} - s_i)/\Delta t_i,\ 1 \le i < N; \qquad a_N = 0; \tag{1b}$$

$$k_i = (a_{i+1} - a_i)/\Delta t_i,\ 1 \le i < N; \qquad k_N = 0; \tag{1c}$$

$$r_i = (b_{i+1} - b_i)/\Delta t_i,\ 1 < i < N; \qquad r_1 = r_N = 0; \tag{1d}$$

where s_i, a_i, k_i, and r_i are the speed, acceleration, jerk, and turn rate values, respectively. Symbols d_i and b_i denote the distance and absolute bearing of the i-th movement in the tripleg, respectively, and $\Delta t_i = t_{i+1} - t_i$. For better precision, we adopt Vincenty's formulae [26] to calculate the distance of two GPS points, denoted by $d_i = \text{Vinc}(\text{lat}_i, \text{lng}_i, \text{lat}_{i+1}, \text{lng}_{i+1})$. Subsequently, the bearing b_i can be constructed considering the true North by Dabiri and Heaslip [20]

$$b_i = \tan^{-1} \frac{\text{Vinc}(\text{lat}_i, \text{lng}_i, \text{lat}_{i+1}, \text{lng}_i)}{\text{Vinc}(\text{lat}_i, \text{lng}_i, \text{lat}_i, \text{lng}_{i+1})},\ 1 \le i < N; \tag{1e}$$

$$b_N = b_{n-1}; \tag{1f}$$

After this representation pre-processing, each GPS tripleg is interpreted with four aligned time sequence motion and displacement attributes of length N. These data are considered sufficiently descriptive to the motion reflected by the GPS points, and further data processing and learning are conducted based on them.

[1] Data representation processing is also widely named as "pre-processing" in the literature. We refer to these two terms interchangeably in the sequel.

3.3. Data Feature Computation

The previous GPS record pre-processing step segments GPS records into triplegs, which are then interpreted by temporally aligned motion and displacement attribute vectors. While fundamental domain-specific knowledge of transport is integrated, one may further refer to signal processing techniques to extract data characteristics within the attribute data. This is especially for travel mode identification as frequency-domain mapping of tripleg attributes are discriminative with respect to various travel modes, since the signals are generally repetitive, and specific signal patterns can be established [11, 13].

In this work, two widely-recognized signal decomposition techniques, i.e., discrete Fourier transform (DFT) and discrete wavelet transform (DWT), are adopted to further extract frequency-domain data features of the tripleg attribute vectors. By DFT, tripleg attributes are interpreted with discrete frequency components:

$$\mathcal{F}(x_i) = \sum_{k=1}^{N} x_k \cdot e^{-j2\pi ki/N}, \tag{2}$$

where x_i refers to the input signals, i.e., s_i, a_i, k_i, and r_i in our case. According to (2), a length-N vector is transformed into another length-N vector. Despite that subsequent DNN structure can be designed to accept data series, the latent data correlation among data points in $\mathcal{F}(x_i)$ is not as strong as in the original attributes. As demonstrated in previous literature, see [27–29] for some examples, statistical features of the spectrum can still summarize the latent information required by DNN. Therefore, we select the following data features of the frequency-domain representation of tripleg attributes as DFT data features [21]: 1) spectral centroid, 2) spectral spread, 3) spectral flatness, 4) spectral roll-off, 5) spectral crest, and 6) spectral kurtosis. Consequently, 4 attributes $\times 6$ statistics $= 24$ DFT data features can be developed from the four GPS tripleg attribute vectors. We use set $\mathfrak{F}$ to represent these features.

Additionally, DWT is also incorporated to further construct data features. The main purpose of adopting DWT is to overcome the drawback of DFT, which develops an unstable spectrum when handling temporally non-stationary signals [30]. Wavelet transform employs discrete wavelets $\psi_{a,b}(t)$ to convolve input signals. With pre-defined mother wavelets $\psi(t)$, input signal $x(t)$ can be interpreted by

$$d_{a,b} = \int_{-\infty}^{+\infty} x(t)\psi^*_{a,b}(t)dt = \langle x(t), \psi_{a,b}(t)\rangle, \tag{3a}$$

where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{2^a}}\psi\left(\frac{t}{2^a} - b\right),\ a, b \in \mathbb{Z}, \tag{3b}$$

$\psi^*_{a,b}(t)$ is the complex conjugate of $\psi_{a,b}(t)$, a and b are scaling parameters of dilation (oscillatory frequency) and translation (shifted position) of the discrete wavelet, respectively.

Nonetheless, (3a) is in most cases intractable for continuous or discrete signals $x(t)$ [29, 31]. To handle this problem, Mallet proposed multi-resolution analysis to decompose signals with successive approximation subspaces so that features that are difficult to be observed in an arbitrary resolution can be discovered in other resolutions [31]. While there are other approaches to interpret wavelet transform, multi-resolution analysis is widely considered as a standard DWT method. For a signal $x(t)$ defined in zero-scale space V_0, it is decomposed into approximation subspaces $\{V_i, i \in \mathbb{Z}\}$ so that the orthogonal basis of V_0 is obtained by

$$V_{i-1} = V_i \times W_i \text{ and } W_i \perp W_k, \forall i \neq k, \tag{4}$$

where $\{W_i, i \in \mathbb{Z}\}$ is the orthogonal complement of V_i in V_{i-1}. Following this subspace decomposition process, $x(t)$ is interpreted by [31]

$$x(t) = \sum_{b} \langle x(t), \phi_{M,b}(t)\rangle \cdot 2^{-M/2}\phi(2^{-M}t - b) \tag{5a}$$

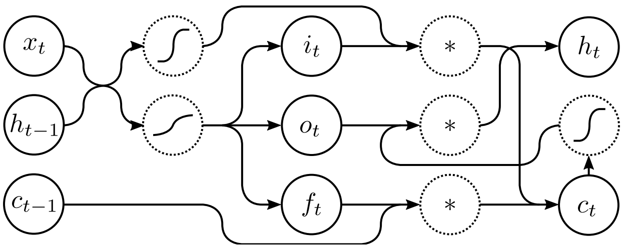


Figure 2: Data flow of LSTM propagation rule (6).

$$+\sum_{a}^{M}\sum_{b}\langle x(t), \psi_{a,b}(t)\rangle \cdot 2^{-M/2}\psi(2^{-a}t-b) \tag{5b}$$

given a wavelet $\psi_{a,b}(t)$, where $M \leq \lfloor \log_2 N \rfloor$ is the number of decomposition levels, and $\phi_{M,b}(t)$ is a companion scaling function [31]. In this process, each pair of approximation coefficient $a_{M,b} = \langle x(t), \phi_{M,b}(t)\rangle$ and detail coefficients $d_{a,b} = \langle x(t), \psi_{a,b}(t)\rangle$ represents a bandpass-filtered signal of the original $x(t)$. The decomposition can be iterated with a_{M_b} being further decomposed in subsequent iterations.

In GPS trajectory data analysis, we are generally more interested in the trend of trajectory attributes than the details [13]. Therefore, only the approximation coefficients are employed to construct DWT data features. Additionally, we follow the discussion in Yu *et al.* [27] and Chen *et al.* [28] and use daubechies (`db`) and symlets (`sym`) mother wavelet families to decompose tripleg attributes. Both families are favored thanks to their robustness to heterogeneous data properties such as signal length and sample size when sufficient samples are available, which is the case for trajectory analysis. Consequently, eight wavelets, i.e., `db1` to `db4` and `sym1` to `db4`, are used to extract the temporal-frequency domain feature of tripleg attribute vectors. Similar to DFT, we also employ a set of statistical features of the respective approximation signals to represent the tripleg attributes: 1) maximum, 2) minimum, 3) mean, 4) standard deviation, 5) skewness, 6) kurtosis, 7) energy, and 8) entropy. As a result, 4 attributes $\times$ 8 wavelets $\times$ 8 statistics $=$ 256 DWT data features can be calculated, which are denoted by $\mathcal{W}$.

3.4. Latent Information Extraction

In addition to the DFT and DWT feature data, the original tripleg attribute vectors are among the data widely employed in travel mode identification task in the previous literature, see [13, 14, 32–34] for some examples. While it is unclear which set of data is the most discriminative in the identification, it is possible to employ ensemble learning technique to agglomerate all information for final decision making [22]. In this work, we adopt a stacked LSTM DNN with jumping links to extract temporally correlated latent information from attribute vectors.

There are two major components in the neural network, namely, LSTM and jumping links. LSTM by Hochreiter and Schmidhuber [35] is a modern architecture of neural networks, which is among the most commonly adopted data mining techniques. While canonical neural networks disdain temporal-correlation with in the training data, LSTM learns from such correlation to extract latent information thanks to the unique design of states that propagates over time. Given a time-series $\mathbf{x} = \{x_i\}_{1\leq i\leq N} \in \mathbb{R}^{F\times N}$ where F is the number of features in x_i, LSTM maps the input data into an R-dimensional time-series $\mathbf{h} = \{h_i\}_{1\leq i\leq N} \in \mathbb{R}^{R\times N}$ by the following propagation rules[2]:

$$h_t = o_t * \tanh(c_t), \tag{6a}$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(\mathbf{w}_c \cdot [h_{t-1}, x_t] + b_c), \tag{6b}$$

$$f_t = \sigma(\mathbf{w}_f \cdot [h_{t-1}, x_t] + b_f), \tag{6c}$$

[2] We use bold symbols to denote multi-dimensional matrices or tensors in the sequel, and italic symbols are one-dimensional vectors.

$$i_t = \sigma(\mathbf{w}_i \cdot [h_{t-1}, x_t] + b_i), \tag{6d}$$

$$o_t = \sigma(\mathbf{w}_o \cdot [h_{t-1}, x_t] + b_o), \tag{6e}$$

where f_t, i_t, and o_t are the activations of forget, input, and output gates, which controls the information retention of previous network state and new input data, as well as the strength of current network state in the output data of the corresponding time step, respectively [22, 35, 36]. Operator $*$ defines Hadamard (element-wise) product, $\sigma(\cdot)$ is the sigmoid function, $\mathbf{w} \in \mathbb{R}^{R\times F}$ matrices are trainable weight parameters, and $b \in \mathbb{R}^R$ vectors are trainable bias parameters. Figure 2 demonstrates the data flow within LSTM. From the figure and (6) it is clear that both the previous LSTM state c_{t-1} and previous output data h_{t-1} are incorporated when calculating the t-th time step output. With this latent information flow design, temporal correlation can be extracted from the input data, which is among the key factors of identifying travel modes [11, 34, 37]. To form a DNN for extracting latent information, six layers of LSTM are stacked. The first and last layers have 128 output features in each layer, and the remaining four have 256. Note that the first layer takes all tripleg attribute vectors as input data, rendering the number of input features to be four as illustrated in (1).

A second key component in the neural network is the jumping links inspired by residual networks [38], which are "bypassing" data flow channels over each of the LSTM layer. Jumping links are included in order to resolve the accuracy saturation problem of common DNN, and they can greatly help optimization techniques to fine-tune network parameters as the solution space are smoothened by the additional data flow channel [38]. The propagation rule of jumping links is formulated as follows:

$$\mathcal{L} = \mathrm{ReLU}\left(h_N^{(6)} + \sum_{l=1}^{5} \mathbf{w}^{(l)} h_N^{(l)} + \mathbf{w}^{(0)} x_N\right), \tag{7}$$

where $\mathcal{L}$ is the LSTM data features extracted, $\mathrm{ReLU}(x) = \max(x, 0)$ is the rectified linear unit [39], $h_N^{(l)}$ is the last feature set of $\mathbf{h}^{(l)}$ output by the l-th LSTM layer, and $\mathbf{w}^{(l)}$ is the jumping link trainable weight parameters for the l-th LSTM layer. The introduction of jumping links effectively improve the system performance as will be demonstrated in Section 5.7.

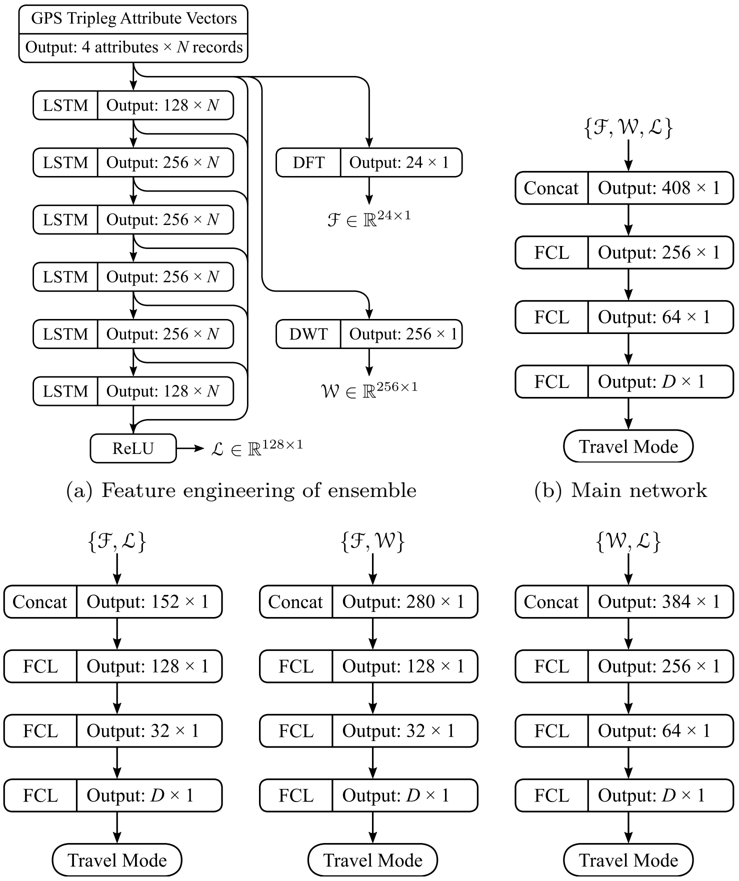
4. Semi-supervised Learning with Ensemble

With data features $\mathcal{F}$, $\mathcal{W}$, and $\mathcal{L}$ extracted by data processing techniques, it is possible to construct a fully-connected neural network [22] that takes all features as input and produces a tag indicating the travel mode. The network can be easily trained end-to-end using categorical cross-entropy loss, and it is a typical solution when trajectory data are all labeled by their respective travel mode information, see [5, 9, 13] for some examples. Nonetheless, this solution cannot handle practical cases in which most data are indeed unlabeled. In this work, we propose a semi-supervised learning scheme for travel mode identification with most trajectories unlabeled. The proposed training method is an implementation of the multi-view training with proxy label methodology for semi-supervised learning [40, 41].

4.1. Ensemble Configuration

Multi-view training refers to a type of semi-supervised learning methods which aim to train different learning models with different "views" of the original data [42]. The views shall ideally complement each other, leading the corresponding models to collaborate in improving system performance over individual models. Typical views are constructed by using different data features and model architectures [42]. By accepting combinations of $\{\mathcal{F}, \mathcal{W}, \mathcal{L}\}$, multiple DNNs are formulated to establish an ensemble learning scheme for travel mode identification.

Fig. 3 presents the architecture of the constructed DNNs. In particular, three fully connected layers [22] are appended to a concatenation operation, which takes $\{\mathcal{F}, \mathcal{W}, \mathcal{L}\}$, $\{\mathcal{F}, \mathcal{L}\}$, $\{\mathcal{F}, \mathcal{W}\}$, and $\{\mathcal{W}, \mathcal{L}\}$ as input features for the four networks, respectively. While the input features are developed using the same techniques as illustrated in the previous section, each ensemble network exploits the latent information with



(a) Feature engineering of ensemble

(b) Main network

(c) View network 1

(d) View network 2

(e) View network 3

Figure 3: Architecture of the proposed ensemble. “Concat” mean “Concatenation layer” and “FCL” means “Fully-connected layer”.

Algorithm 1: Semi-supervised Training

```
Data: $\mathcal{D}^*$, $\mathcal{D}^0$, and $\mathcal{M}^*$
Result: Trained parameters of $\mathcal{N}_i$, $i \in \{0..3\}$
$\mathcal{S} \leftarrow \langle \mathcal{D}^*, \mathcal{M}^* \rangle$ repeat
    for $i \in \{0..3\}$ do
        train $\mathcal{N}_i$ with $\mathcal{S}$;
    end
    for $x \in \mathcal{D}^0$ do
        for $i \in \{0..3\}$ do
            $m_i \leftarrow \mathcal{N}_i(x)$;
        end
        for $d \in \{1..D\}$ do
            if $m_1 = m_2 = m_3 = d$ or $m_0 = m_i = d, \exists\, i \in \{1..3\}$ then
                $\mathcal{S} \leftarrow \mathcal{S} \cup \langle x, d \rangle$;
                $\mathcal{D}^0 \leftarrow \mathcal{D}^0 \setminus \{x\}$
            end
        end
    end
until $m_i$, $i \in \{0..3\}$ does not change for all $p \in \mathcal{D}^0$;
```

different hyperparameter configurations. For networks with a number of features greater than 300 after concatenation, 256 and 64 neurons are employed in the first two ReLU-activated layers of the respective fully connected networks. Otherwise, the number of neurons is halved in these layers. Finally, a third softmax-activated layer of D neurons are employed to infer the final travel mode index, where D is the total number of travel modes possible.

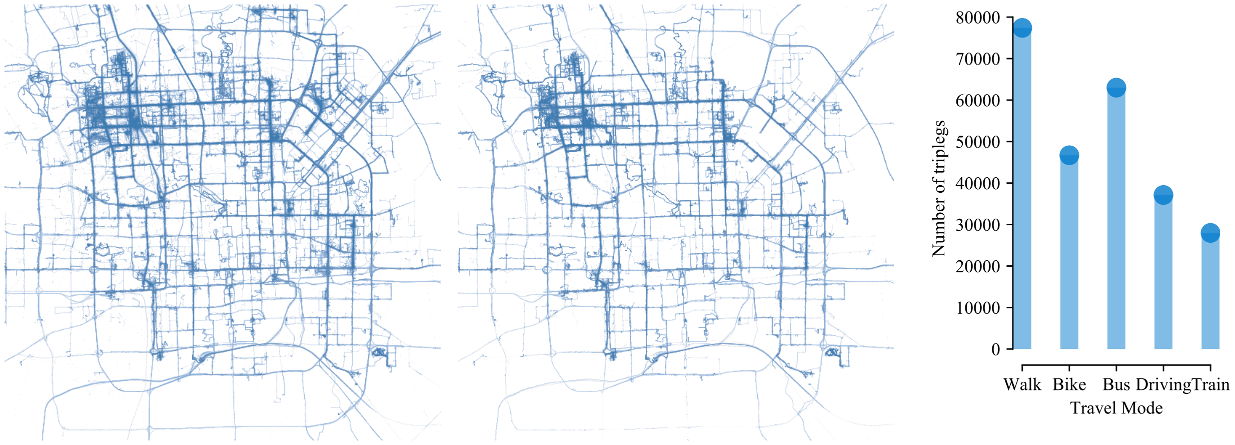
In order to distinguish the four DNNs, we name the one that takes all data features "main" network ($\mathcal{N}_0$), and the remaining ones "view" networks ($\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$). Intuitively speaking, the inference made by the main network shall be more reliable than the others as more information is provided as input. This characteristic is the basis of training the ensemble in a semi-supervised manner, which will be introduced next.

4.2. Semi-supervised Training

In many real-world trajectory data sets, e.g., GeoLife [9], only a part of the GPS records is labeled by the respective travel modes. Let $\mathcal{D}$ be the complete set of triplegs segmented by the approach introduced in Section 3.2, among which the ones that have travel mode labels are denoted by subset $\mathcal{D}^* \subsetneq \mathcal{D}$, and $\mathcal{D}^0 = \mathcal{D} \setminus \mathcal{D}^*$. Symbol $\mathcal{M}^*$ represents the respective travel modes of $\mathcal{D}^*$. The proposed semi-supervised learning scheme trains the four ensemble networks iteratively by progressively including triplegs of $\mathcal{D}^0$ in $\mathcal{D}^*$ with asserted travel mode developed by the ensemble. Specifically, an initial training set $\mathcal{S} \leftarrow \langle \mathcal{D}^*, \mathcal{M}^* \rangle$ is constructed first, which is adopted to train the four DNNs end-to-end. Categorical cross-entropy is employed as the loss function, and Adam optimizer by Kingma and Ba [43] is used to fine-tune network parameters. These models then make predictions on $\mathcal{D}^0$ to develop m_i, $i \in \{0..3\}$ for any arbitrary tripleg p with $\mathcal{N}_i$, respectively. Subsequently, p is included in $\mathcal{S}$ if and only if one of the following holds:

1. All three view networks infer the same travel mode, i.e., $m_1 = m_2 = m_3$ holds, or
2. One of the view networks infers the same travel mode with the main network, i.e., $m_0 = m_i$, $\exists\, i \in \{1..3\}$.

This ends one iteration of the training. The whole semi-supervised training terminates until none of the predictions on $\mathcal{D}^0$ changes. Algorithm 1 presents the pseudo-code of the training scheme.



(a) Original distribution of trajectories in urban Beijing region.

(b) Distribution of triplegs after pre-processing.

(c) Histogram of travel modes.

Figure 4: Distribution of GeoLife GPS points, triplegs, and travel modes.

The online inference of new triplegs is similar to the training process. If the three view networks generate the same result, this travel mode is considered as the true mode corresponding to the tripleg. Otherwise, the result developed by the main network is final.

5. Case Studies

In this work, we propose a semi-supervised deep ensemble learning scheme for travel mode identification. In order to investigate its performance, we conduct a series of comprehensive case studies. Specifically, we first compare the proposed scheme with previous travel mode identifiers in the literature, as well as other machine learning techniques. Additionally, the impact of measurement uncertainty is evaluated via empirical studies. Subsequently, we investigate the contribution of data features, i.e., $\mathcal{F}$, $\mathcal{W}$, and $\mathcal{L}$, on the system performance. Finally, a hyperparameter test is carried out to illustrate how the DNN architecture should be formed to achieve the best overall performance.

5.1. Data Set and Settings

In subsequent case studies, we employ the raw trajectory data from GeoLife project [9, 19] for investigation, which presents the movement trajectory of 182 users in a period of over five years. In the data set, each trajectory is presented in the form of a GPS record sequence, and 69 users among all also provide the travel mode of each respective trajectory. We consider these user-tagged information as ground truth and roughly follow the previous work by Shang *et al.* [44] and Yu *et al.* [13] to pre-process the raw data and assign travel modes. In particular, five ground-based travel modes are considered, i.e., walk, bike, bus, driving, and train[3]. Then the 8120 trajectories of these five modes with more than 20 GPS records are further segmented into smaller triplegs. Specifically, all trajectories with more than 20 GPS records are divided into triplegs, each of which has 20 records. Then the last two triplegs are merged into one so that all of them will have at least 20 records. The incentive behind this segmentation is that these trajectories can sometimes constitute thousands of GPS records, which is far beyond enough for identification. As a result, the data set is augmented to possess 252 190 triplegs. All subsequent case studies are conducted on these triplegs Fig. 4 presents the distribution of triplegs and their travel modes.

[3] In GeoLife, eleven travel modes are tagged in total. In accordance with the literature [13, 44], we jointly consider taxi and car as driving, and other transportation modes are discarded, namely, airplane, boat, motorcycle, run, and subway

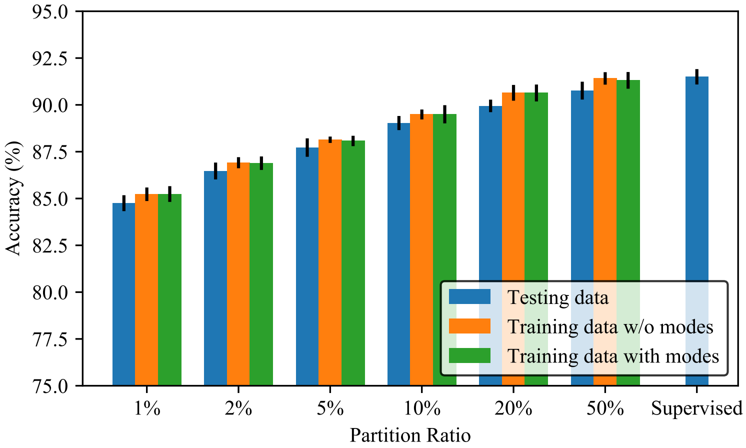


Figure 5: Travel mode identification accuracy of the proposed scheme.

To adjust the network parameters of the proposed DNN with semi-supervised learning, we randomly partition all 252 190 triplegs into two data sets by the ratio of 3 : 1. The first data set is considered as the training set $\mathcal{D}$ whose travel mode tags are used to train the network, and the other data set is called testing set, whose travel mode tags are employed to evaluate the identification accuracy after the respective DNN is trained. This configuration is adopted for cross-validation and over-fitness detection, and accords with the common practice, see [13, 45] for examples. Furthermore, in order to emulated practical scenarios in which only a portion of trajectories have their travel mode tagged, we consider six training data partition plans with different "tag rates" α where 1%, 2%, 5%, 10%, 20%, and 50% of all triplegs in the training set are included in the initial $\mathcal{D}^*$. The remaining ones, whose travel modes are discarded during training, constitutes $\mathcal{D}^0$. In addition, ten random partitions for each tag rate are established for statistical significance, rendering 60 independent $\mathcal{D}^*$ sets. Last but not least, we also test the performance of the proposed identifier under supervised learning scenarios by employing the whole training set for parameter tuning. The performance of the proposed travel mode identifers is evaluated by classification accuracy, precision, and recall metrics.

All case studies are conducted on an nVidia DGX-2 enterprise AI research system equipped with nVidia Tesla V100 GPUs for parallel computing acceleration. PyTorch [46] is employed in neural network construction for computation acceleration.

5.2. Identification Accuracy

We first assess the accuracy of travel mode identification. Totally 70 individual neural networks with the architecture depicted in Fig. 3 are constructed and trained. Among them, the first 60 networks are trained in a semi-supervised manner with the respective $\mathcal{D}^*$ sets created previously according to Algorithm 1, and the last ten are individual models trained with the same complete set $\mathcal{D}$ which demonstrates the performance of the network under supervised learning. The simulation results are summarized in Fig. 5. In the figure, the identification accuracy with respect to the testing set, $\mathcal{D}^0$ (labeled by "training data w/o modes" in the figure), $\mathcal{D}^*$ (labeled by "training data with modes" in the figure) are presented. Additionally, the last supervise-trained neural network produces an accuracy performance on the testing data, which is presented in the last bar plot of the figure.

The simulation results indicate that the proposed semi-supervised learning scheme develops satisfactory travel mode identification results. While there is slight performance degradation when less tagged samples are employed in training, i.e., reduced rag rate, the accuracy with only 1% of training data employed can still achieve approximately 85%. Furthermore, the identification accuracies of training data with and without travel mode tags are similar to their respective testing data performance. The result demonstrates that the neural network does not suffer from over-fitting issue notably. This is greatly contributed by the jumping link design of the proposed architecture.

Table 1: Confusion Matrices of the Proposed Travel Mode Identification Scheme with Selected Tag Rates

		$\alpha = 1\%$					
		Predicted Mode					Recall (%)
		Walk	Bike	Bus	Car	Train	
Real Mode	Walk	17241	1127	487	307	102	89.5
	Bike	1027	10021	331	232	83	85.7
	Bus	171	575	13308	1337	435	84.1
	Driving	127	367	917	7380	310	81.1
	Train	114	263	469	595	5721	79.9
Precision (%)		92.3	81.1	85.8	74.9	86.0	**85.1**
		$\alpha = 5\%$					
		Predicted Mode					Recall (%)
		Walk	Bike	Bus	Car	Train	
Real Mode	Walk	18012	698	301	190	63	93.5
	Bike	812	10372	262	183	65	88.7
	Bus	149	503	13625	1168	380	86.1
	Driving	114	328	820	7562	278	83.1
	Train	97	224	399	506	5936	82.9
Precision (%)		93.9	85.5	88.4	78.7	88.3	**88.0**
		$\alpha = 20\%$					
		Predicted Mode					Recall (%)
		Walk	Bike	Bus	Car	Train	
Real Mode	Walk	18016	698	301	190	63	93.5
	Bike	668	10606	215	151	54	90.7
	Bus	121	409	14036	950	309	88.7
	Driving	84	242	606	7962	205	87.5
	Train	75	171	306	387	6222	86.9
Precision (%)		95.0	87.5	90.8	82.6	90.8	**90.2**
		Supervised Training					
		Predicted Mode					Recall (%)
		Walk	Bike	Bus	Car	Train	
Real Mode	Walk	18323	526	227	143	48	95.1
	Bike	610	10700	197	138	49	91.5
	Bus	109	365	14226	849	276	89.9
	Driving	73	210	524	8117	177	89.2
	Train	64	146	261	331	6358	88.8
Precision (%)		95.5	89.6	92.2	84.6	92.0	**91.6**

Besides the statistical results summarized in Fig. 5, we also take a closer look into the confusion matrices of selected tag rates in Table 1. Specifically, the medium-performing results of $\alpha \in \{1\%, 5\%, 20\%\}$ and fully-supervised training are presented. The bolded value denotes the overall identification accuracy of all testing samples. The confusion matrices illustrate similar observations with previous work [13, 20]: walking is generally more precisely identified due to the larger number of training samples, while walk-bike and bus-driving are two pairs of modes that are relatively easily misclassified due to their similar motion and displacement characteristics.

Furthermore, we are in particular interested in the test cases where the proposed approach failed to develop the correct travel mode. They can provide guidelines to further improve the quality of future travel mode identifiers. The following observations can be developed from the mis-identified triplegs:

- Walking is mostly mis-identified as cycling. A majority of these triplegs involves fast-pace walking (greater than $6\,\mathrm{km\,h^{-1}}$) along a straight street. In this work, we take triplegs of 20 GPS records as the input. Increasing this value may lead to more trajectory characteristics for more accurate identification.

- Besides walking, cycling is sometimes classified as bus-driving. This happens most when the bike starts at traffic lights, which shares very similar speed, acceleration, and turn rate properties with bus-driving. Besides increasing length of the tripleg, we are yet to find an effective way to correct the problem.

- Bus and car are a pair of commonly mis-identified modes due to their relatively similar movement pattern. This issue can be potentially resolved by inspecting the stopping time between triplegs. In addition, external lane changing information is another possible source of data that may contribute to the classification.

- Train is sometimes regarded as car and bus during the gradual acceleration/deceleration process (speed around $60\,\mathrm{km\,h^{-1}}$). This issue can be address if transportation network information – especially road and rail networks – can be incorporated.

- There is no significant difference in terms of mis-identification patterns among different levels of tag rates.

Last but not least, the computation time required for training the ensemble and inferring travel modes are recorded. The training is conducted in parallel mode, i.e., line 2–4 of Algorithm 1 is performed in parallel on different GPUs. When one Tesla V100 is allocated to each ensemble network, the whole training can be finished within 7.4 h (supervised training with all tags) to 9.1 h ($\alpha = 1\%$). As neural network training process is typically conducted offline [22], the above training time can be considered acceptable given that well-tuned networks can be adopted to handle unknown trajectories without further online training. Additionally, the average travel mode identification time for the 63 047 testing triplegs is approximately 8.5 ms. The fast online inference time indicates that the proposed scheme can be applied in online travel mode identification cases where real-time classification results are readily available.

5.3. Performance on Heavily Congested Urban Region

In addition to the analysis on general trajectories as presented in Section 5.2, we are in particular interested in the system performance when trajectories in heavily congested urban areas are being identified. These trajectories, though generated by different travel modes, may look similar due to the overcrowded traffic condition, rendering mode identification a challenging task. To test the proposed approach in handling such scenarios, we construct a new subset of GeoLife called GeoLife-Z (Fig. 6) by selecting its trajectories that fall into the region of Zhongguan Cun in Beijing, which is also known as “Silicon valley of China” and is among the most congested regions in Beijing. The new dataset is used to test the identification accuracy of previously trained models in Section 5.2 that are trained using GeoLife dataset. In addition, we also train new models with only GeoLife-Z trajectories and investigate if the training data size and characteristics

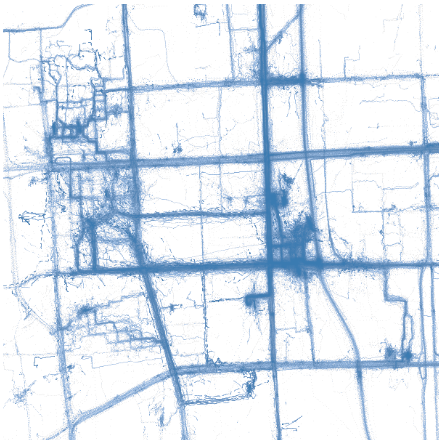


Figure 6: Trajectories in the new GeoLife-Z dataset

Table 2: Performance on Heavily Congested Region

Training	Testing	Accuracy (%) ± Standard Deviation (%)						
		$\alpha = 1\%$	$\alpha = 2\%$	$\alpha = 5\%$	$\alpha = 10\%$	$\alpha = 20\%$	$\alpha = 50\%$	Supervised
GeoLife	GeoLife	84.8±0.42	86.5±0.45	87.7±0.44	89.0±0.38	90.0±0.41	90.8±0.48	91.5±0.41
GeoLife	GeoLife-Z	84.5±0.40	85.7±0.46	87.2±0.40	88.7±0.37	89.2±0.44	89.9±0.41	91.4±0.42
GeoLife-Z	GeoLife-Z	86.3±0.41	88.0±0.36	89.5±0.37	89.5±0.40	91.0±0.38	91.7±0.36	92.7±0.42

influence the performance. All other simulation configurations are kept identical to those in Section 5.2 and the results are presented in Table 2.

From the simulation results it is clear that the proposed approach works well for both general urban regions and heavily congested regions. When directly applying previous trained models to test trajectories in GeoLife-Z, the identification accuracy is hardly undermined: a 0.1% to 0.8% accuracy decrease can be observed across different tag rates. This is mainly contributed by the temporal-frequency domain feature extraction capability of the proposed solution. For instance, while the trajectories for driving and walking may share a similar spatial pattern in congested areas, introducing a temporal axis to the 2-D trajectory space is sufficient to distinguish these two types of travel modes. While driving in such area also leads to slow average driving speed, vehicles are typically switching between accelerating-decelerating during the course, which is different from a regular pedestrian. Furthermore, the performance on such scenarios can be further improved by adopting specialized training dataset. When training the same deep learning models using GeoLife-Z dataset, the identification accuracy even surpasses that of the original combination. This is because that the small GeoLife-Z dataset does not include travel mode “train”, rendering a simplier classification problem. In addition, such a more specialized dataset may potentially better reveal the distinguishing factors among different modes in congested areas. To conclude, the proposed scheme works satisfactorily in both general and heavily congested regions.

5.4. Comparison with Other Approaches

Besides investigating the performance of the proposed scheme, it is also of interest to compare the results with other approaches presented in the literature. Since the proposed scheme can work under both supervised learning and semi-supervised learning scenarios, we include the travel mode identification results of approaches in both classes in the comparison. Specifically, we implement a series of baseline approaches and compare their identification accuracy on various tag rates, including semi-supervised convolutional autoencoder (SECA) [7], semi-two-steps [7], semi-pseudo-label [7], and generative adversarial network (GAN) [25]. Additionally, classical machine learning techniques are also adopted in the comparison, including

Table 3: Comparison of Identification Accuracy

Approach	Accuracy (%) ± Standard Deviation (%)						
	$\alpha = 1\%$	$\alpha = 2\%$	$\alpha = 5\%$	$\alpha = 10\%$	$\alpha = 20\%$	$\alpha = 50\%$	Supervised
Proposed	**84.8**±0.42	**86.5**±0.45	**87.7**±0.44	**89.0**±0.38	**90.0**±0.41	**90.8**±0.48	**91.5**±0.41
SECA [7]	52.0	54.5	56.1	62.3(*62.9*)	71.6(*69.3*)	72.9(*73.2*)	77.2(*76.8*)
Semi-two-step [7]	50.7	51.4	53.0	50.6(*54.4*)	54.4(*56.2*)	57.7(*58.8*)	59.1(*60.5*)
Semi-pseudo-label [7]	50.9	53.6	56.0	61.8(*58.9*)	68.6(*66.3*)	72.5(*70.7*)	74.9(*75.4*)
GAN [25]	68.4	74.1	77.7	80.5	82.1	83.1	83.8
k-NN	42.1	44.0	51.6	46.9	50.8	54.9	57.9
SVM	51.7	53.3	55.3	41.7	46.0	47.0	53.2
DT	62.1	65.8	66.9	66.1	67.2	67.8	69.4
MLP	38.2	37.9	39.0	27.4	30.9	33.1	35.4
CNN [20]	56.3	58.8	61.1	70.5	74.9	83.6	84.3(*84.6*)
DT-heuristic [19]	52.6	51.8	55.5	63.0	68.7	68.6	74.4(*76.2*)
Image-DNN [23]	45.6	49.2	50.5	55.3	60.3	67.8	67.2(*67.9*)
RF	50.1	57.7	59.0	64.9	74.4	71.8	77.6(*78.1*)
LSTM [21]	49.8	50.7	58.2	60.9	70.1	73.5	81.7(***96.8***)
CNN ensemble [16]	62.0	60.7	64.4	73.7	81.9	83.9	90.6(*91.8*)

decision tree (DT)-based heuristic [19], convolution neural network (CNN) [20], image-based DNN [23], k-NN, support vector machine (SVM), multi-layer perceptron (MLP), random forest (RF) [20], LSTM [21], and CNN ensemble [16]. While the second class of approaches are not designed for semi-supervised learning by nature, we only use labeled data to train these models and investigate their performance with the same testing set data. The simulation results are summarized in Table 3. In this table, all values in roman font are developed by our implementations, and those whose accuracy values are provided by the original literature are also presented in italic for reference. The best performing values are presented in bold, and the standard deviation of the proposed approach is also demonstrated.

From the results, a clear conclusion can be drawn that the proposed scheme can significantly outperform all compared semi-supervised travel mode identifiers in the literature. The second best-performing approaches, i.e., SECA, still scores more than 17% less accuracy than the proposed one. This can be credited to the different design principles of these two approaches. While the auto-encoder design of SECA can exploit the low-dimensional representation of trajectories, frequency-domain features greatly help the proposed scheme achieve better identification results [21]. Additionally, the proposed scheme is also competitive when all travel mode information is available for training, i.e., in supervised learning scenarios. In the comparison, it scores the third (first if only consider the simulations conducted by us) among all approaches. While the best performing LSTM by Güvensan and Asci [21] significantly outperforms the others, it adopts a different data set (HTC data set [47]) whose characteristics may not be similar to those of GeoLife, and the performance significantly deteriorates with GeoLife. The results also demonstrate the efficacy of the adopted ensemble design, as ensemble-based approaches provide 6.3% accuracy improvement based on others (CNN ensemble versus typical CNN). We will further investigate its importance in subsequent tests.

5.5. GPS Accuracy

GPS signals are broadcast by satellites in space with a certain accuracy. However, GPS receivers capture the signals subjected to additional influencing factors, e.g., satellite geometry, atmospheric conditions, and signal blockage, etc [48]. The consequent inaccuracy GPS records may potentially adversely impact the performance of travel mode identifiers. Therefore, we investigate the influence of GPS accuracy on the proposed scheme with a preliminary case study. In accordance with the GPS horizontal accuracy statistics recorded in [49], all GeoLife GPS records are distorted with Rayleigh distribution with a probability density function $f(x;\sigma) = \frac{x}{\sigma}e^{-x^2/2\sigma^2}$. We re-train the proposed scheme with case study parameters $\{\alpha \times \sigma\} \in \{1\%, 10\%, 50\%, 100\%\} \times \{0.2, 0.5, 1.0, 2.0, 5.0\}$. The scale of Rayleigh distribution, i.e., σ, determines the

Table 4: Influence of GPS Accuracy

Rayleigh	Accuracy (%)			
Scale (σ)	$\alpha = 1\%$	$\alpha = 10\%$	$\alpha = 50\%$	Supervised
0.2	84.6	88.9	90.8	91.5
0.5	84.6	89.0	90.8	91.4
1.0	84.5	88.7	90.9	91.3
2.0	84.1	88.3	90.6	91.1
5.0	82.8	86.2	88.0	89.1

Table 5: Feature Sets and Their Performance

Set	Attributes		DFT Features		DWT Features		Accuracy (%)			
	s_i,a_i	k_i,u_i	Less	Full	Less	Full	$\alpha = 1\%$	$\alpha = 10\%$	$\alpha = 50\%$	Supervised
Proposed	✓	✓		✓		✓	**84.8**	**89.0**	**90.8**	**91.5**
A	✓	✓	✓			✓	82.3	86.6	89.1	90.9
B	✓	✓		✓	✓		83.7	88.1	90.7	**91.5**
C	✓	✓	✓		✓		81.3	86.3	89.0	90.5
D	✓			✓		✓	81.7	85.6	87.1	88.0
E	✓		✓			✓	80.6	85.2	83.5	84.9
F	✓			✓	✓		80.2	85.5	86.8	87.1
G	✓		✓		✓		80.5	84.5	83.5	84.8

overall accuracy of GPS records, where a larger scale denotes a larger GPS error. The statistics depicted in [49] shows an approximately $\sigma = 0.7$ Rayleigh distribution.

The identification accuracy results are summarized in Table 4. From the simulation results, we can observe that GPS record error does not have a notable influence on the travel mode identification results when the accuracy is above or around the standard level ($\sigma \leq 1.0$). In the meantime, the performance degradation is noticeable for significantly large GPS error, where the identification accuracy is decreased by approximately 2%. Nonetheless, when comparing with other approaches as presented in Table 3, the proposed scheme still provides solid advantages. Additionally, the results indicate that the degradation does not show a clear correlation to the volume of available travel mode data (α). To conclude, the proposed scheme can achieve almost the same performance considering mild and practical GPS measurement uncertainties, and can still provide satisfactory identification results comparing with others.

5.6. GPS Data Features

In Section 3 we discussed a data representation processing and two feature engineering techniques, each of which develops several unique features of input GPS triplegs. While these features are the only input to the ensemble, we are interested in their contribution to the performance of the proposed scheme. In this case study, selected features are removed from the input to see if a comparable identification accuracy can be obtained without these features. Specifically, we only utilize two of the four motion and displacement attributes in Section 3.2, namely, speed and acceleration. Furthermore, we also consider only the spectral centroid, spread, and flatness of DFT data and maximum, minimum, mean, and standard deviation of DWT approximation coefficients in the test. These input feature sets are called “Less DFT/DWT features” in contrast to the original “Full” inputs. New neural network models are constructed adopting the new number of inputs in the first layer, and the same training and testing approach as elaborated in Section 4 is employed.

The new input feature sets and their respective identification performance are demonstrated in Table 5. A straightforward observation from the comparison is that all features contribute to the identification, with a clear emphasis on the four motion and displacement attributes. This can be developed by comparing the accuracy of Sets A–C with that of Sets D–G. The fundamental reason is that these attributes also serve as the basics of both DFT and DWT according to Fig. 1, thus removing the jerk and turn rate notably

Table 6: Performance of Other Training Approaches with and without Jumping Link

Approach	View Netw.	Accuracy (%)	
		$\alpha = 1\%$	$\alpha = 10\%$
Proposed	1,2,3	**84.8**	**89.0**
Proposed w/o jumping link	1,2,3	77.9	81.5
Co-training [40]	1	80.3	85.8
	2	79.9	85.6
	3	78.8	83.4
Tri-training [50]	1,2	83.5	86.4
	2,3	82.9	84.8
	1,3	83.5	85.5
Tri-training w. Disagreement [51]	1,2	84.0	86.2
	2,3	83.1	85.6
	1,3	82.7	86.0
Tri-training w. Disagreement [51] w/o jumping link	1,2	78.2	80.7
	2,3	77.1	79.2
	1,3	76.0	80.0

reduces the dimensionality and data characteristics of input data. Additionally, the result implies a higher weighting on the DFT features than those of DWT. Sets B and F generally performs better than Sets A and E, respectively, and the difference between them is the removal of either partial DFT or DWT features. To conclude, all data features help the system to identify travel modes. The motion and displacement attributes are the most critical ones, followed by DFT and DWT features.

5.7. Jumping Links and Ensemble Training Method

In this work, we propose four neural networks to construct an ensemble and develop a semi-supervised learning scheme to train them. Each of the neural networks is supported by a jumping-link-enabled feture engineering neural network backbone. In this sub-section, we test the efficacy of the proposed semi-supervised training approach for the travel mode identifier and investigate the effectiveness of the jumping link. While multi-view training inspires the design of proposed training approach, it also has other widely adopted training approaches, e.g., Co-training [40], Tri-training variants [50, 51], etc. Specifically, we adopt the main network and one of the view networks in Fig. 3 to test co-training, and remove one view network from others to test tri-training variants. Additionally, architectural variants of the best performing training methods are also re-investigated without the jumping link in the feature engineering block. All other simulation configurations are kept identical to previous case studies. The results are summarized in Table 6. It can be observed that the performance difference between the proposed approach and others is notable. In particular, the proposed approach achieves a 3% to 4.5% accuracy improvement based on co-training. This is due to that co-training employs only two learning models to process input data without tags, which may suffer from misclassification more easily. In such cases, wrongly identified training cases are retained in the training data set, which undermine the system performance. While tri-training variants also improve the performance, the missing view network still imposes negative influence on the semi-supervised training process, rendering inferior accuracy than the proposed one. Furthermore, the ablation study w.r.t. the jumping link also demonstrates its importance in further improving the system performance. When this network feature is removed from the model, both the proposed training method and the best performing existing one, i.e., tri-training with disagreement, suffer from an approx. 6% accuracy degradation. This is contributed by the fact that GPS triplegs with different lengths and volumes of latent information may require variable model complexity to be fully captured. If a network of excessive number of neuron layers is employed to handle relatively simple triplegs, the model potentially focuses more on the details of them in which sampling noise is a major component. The jumping link effectively skips unnecessary layers for selective triplegs by adopting identity mapping scheme [38], therefore overcoming the aforementioned issue.

6. Conclusions

In this paper, we propose a novel travel mode identifier based on semi-supervised deep ensemble learning. This approach can utilize the abundant unlabeled human mobility trajectory data to enhance system performance. Specifically, a new DNN architecture is proposed to employ both the time domain trajectory attributes and frequency domain trajectory statistics for travel mode identification. Based on the DNN, a neural network ensemble of four networks are constructed to generate proxy labels for unlabeled data, based on the knowledge of existing but scarce travel mode label information in the data set. These networks collaborate to determine the credibility of proxy labels, and those reliable are included in the subsequent training process for data augmentation.

To assess the performance of the proposed approach, we conduct a series of case studies based on GeoLife data set. We first evaluate the accuracy of the generated travel modes with different volume of available information. Comparing with state-of-the-art approaches for travel mode identification, the proposed one surpasses others with semi-supervised learning tests under all data set configurations. While not specifically designed for fully-supervised learning, the proposed approach can still develop satisfactory results with such a learning paradigm. Subsequently, empirical studies indicate that the accuracy of GPS positions does not have a notable influence on the system performance in practice, and feature sensitivity tests show that all data features employed by the system contribute to the identification task with different importance. Finally, we conduct a test to demonstrate the efficacy of the proposed semi-supervised training scheme.

References

[1] J. L. Adler and V. J. Blue, "Toward the design of intelligent traveler information systems," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 3, pp. 157–172, Jun. 1998.

[2] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[3] J. Scheiner and C. Holz-Rau, "Travel mode choice: affected by objective or subjective determinants?" *Transportation*, vol. 34, no. 4, pp. 487–511, Jul. 2007.

[4] F. Y. Wang, "Parallel control and management for intelligent transportation systems: concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[5] 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.

[6] J. J. Q. Yu and A. Y. S. Lam, "Autonomous vehicle logistic system: joint routing and charging strategy," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2175–2187, Jul. 2018.

[7] S. Dabiri, C. 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.*, p. 1, 2019.

[8] A. Perallos, U. Hernandez-Jayo, I. J. G. Zuazola, and E. Onieva, *Intelligent transport systems: technologies and applications.* John Wiley & Sons, 2015.

[9] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proc. International Conference on World Wide Web*, Beijing, China, 2008, pp. 247–256.

[10] "Déplacements MTL Trajet - Ensembles de données - Portail données ouvertes," accessed Jun. 2019. [Online]. Available: http://donnees.ville.montreal.qc.ca/dataset/mtl-trajet

[11] A. Bolbol, T. Cheng, and I. Tsapakis, "A spatio-temporal approach for identifying the sample size for transport mode detection from GPS-based travel surveys: a case study of London's road network," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 176—187, 2014.

[12] P. Nitsche, P. Widhalm, S. Breuss, N. Brändle, and P. Maurer, "Supporting large-scale travel surveys with smartphones – a practical approach," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 212–221, 2014.

[13] J. J. Q. Yu, "Travel mode identification with GPS trajectories using wavelet transform and deep learning," *IEEE Trans. Intell. Transp. Syst.*, under minor revision.

[14] 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.

[15] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning.* MIT Press, 2009.

[16] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Ensemble convolutional neural networks for mode inference in smartphone travel survey," *IEEE Trans. Intell. Transp. Syst.*, in press.

[17] L. Wu, B. Yang, and P. Jing, "Travel mode detection based on GPS raw data collected by smartphones: a systematic review of the existing methodologies," *Information*, vol. 7, no. 4, pp. 1–19, Nov. 2016.

[18] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, "A survey on approaches of motion mode recognition using sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1662–1686, Jul. 2017.

[19] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proc. International Conference on Ubiquitous Computing*, Seoul, Korea, 2008, pp. 312–321.

[20] 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, Jan. 2018.

[21] A. M. Güvensan and G. Asci, "A novel input set for LSTM based transport mode detection," in *Proc. IEEE International Conference on Pervasive Computing and Communications*, 2019.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, F. Bach, Ed. Cambridge, MA: MIT Press, Nov. 2016.

[23] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, "Deep feature extraction from trajectories for transportation mode estimation," in *Proc. Advances in Knowledge Discovery and Data Mining*, Cham, Switzerland, Apr. 2016, pp. 54–66.

[24] 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 *Proc. ACM International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, Singapore, Oct. 2018, pp. 1606–1615.

[25] A. Yazdizadeh, Z. Patterson, and B. Farooq, "Semi-supervised GANs to infer travel modes in GPS trajectories," 2019, `arXiv:1902.10768 [cs.LG]`.

[26] 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.

[27] J. J. Q. Yu, Y. Hou, A. Y. S. Lam, and V. O. K. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1694–1703, Mar. 2019.

[28] D. Chen, S. Wan, and F. S. Bao, "Epileptic focus localization using discrete wavelet transform based on interictal intracranial EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 5, pp. 413–425, May 2017.

[29] J. J. Q. Yu, Y. Hou, and V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3271–3280, Jul. 2018.

[30] D. H. Evans and W. N. McDicken, *Doppler ultrasound: physics, instrumentation and signal processing.* John Wiley & Sons, 2000.

[31] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[32] X. Su, H. Caceres, H. Tong, and Q. He, "Online travel mode identification using smartphones with battery saving considerations," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2921–2934, Oct 2016.

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

[34] A. C. Prelipcean, G. Gidófalvi, and Y. O. Susilo, "Transportation mode detection – an in-depth review of applicability and reliability," *Transport Reviews*, vol. 37, no. 4, pp. 442–464, Oct. 2017.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[36] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.

[37] J. J. Q. Yu, W. Yu, and J. Gu, "Online vehicle routing with neural combinatorial optimization and deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3806–3817, Oct. 2019.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp. 770–778.

[39] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, p. 947, 2000.

[40] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.* New York, NY, USA: ACM, Jul. 1998, pp. 92–100.

[41] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *ICML*, 2000, pp. 327–334.

[42] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, `arXiv:1304.5634 [cs.LG]`.

[43] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. International Conference on Learning Representations*, San Diego, CA, Dec. 2015.

[44] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY: ACM, Aug. 2014, pp. 1027–1036.

[45] J. J. Q. Yu, A. Y. S. Lam, D. J. Hill, Y. Hou, and V. O. K. Li, "Delay aware power system synchrophasor recovery and prediction framework," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3732–3742, Jul. 2019.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, CA, Dec. 2017.

[47] 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," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[48] "GPS Accuracy – Official U.S. government information about the Global Positioning System," accessed Jun. 2019. [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/

[49] "Global positioning system (GPS) standard positioning service (SPS) performance analysis report," Federal Aviation Administration GPS Product Team, Tech. Rep. Report #96, Jan. 2017.

[50] Z.-H. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge & Data Engineering*, no. 11, pp. 1529–1541, 2005.

[51] A. Søgaard, "Simple semi-supervised training of part-of-speech taggers," in *Proceedings of the ACL 2010 Conference Short Papers.* Association for Computational Linguistics, 2010, pp. 205–208.