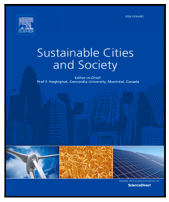
Sustainable Cities and Society 97 (2023) 104703

Contents lists available at ScienceDirect

Sustainable Cities and Society

journal homepage: www.elsevier.com/locate/scs



An ensemble federated learning framework for privacy-by-design mobility behaviour inference in smart cities

Godwin Badu-Marfo [a,1], Bilal Farooq [a,*,1], Daniel Opoku Mensah [a], Ranwa Al Mallah [b]

[a] Laboratory of Innovations in Transportation (LiTrans), Toronto Metropolitan University, Canada
[b] Cybersecurity Lab, Royal Military College, Canada

ARTICLE INFO

Keywords:
Federated learning
Ensemble model
Deep neural network
Mode inference
GPS trajectories
Mobility behaviour

ABSTRACT

Inferring the travel behaviour of users in their GPS trajectories, while protecting their privacy is a significant issue for smart and sustainable cities. To address this challenge, we use Federated Learning (FL), a privacy-preserving machine learning technique that aims at collaboratively training a robust global model by accessing users' locally trained models, but not their data. Specifically, we design a novel eNsemble federATed leArning for mobiLity InfErence (NATALIE) framework. The ensemble method combines the outputs from different DNN models learned via FL and shows an accuracy that surpasses comparable models reported in the literature. Extensive benchmarking experiments on open-access MTL Trajét and GeoLife GPS datasets demonstrate that the proposed inference model can achieve comparable accuracy in the identification of mode of travel without compromising privacy. The evaluation of the proposed model against non-i.i.d. data at varying sample sizes and different worker numbers shows improved performance. Findings are expected to contribute to the advancement of the transportation sector in smart and sustainable cities.

1. Introduction

A smart city can claim to be sustainable when it is able to run an effective transportation system that enables commuters to choose from the safest and most desirable modes for participation in spatially distributed activities. Sustainable transportation in smart cities can influence human mobility behaviour. Influencing mobility behaviour by encouraging commuters to use alternative modes of transportation has the potential to reduce energy use and emissions while minimizing traffic congestion, improving public safety, and boosting the economy, thereby significantly affecting both the quality of life and the environment (Dostál, Přibyl, & Svítek, 2020; Moudra, Matowicki, Přibyl, & Foltýnová, 2019). Governments, private organizations, and research institutions require an accurate and efficient way to comprehend citizens' mobility behaviour while improving transportation operations, modelling, and planning (Prelipcean, Gidófalvi, & Susilo, 2017). One of the accurate and efficient ways of understanding mobility behaviour is the accessibility to the right information about the travel experience of the citizens. Over the years, researchers have studied citizens' mobility behaviour by collecting information from travellers via offline or online surveys, which are costly and inefficient because of inaccurate or incomplete answers and poor response rates (Dabiri & Heaslip, 2018). Different types of machine learning algorithms such as decision tree, random forest, support vector machine, and deep neural networks have been for travel mode inference (James, 2020b; Nitsche, Widhalm, Breuss, Brändle, & Maurer, 2014). While the application of ML algorithms for inference of travel mode has been used in a number of studies, there are now serious concerns about the privacy of users who share their personal data for analysis. Although there is a growing interest and investment in the public deployment of "smart city" technologies, there are concerns about the sharing of sensitive data as users express their reluctance to do so due to the possible monitoring of their personal behaviours (Cottrill, Jacobs, Markovic, & Edwards, 2020). These concerns have resulted in strong public resistance to and later the failure of some smart city projects, for instance, the Toronto Waterfront project by Sidewalk Labs (Peel & Tretter, 2019). Therefore, there is a strong need for sustainable solutions that can incorporate privacy-by-design principles in the mobility inference process and address societal concerns (Yue, Lan, Yeh, & Li, 2014).

Ensemble-based Federated Learning (FL) technique is a promising solution to the data privacy issues by inferring users' mode of travel without compromising accuracy. As an emerging technology, FL allows large-scale nodes such as mobile and edge devices to train and

* Corresponding author.
E-mail addresses: gbmarfo@torontomu.ca (G. Badu-Marfo), bilal.farooq@torontomu.ca (B. Farooq), daniel.mensah@torontomu.ca (D.O. Mensah), ranwa.al-mallah@rmc-cmr.ca (R. Al Mallah).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.scs.2023.104703
Received 23 January 2023; Received in revised form 2 June 2023; Accepted 3 June 2023
Available online 9 June 2023

exchange models globally without revealing their local data. FL has been recently adopted in Intelligent Transportation Systems (ITS) (Al Mallah, Badu-Marfo and Farooq, 2021). Prior to our proposed model, current approaches for mode inference involved directly uploading GPS data to a centralized system for training. Again, to infer travel modes while ensuring privacy, some researchers employ anonymization techniques (Patil, Parikh, & Atrey, 2019; Stenneth & Phillip, 2010). However, the approaches proposed in the literature have some key limitations. First, they prioritize accuracy at the expense of privacy. Secondly, though accuracy is a major priority, different approaches can be achieved with better accuracy, while preserving the privacy of users' data. In order not to violate the General Data Protection Regulation (GDPR) on leakage of data, we need to develop new methods to account for the general public growing sense of privacy (Liu, James, Kang, Niyato and Zhang, 2020). To close the research gaps in privacy protection of the existing travel mode inference approaches, while addressing the specificities of travel related trajectory data, we propose a novel ensemble-based federated learning framework for travel behaviour inference. This research aims to adopt an ensemble-based structure for FL and utilize the strengths of Deep Neural Networks (DNNs) to infer users' travel mode using real-world GPS trajectories data obtained from the users' smart devices. Moreover, there is a challenge of statistical heterogeneity where the devices are used to collect data in a highly non identically distributed manner, as datasets of each data owner may come from different distribution. For instance, some users may prefer to drive to their personal vehicles whilst others may walk, which is in violation of the assumption of Independent and Identical Distributed (i.i.d.) data. In summary, statistical heterogeneity is a hard challenge in FL modelling and optimization that needs to be addressed.

To address the data privacy concerns coupled with statistical heterogeneity, we propose a novel eNsemble federATed leArning for mobiLity InfErence (NATALIE) framework. The main contributions of this study are summarized as follows:

- Design and development of a novel eNsemble federATed leArning for mobiLity InfErence (NATALIE) framework using Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convoluted Neural Network (CNN) as the base-learners and Multilayer Perceptron (MLP) as the meta-learner to integrate the optimal global model and capture the spatio-temporal correlation of GPS trajectories data.
- Integrating clustering algorithm into the proposed model in order to partition the workers and aggregate local model parameters according to the distribution of the worker dataset. This approach deals with the challenge of statistical heterogeneity to evaluate the model's robustness to non-i.i.d data.
- To the best of our knowledge, for the first time in the literature, conducting extensive benchmarking and sensitivity experiments on two large open-access GPS datasets, i.e., MTL Trajét and GeoLife, to demonstrate the performance of the proposed ensemble-based FL for travel mode inference.

The rest of this paper is organized as follows: A review of previous studies on federated and ensemble learning is provided in Section 2. Section 3 discusses preliminaries, while Section 4 discusses the methodology and the proposed ensemble-FL architecture for travel mode inference. Data and case study are discussed in Section 5. Experimental results and the application of our proposed framework on the data are provided in Section 6. Section 7 provides the results and detailed analysis. Section 8 provides a detailed discussion on implications for smart and sustainable cities and society, while Section 9 discusses the limitations of the current work. Finally, Section 10 discusses conclusions and future research directions.

2. Literature review

We discuss the state-of-the-art approaches for mobility inference based on centralized ML and privacy-preserving ML and identify key shortcomings in the existing literature.

2.1. Centralized machine learning

Travel mode inference methods based on GPS data using centralized ML can be divided into two broad categories.

2.1.1. Classical machine learning-based method

Decision trees, Bayesian network, and support vector machines have been used to classify input data (e.g., GPS trajectory, location data) for travel mode inference. For instance, Zheng, Liu, Wang, and Xie (2008) developed an instant mode inference approach that included three components: a change point-based segmentation method, an inference model, and a conditional probability post-processing algorithm. The four inference models used in the study during the inference step were decision tree, bayesian network, support vector machine, and conditional random field. Besides, the performance of the four inference models was analyzed based on two criteria: accuracy by length and accuracy by duration. When compared to other models, the decision tree attained a higher level of accuracy in travel mode identification. However, this study ignored some preprocessing steps that can cause a decrease in identification accuracy. Inspired by research conducted by Lari and Golroo (2015) and Zheng et al. (2008) used Random Forest to identify travel modes. In this study, they used attributes such as speed, accuracy, delta bearing, delta speed, acceleration, and delta acceleration for classification. The accuracy of mode identification was reported to be 96.91%. However, the study involved only three modes (bus, car, and walk) and 35 individuals.

Bolbol, Cheng, Tsapakis, and Haworth (2012) identified six modes of travel using relatively coarse-grained GPS trajectories acquired in Greater London, such as walk, car, underground, cycle, train, and bus. They accomplished 88% accuracy by using support vector machines (SVM). Another machine learning method currently adopted in travel mode identification is the Bayesian network. Xiao, Juan, and Gao (2015) proposed a Bayesian network to predict travel modes based on GPS data obtained from smartphone-based trip survey from mid-October 2013 to mid-July 2014. They used four attributes (average speed, 95% percentile speed, average absolute acceleration, and travel distance) to build Bayesian networks to classify the modes of transportation. Similarly, Stenneth, Wolfson, Yu, and Xu (2011) proposed a novel robust method for identifying modes of travel. In the proposed work, they explored and utilized transportation network data from several cities, including real-time bus, train, and bus stop spatial data. In this work, five distinct inference models were applied: bayesian net, decision tree, random forest, naive bayesian, and multilayer perceptron. The random forest model is the most dominant classification model among the inference models evaluated, with over 93% precision and recall accuracy. When transportation network categorization parameters are not taken into account, precision accuracy falls below 76%. This decline in accuracy is especially noticeable for motorized travel modes and bikes due to the absence of transportation network-related variables.

2.1.2. Deep neural network-based method

Due to recent developments in deep learning, several researchers began to focus on using deep neural networks to identify travel modes by extracting high-level data features. Gonzalez et al. (2008) employed neural networks (NN) to estimate travel modes by collecting data from smartphones for an unknown number of users and periods. The authors utilized two types of datasets: GPS points and a carefully selected subset of GPS points. They utilized 10-fold cross-validation to train and evaluate the algorithm, and the maximum accuracy was 91%. There are two potential issues with their research. The first constraint is that just 79 trips are used for training, which is insufficient for reliability, and the dataset was manually labelled by users. Both drawbacks might have an impact on mode identification accuracy. Another drawback was that the study only analyzed three different modes of transportation. Byon and Liang (2014) employed the neural

networks (NNs) technique to infer different mode of travel. The purpose of this study was to evaluate the performance of mode inference using NNs between conventional GPS data and smartphone data. In this study, they focused on two unique aspects. First, they assessed the effect of sample rates and monitoring durations on mode identification accuracy. Second, they distinguished between travel modes under various settings, such as peak against non-peak situations, generic versus route-specific, and fixed versus no fixed orientation of smartphone. Similarly, Yang, Yao, and Jin (2015) employed a Convolutional Neural Network (CNN) model to identify the mode of travel for each trip segment. As a result, the identification of travel modes is more than 86% accurate. Furthermore, the accuracy of bus mode identification is better than in any other study.

Dabiri and Heaslip (2018) proposed an ensemble CNN for mode inference in GeoLife dataset, an open-access labelled dataset provided by Zheng, Fu, Xie, Ma, and Li (2011). An accuracy of 84.8% was achieved in terms of mode inference by testing various configurations of CNN models. In a work by Yazdizadeh, Patterson, and Farooq (2019), an ensemble CNN was proposed to identify four modes of transportation (walk, bike, automobile, and public transportation) in MTL Trajét dataset, one of the largest labelled open-access dataset made available by Montréal (2018). They used a centralized ML strategy to evaluate several CNN architectures and merged their results using different ensemble models (average voting, majority voting, optimal weights, and Random Forest meta-learner) to get a higher prediction accuracy. They detected the segments using the trip-breaking technique, taking into account the dwell time between GPS location recordings. Based on 3-min gaps in the data, the trip-breaking algorithm identified trip segments. The ensemble method, using the RF as a meta-learner, produced an accuracy of 92%, which is superior to the performance achieved in Dabiri and Heaslip (2018) on GeoLife dataset.

Recently, James (2020a) proposed DNN, a semi-supervised deep ensemble learning algorithm, for transit mode identification when only a small amount of labelled data is accessible. The author developed a novel DNN architecture that uses both the time and frequency dimensions of trips to determine transport modes. Leveraging on the DNN architecture, a neural network consisting of four networks was designed to produce proxy labels for unlabelled data using knowledge of available, but sparse, transit mode label information in the dataset. Yazdizadeh, Patterson, and Farooq (2021) developed a semi-supervised Generative Adversarial Network (GAN) model on GPS data from Montréal for mode inference. The GAN model achieved a performance of 83.4%. It was also shown that GAN performed better than CNN based semi-supervised model. Kim, Kim, and Lee (2022) developed a Long Short-Term Memory (LSTM) architecture for mode inference. They mixed the GeoLife dataset with their own proprietary dataset involving 105 additional users. The resulting model achieved an accuracy of more than 90% and outperformed the ensemble CNN model.

2.2. Federated learning

The prevalent strategy is the central server approach, where data is sorted from citizens and training and inference on that data take place on the server. Mobility inference models are primarily using this centralized approach to train their models. Such an approach may pose privacy issues for the citizen. In the federated learning approach, the training and inference are performed on the citizen's devices without them sharing their data. At each iteration of distributed learning involving DNN training, the central coordinator (referred to as chief) distributes the workload among the participants (referred to as workers) (Dean et al., 2012). After distributing sections of the model training to the workers, the chief aggregates the gradients they returned and modifies the model's weights before disseminating the updated weights to all employees in the subsequent iteration. This eliminates the concern about data privacy and decentralizes ML since workers with fewer data may train the ML model in conjunction with workers who have more data by working together through the FL technique.

Depending on how the data is distributed, federated learning may be classified into three groups: horizontal FL, vertical FL, and federated transfer learning. When data samples from several workers have the same properties, the vanilla FL arrangement is equivalent to horizontal FL. Specifically, worker nodes track the same attributes for the data points even while they do not have any data points that overlap. In vertical FL, workers share the same samples, but features are distributed among them. Federated Transfer Learning (FTL) is another FL setup where transfer learning is used to create customized models from the global model (Liu, Kang, Xing, Chen and Yang, 2020). When two workers have datasets that vary not just in terms of samples but also in terms of feature space, FTL is employed. The application of DNNs in transfer learning to explore implicit processes has become prevalent. A specific instance of FL is model fusion. A global neural network must be built in the most significant case using just one communication round (Claici, Yurochkin, Ghosh, & Solomon, 2020). To average distinct neural networks, model fusion approaches attempt to build some sort of relationship between their neurons.

In recent studies, the idea of federated learning has been adopted in smart mobility systems. Fiosina (2021) developed a horizontal FL framework to evaluate taxicab trip time in Brunswick utilizing Floating Car Data (FCD) trajectories collected from two distinct taxi service providers. This FL approach has enabled the processing of distributed data while maintaining their privacy. Al Mallah, Badu-Marfo et al. (2021) and Liu, James et al. (2020) have used federated learning to predict traffic flow, while protecting privacy. Both studies reported that the federated version achieved comparable performance to the centralized version. Similarly, Kweon, Sun, and Park (2021) used federated learning to forecast driver route choice decisions in route navigation systems, and the model's effectiveness was successfully demonstrated by comparing its performance to a centralized server-based model. The findings of real-world route navigation usage data from around 30,000 vehicles collected over a year proved that the proposed federated learning technique achieved comparable accuracy to the conventional centralized global model while ensuring privacy. Sharma, Park, and Cho (2020) combined blockchain and federated learning technologies to develop an image classification and defence framework for the Internet of Things in smart and sustainable cities. Ramu et al. (2022) outlined a conceptual framework for federated learning enabled digital twins for smart cities. It was noted that several key challenges, such as privacy and data availability, still exist in the large-scale operationalization and utilization of this integration for social good.

2.3. Concluding remarks

The mobility inference problem using GPS data has extensively been studied in a centralized manner and several deep learning based architectures are proposed. Among these architectures, LSTM and CNN based ensemble architectures have shown promising results. LSTM ensures that the sequential characteristics of GPS data are incorporated into the model, while the ensemble of models ensures that the training and inference are robust to the specificities of different modes. However, a centralized approach pose privacy and cybersecurity issues (Ramu et al., 2022). A federated architecture for mobility inference can potentially overcome these issues. However, there is a clear lack of such a comprehensive FL architecture for mobility inference that can (a) take advantage of the recent methodological developments in centralized deep learning, (b) address specific issues resulting from the distributed architecture, and (c) develop comprehensive benchmarking on multiple datasets. Another shortcoming in the mobility inference literature is the availability of rich and open-access labelled GPS datasets for testing and benchmarking. Currently, only two open-access datasets, i.e., GeoLife (Zheng et al., 2011) and MTL Trajét (Montréal, 2018), exist that have enough participants, trajectories, and modes that they are suited for federated learning based model training and inference.

3. Preliminaries

Here we first explain the basic building blocks that are used to develop our proposed framework.

3.1. Deep neural networks for FL process

Based on the recent advances in deep learning, we chose the four most suitable models for the proposed FL based mobility inference architecture.

Multilayer Perceptron: Multi-layer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. The MLP models are the most fundamental deep neural network, consisting of several fully linked layers (Feng & Timmermans, 2016). Except for the input nodes, each new node is a neuron with nonlinear activation functions of a weighted sum of all outputs (fully connected). Consider training sets where $x_i \in R^n$ and $y_i \in 0, 1$, an MLP learns the function

$$f(x) = W_1 g(W^T x + b_1) + b_2 \tag{1}$$

where $W \in R^m$ and $W_1, b_1, b_2 \in R$ are model parameters. W, W_1 are the weights of the input layer and hidden layer, respectively; and b_1, b_2 are the bias of the hidden layer and the output layer, respectively. $g(.)$: Softmax function, which is given as,

$$Softmax(z)_i = \frac{exp(z_i)}{\sum_{l=1}^{k} exp(z_1)} \tag{2}$$

where z_i represents the ith element of the input to Softmax. Hence, the loss function for classification is given as,

$$Loss(\hat{y}, y, W) = \frac{1}{n}\sum_{i=0}^{n}(y_i ln\hat{y}_i) + ((1 - y_i) ln(1 - \hat{y})) \tag{3}$$

Convolutional Neural Networks: While CNN has primarily been used in the processing of images, it has also shown promise in evaluating patterns in sequences with long-term dependencies. The CNN is made of an input layer, hidden CNN layers, and fully connected layers that end with an output layer (Abdeljaber, Avci, Kiranyaz, Gabbouj, & Inman, 2017). For CNN, we compute the output of convolution by convolving an input(I) with a number of filters as follows:

$$x_j = I * Wj + b_j, \quad j = 1, 2, 3, \ldots, F \tag{4}$$

where F is the number of filters, x_j is the output that corresponds to the jth convolution filter, W_j is the weights of the jth filter, and b_j is the jth bias. For batch normalization, scale and shift the normalized values as follows:

$$y_i = \sigma\hat{x}_i + \beta \tag{5}$$

where y_i is output value, $\hat{x}_i$ is the normalized input value, and σ and β are the scale and offset factors that are learnable during the network training. We use ReLu as activation function for the equation:

$$f = max(0, y) \tag{6}$$

where y is the input value to ReLU, and f is the output. Repeat steps from Eqs. (5)–(7) for each subsequent convolutional layer. After the ReLU module, the feature map f is used as the input (I) of the next convolutional layer. After the final convolutional layer, apply the Softmax layer to create the probability distribution of the classification results. Let ϕ_i denote the feature map that corresponds to the ith class (i = 1, 2, …, M). Then Let ϕ denote the exponential summation of the features map calculated as follows:

$$\phi(k, l) = \sum_{i=1}^{M} exp(\phi_i(k, l)) \quad 1 \le k \le K, 1 \le l \le L \tag{7}$$

Class probability P_i for the data is then calculated using the following equation:

$$P_i(k, l) = \frac{exp(\phi_i(k, l))}{\phi(k, l)} \tag{8}$$

Long Short-Term Memory (LSTM): As a variant of the classic Recurrent Neural Network architecture, LSTM implements four interactive gates to allow learning sequence labels for extensive time intervals (Kalatian & Farooq, 2021). This fixes the vanishing gradient issue with RNNs as they are unable to retain long term dependencies or contexts (Graves, 2013). The unit of LSTM neural network has three gates and a memory cell: input gate (i_t), forget gate (f_t), output gate (o_t) and memory cell (c_t) (Ran, Shan, Fang, & Lin, 2019). The memory cell is in charge of sending data via the network, and the three gates either filter out unnecessary information or add pertinent data to the memory cell. The forget gate ignores information that is not necessary to learn about the predictions and chooses whether the information may proceed through the different network layers while the input gate modifies the cell state to determine the significance of the information. The output gate, last gate of the circuit determines the next hidden state of the network. In a typical LSTM neural network, the parameters that are used during training include the sigmoid function, the input vector that is fed into the input layer of the LSTM unit, the output vector of the hidden layer of the LSTM unit and the memory cell of the LSTM unit. The equations for the LSTM gates are:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{9}$$

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{10}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{11}$$

$$c_t = tanh(w_c[h_{t-1}, x_t] + b_c) \tag{12}$$

where x_t = input to the LSTM layer at time t; w_f, w_i, w_o and w_c are the weight matrices for mapping the input layer into the three gates and the cell state; x_t is the input at the current timestamp; h_{t-1} is the output of the previous lstm block (at timestamp t − 1); b_f, b_i, b_o and b_c are biases for the 3 gates and the cell state; σ represents the sigmoid activation function while tanh represents the activation function for the current cell state. The current cell state is computed as:

$$\hat{c}_t = f_t.\hat{c}_{t-1} + i_t.c_t \tag{13}$$

Hence, the output of the LSTM layer h_t is computed as:

$$h_t = o_t.tanh(\hat{c}_t) \tag{14}$$

Gated Recurrent Unit: Another modification of Recurrent Neural Network (RNN) that handles times-series data is the Gated Recurrent Unit (GRU), proposed by Cho et al. (2014). According to Chung, Gulcehre, Cho, and Bengio (2014), the GRU has gating units that controls the flow of information inside the unit, without having separate memory cells. It uses the hidden state to transfer information and only has two gates: a reset r_t and update gate z_t (Liu, James et al., 2020). The reset gate is the network's short-term memory that determines the number of prior data to erase. On the other hand, the update gate is the network's long-term memory, deciding what information to discard and what new information to include. The equation of the GRU layer can be explained as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{15}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{16}$$

$$h_t = tanh(W_h x_t + U_h(r_t.h_{t-1}) + b_h) \tag{17}$$

$$\hat{h}_t = z_t.\hat{h}_{t-1} + (1 - z_t).h_t \tag{18}$$

where r_t and z_t represent reset gate and update gate at time t, respectively; x_t denote input to the GRU layer at time t; h_t represent

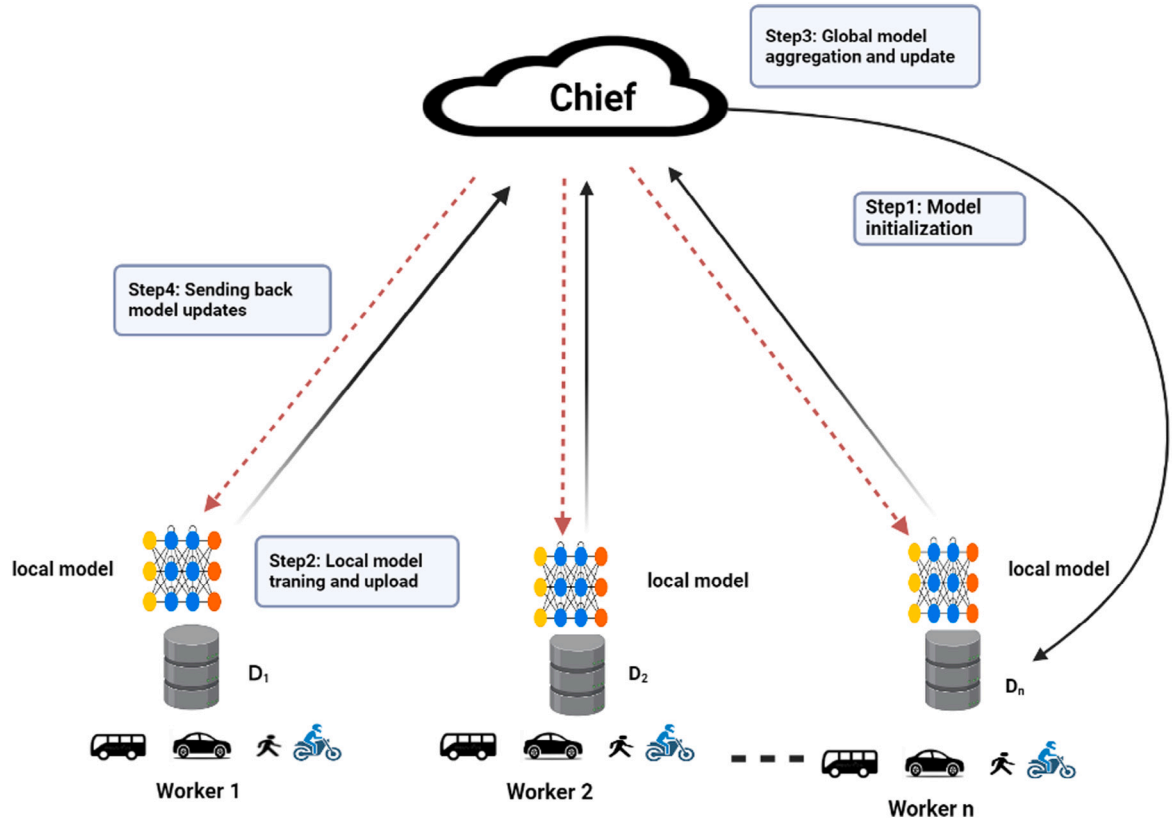


Fig. 1. Federated learning-based travel mode inference architecture.

the candidate activation vector while W_r, W_z and W_h are the weight matrices; b_r and b_z represent the bias vectors for the reset gate and update gate, respectively; σ represents the sigmoid activation function while tanh represents the activation function.

3.2. Federated Averaging (FedAvg) algorithm

Typically, FL systems employ the Federated Averaging (FedAvg) algorithm to train a shared global model collaboratively through multiple decentralized users without sharing the raw data (McMahan, Moore, Ramage, Hampson, & y Arcas, 2017). It computes the average of the clients' local model updates as the global model update, where each client is weighted by its number of training examples. Using a neural network algorithm, local model parameters are built from each user's (i.e., worker's) own data that is likely to have different numbers of data points (i.e., mode of trips). The FedAvg algorithm combines the local model updates of each worker at a chief node through model averaging. The training steps of FedAvg are outlined as follows:

- *Model Initialization:* Before conducting the FL training task, the chief uses the public dataset to pre-train a global model. Then, at each round of training t, the chief randomly selects a subset of workers to participate in the FL task. The chief sends the pre-trained global model parameters wt to each participating worker c, in the FL framework via a secure mechanism.
- *Local model training and upload:* Each worker trains the received global model for several epochs E with local data D_c. For each epoch, the worker conducts gradient optimization by batch size to find optimal parameters to minimize its local loss function. Then each worker sends back its own model updates to the chief.
- *Global model aggregation and update:* The chief uses FedAvg algorithm to aggregate the model parameters of all participating workers in the training through a secured mechanism and updates the global model for the next iteration. The system makes a repetition of the training process until the global model achieves convergence.
- *Sending back model updates:* Finally, the chief broadcasts the updated global model parameters back to the workers for the next training round and aims at minimizing the global loss function. The process is iterated until the global loss function converges.

FedAvg has demonstrated effectiveness in conventional Federated Learning problems, however it suffers from slower convergence and low accuracy in most non-i.i.d. contents (Li, Huang, Yang, Wang, & Zhang, 2019).

4. Methodology

Here, we present our proposed framework to address the mobility behaviour tasks described in Section 1. First, we explain the architecture of our proposed framework. Second, we describe the design of the eNsemble federATed leArning for mobiLity InfErence (NATALIE). The proposed NATALIE framework is designed to achieve model robustness against non-i.i.d. data distribution (see Fig. 1).

4.1. General setup

We use the term *'worker'* to denote each participant in the FL framework, and the term *'chief'* to define the participant that aggregates workers' model parameters. Hence, let $workers$, $P = \{P_1, P_2, \ldots, P_n\}$ with each of them equipped with a local dataset $D_n = \{D_1, D_2, \ldots, D_n\}$. The problem is to build an ensemble model based on all local datasets with the existence of a $chief$. Furthermore, each local dataset evolves with the time. Upon a new local dataset arriving, the old dataset will be instantly discarded and no longer available for the training stage.

In this work, we assume a dataset of GPS trajectories depicting motion characteristics of members of the population across the main travel modes. First of all, we extract trip characteristics, including

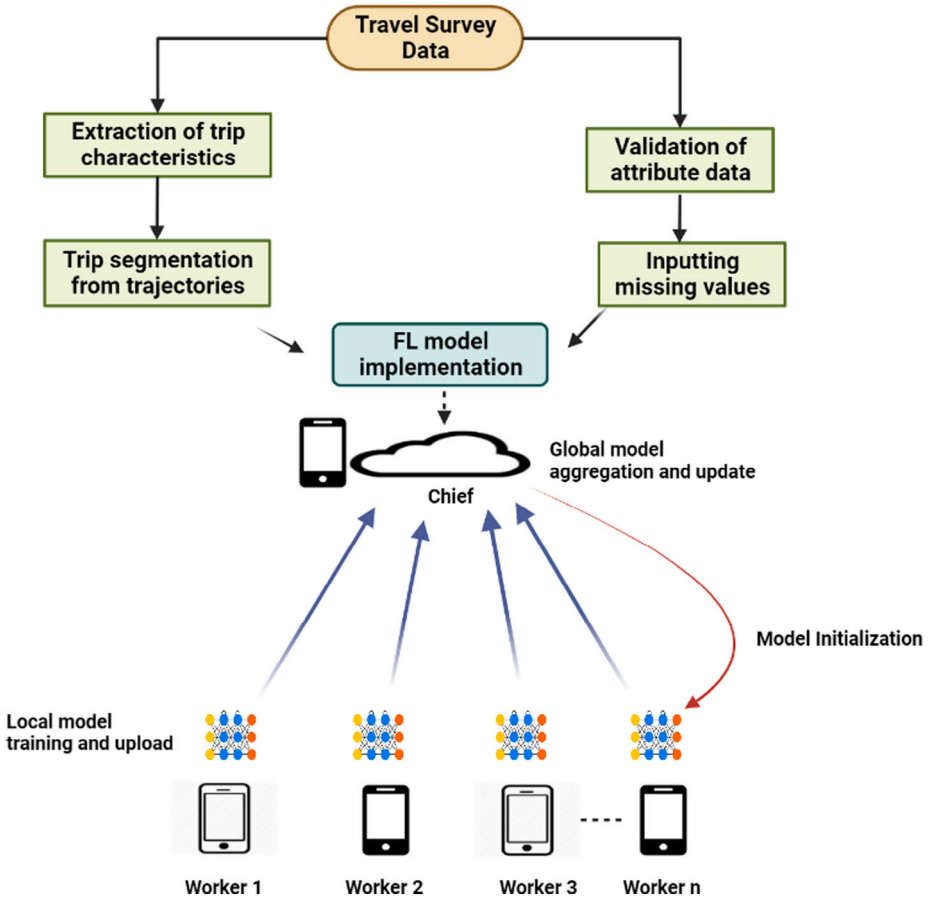


Fig. 2. Steps for model implementation.

"speed, acceleration, and jerk" from sequences of GPS logs recorded from smartphones having GPS capability. The motion characteristics are subsequently segmented into sequence lengths of ten (10) data points having three (3) features (speed, acceleration, jerk). These input features are scaled and normalized to improve training performance in the neural networks. For sequences with less than a length of ten (10), zero padding is performed to achieve a fixed sequence length. The prediction output comprises thirty-five (35) distinct classes of travel modes. These classes are pre-processed and one-hot encoded in the standard format for DNN training. Fig. 2 shows the step-by-step approach that describes the implementation of the FL model.

We implement ten (10) worker nodes and use a single chief node to aggregate local model updates from the worker nodes. On each worker node, three (3) base learners comprised of LSTM, CNN, and GRU are deployed. Similarly, these base learners are deployed at the chief for aggregation. When the chief receives the local updates for each iteration, it aggregates for each of the base learner models. After aggregation, an ensemble stacking generation is undertaken, which stacks the prediction outputs from the base-learners to a meta-learning framework composed of a multi-layer perceptron. We observe the accuracy of the classification predictions from the aggregated base learners and the ensemble stacked model when inference is undertaken on the testing dataset.

4.2. Description of DNN models for NATALIE framework

Our proposed ensemble model uses deep neural networks for the classification task. Thus, we use CNN, LSTM, GRU and MLP.

CNN is the first proposed model for NATALIE, as shown in Fig. 3. It is composed of 3 different convolutional layers that use Rectified Linear Unit (ReLu) as the activation function for the layers. They are followed by a dense layer that has a Softmax activation function for the layer. To prevent overfitting, a dropout method was used to add penalty to the loss function by randomly reducing inputs during the training. A dropout layer with a value of 0.2 was added after each convolutional layer.

LSTM is the second model used in the framework and is shown in Fig. 4. It is made up of 3 distinct blocks of LSTM. The initial block has a recurrent dropout rate of 0.2, which helps in dropping the fraction of units needed for the recurrent state's linear transformation. The three different blocks of LSTM layers implement Leaky Rectified Linear Unit (LeakyReLu) as the activation function for the layers. They are followed by a dense layer that has a Softmax activation function for the layer.

GRU is the last model used in NATALIE framework, as shown in Fig. 5. The model has a similar architecture to LSTM. It comprises of three distinct GRU blocks and a recurrent dropout rate with a value of 0.2 that can help drop a fraction of units for the linear transformation of the recurrent state

4.3. eNsemble federATed leArning framework for mobiLity InfErence (NATALIE)

Our proposed framework, eNsemble federATed leArning for mobiLity InfErence (NATALIE) is based on DNNs discussed in the previous section. NATALIE, an ensemble-based system that integrates the optimal global model and captures the spatio-temporal correlation to achieve high predictive performance. Building on our initial work, eFedDNN (Mensah, Badu-Marfo, Al Mallah, & Farooq, 2022), NATALIE aims to reliably infer the mobility behaviour leveraging FL and DNNs, while maintaining privacy. eFedDNN used a vanilla version of FL and a standard ensembling scheme with a small scale application.

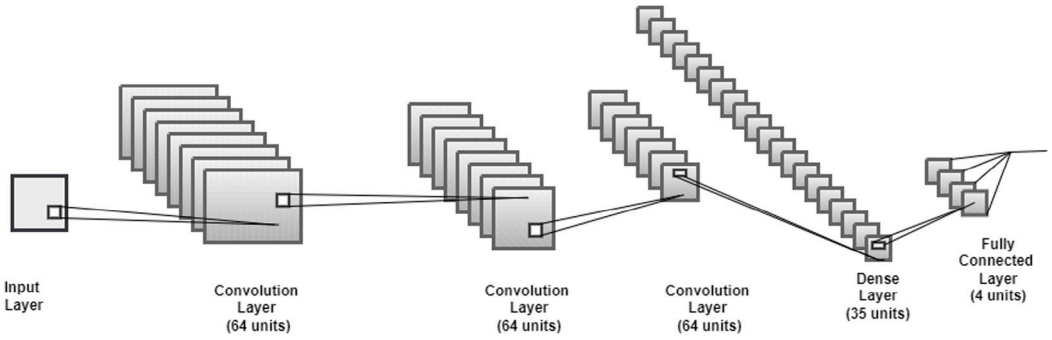


Fig. 3. Architecture of CNN model.

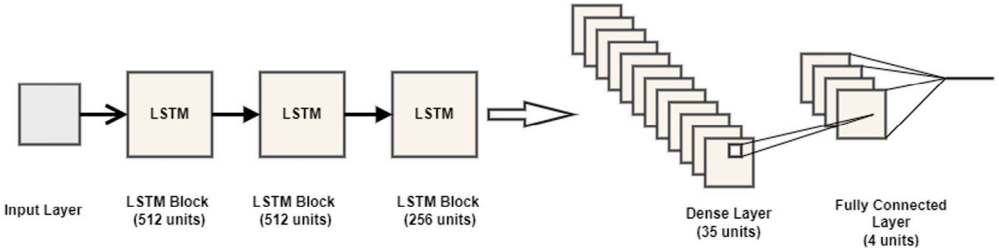


Fig. 4. Architecture of LSTM model.

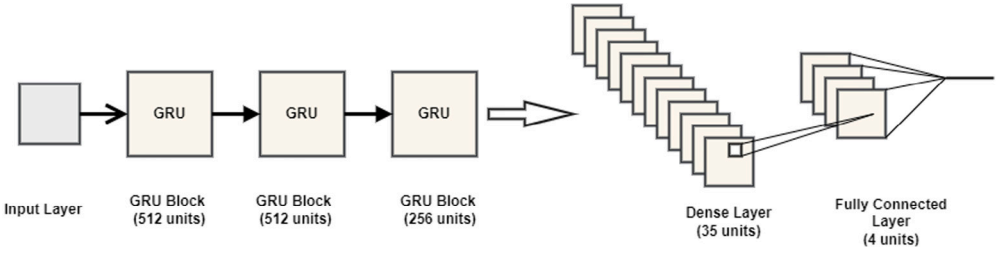


Fig. 5. Architecture of GRU model.

Furthermore, the proposed model could not account for non-i.i.d. data and lacked extensive performance, sensitivity, and scalability analyses.

In this study, we employ different DNNs: Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), Convolution Neural Network (CNN), and the Multi-Layer Perceptron (MLP). We use DNNs for the ensemble approach since they are nonlinear techniques with enhanced flexibility and the capacity to scale depending on the amount of training data available (Hinton, Vinyals, & Dean, 2015). To infer human mobility behaviour, we demonstrate that the DNN-based ensemble that functions as the local model on the device of each worker in the FL process must learn every sample of the GPS dataset located on each smartphone device. For the proposed ensemble model, we use the stacked generalization (stacking) approach, which combines the outputs of several base learners (LSTM, GRU, CNN), and another algorithm, the meta-learner, to compute the final model predictions. For the purpose of aggregating weights, we assign the base learners at both the workers and the chief nodes while MLP is used as the meta-learner at the chief node to receive inputs of the predictions from the average global models of the base learners. Fig. 6 is a two-tier architecture of NATALIE depicting the base learners as GRU, LSTM, and CNN with MLP as the meta learner.

During the FL process,workers have GPS datasets kept on each device. Each neural network classifier at the chief node predicts inputs sent to the next model for stacking. The chief begins the model training by establishing base learners of certain types upon determining the number and type of base learners. The learner parameters, w_o, are randomly initialized. The parameters are updated iteratively for a total number of rounds via communication between the chief and all workers. The chief aggregates the local model parameters from the workers and calculate the final model by averaging the models. Particularly, each worker receives a copy of the model parameters from the chief via broadcasting to modify its local model parameters at each round t. The chief computes a final model via model averaging after collecting model parameters from all of the workers. The chief computes a final model via model averaging after aggregating model parameters from the workers and sends the new global model to each worker.

As a classification problem, we combine the predictions of the individual classifiers to achieve the best model performance for the stacking ensemble method. To achieve a high predictive performance for the model, we apply the voting technique in which each of the neural networks for prediction assigns a vote for the class that it predicts. We use three different approaches of voting to combine the predictions of the classifiers to evaluate the performance of the proposed model. The voting techniques include (a) average voting (b) majority voting, and (c) weighted voting. Regarding average voting, we employ the average of all the DNN model predictions to determine the final prediction by computing the unweighted average of the labels from the base learners and selecting the highest value. With respect to majority voting, the predictions obtained from majority of the DNN models are used for the final prediction. For weighted voting technique, we determine the optimal weights of the base learners by minimizing the loss function so that classifiers with better performance are assigned more weights.

The eNsemble federATed leArning framework for mobiLity InfErence (NATALIE) is thus presented in **Algorithm** 1. Let $W_k = \{w_1, w_2,$

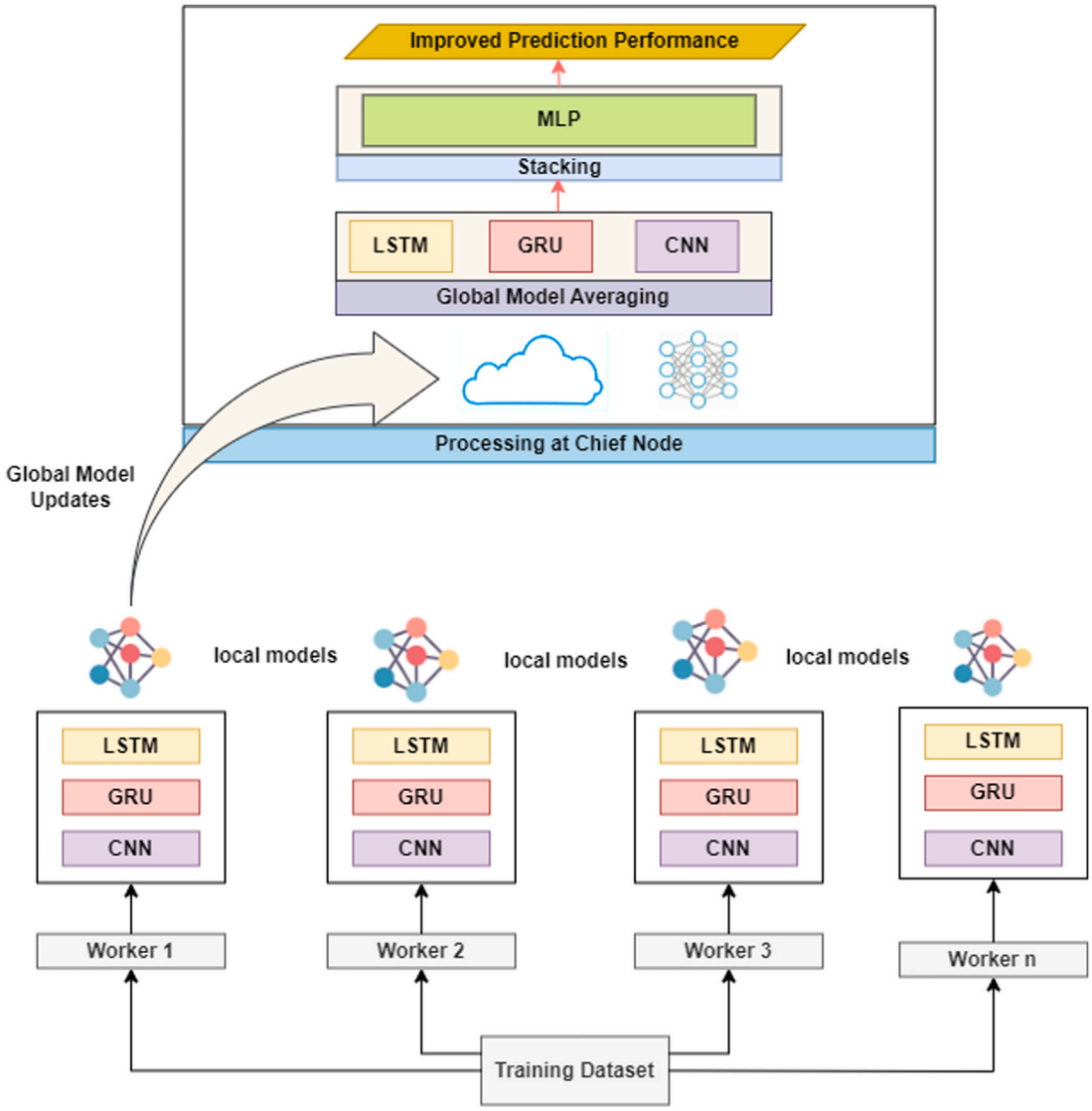


Fig. 6. Architecture diagram of NATALIE framework.

$\ldots, w_k\}$ denote the global model of the optimal set P_k. As shown in Fig. 6, we use the NATALIE to find the optimal ensemble model by integrating the global model with the best accuracy after executing the NATALIE framework.

4.4. Dealing with non-i.i.d. data distributions

Given that the workers' data are in non-i.i.d. format, it becomes challenging for the global model to converge in order to obtain high accuracy. In practical terms, data from each of the workers may not meet the requirement of i.i.d. data distribution and it may result in challenges with optimization and convergence of NATALIE model. Hence, the proposed NATALIE framework makes a significant empirical improvement that guarantees convergence.

The proposed algorithm implements K-means algorithm (Fahad & Alam, 2016) with the aim of grouping the workers into clusters before implementing the NATALIE. The clustering decision is determined by using the latitude and longitude information of the workers. The algorithm partitions the training instances into k clusters using Euclidean distance similarity. The main purpose of this approach is to determine the cluster center that minimizes error. Specifically, K-means algorithm uses the class distribution as a measure to partition the worker updates into groups according to the Euclidean distance of the class distribution. As shown in Fig. 6, we employ the DNN-based ensemble model to determine the optimal ensemble model by integrating the optimal global model from the neural network after executing the K-means algorithm

The main steps of the proposed model are summarized as follows:

- **Step 1:** Before conducting the FL training task, we randomly initialize clusters of the datasets and implement K-means algorithm. Before then, the chief node uses proxy dataset to pre-train a global model. At the same time, the chief selects a subset of workers at random from a subset of dataset, D_n.
- **Step 2:** The chief broadcasts the pre-trained global model parameters, w_o to each selected worker in the FL framework via a secure mechanism. Each of the selected workers $n \in B$ perform gradient optimization by batch size to find optimal parameters to minimize its local loss function. Then each worker sends back its own model updates to the chief.
- **Step 3:** The chief aggregates the model parameters of all participating workers in the training through a secured mechanism and updates the global model for the next iteration. The system makes a repetition of the training process until the global model achieves convergence.

Algorithm 1 eNsemble federATed leArning framework for mobiLity InfErence (**NATALIE**)

Input: Worker datasets $D = \{D_1, D_2, D_3, \ldots, D_n\}$, Worker set $K = \{k_1, k_2, \ldots, k_n\}$, # of epochs, E and learning rate, α, # of clusters c, where n > c
Output: Optimal global model $\{\emptyset\}$;
Randomly divide n workers into batches: $B \leftarrow \{B_1, B_2, \ldots, B_n\}$
Initialize global model parameter, w_o ;
for $t = 1, 2, 3 \ldots, T$ **do**
  // **The side of workers**
  for $n = 1, 2, \ldots, N$ **do**
    Randomly select k_n worker from D_n;
    Receive w_o from chief;
    Chief broadcasts w_o to workers B_n;
    for *each worker n in B* **do**
      Accuracy ← validate (output);
    end
    Send the updated parameters,w_k to the chief;
  end
  // **The side of the chief**
  Receive w_k from all workers;
  for *each DNN d in DNNs* **do**
    Execute d;
  end
  Obtain the global model set, $\{\emptyset\}_k$;
  Execute the ensemble to find $\{\emptyset\}$;
  Chief sends $\{\emptyset\}$ to each worker;
end
return Global model the ensemble classifier $\{\emptyset\}$;

4.5. Ensemble types of NATALIE

To evaluate the stability and robustness of NATALIE model, we design different ensemble types of the model. Fig. 7 shows NATALIE framework that are designed with different DNN classifiers. In Fig. 7a, we use LSTM and CNN algorithms as base learners to design a DNN ensemble model while MLP is used as meta learner for combination of the classifiers. Similarly, in Fig. 7b, we use LSTM and GRU algorithms as base learners for designing of NATALIE framework and MLP as meta learner for classifiers combination.

5. Primary data and case study

5.1. Dataset details

In this study, we used MTL Trajét, a real-world open-access GPS trajectories dataset collected in 2017 (Montréal, 2018). It comprises over 13 million GPS location points representing 185,285 labelled trips from more than 5000 individuals. The mobility data was collected via a smartphone application with individuals who willingly participated in the trip survey. When travels ended, participants confirmed their trips by identifying their mode of transport and purpose of trip (i.e., school, home, job, business). The modes included, walking, bike, automobile, public transit, carshare, taxi, and any valid combination of these modes (e.g., walking and public transit) to complete a trip, resulting in a total of 30 different modes.

5.2. Data processing

During the data processing stage, we split the users' trip mode data randomly to make sure that the training and validation data are not exposed to the assessment of the proposed FL model approaches. First, we split 5% of all trip data to serve as a proxy dataset which is used to train the global model. Then, we divide the rest of the data by using 80% of the data for training and 20% is used for testing.

The raw GPS data is the user's travel information acquired by the smartphone device over a period of time, including longitude, latitude, and sample timestamp. As a result, we transformed the original GPS data with the same travel mode into a trip based on its timestamp. Let $G = \{G_1, G_2, \ldots, G_n\}$ represent the GPS record in the segment with length, N. Each GPS record is represented by a triple $G_i = (lat_i, long_i, t_i)$ as the latitude (lat_i) and longitude ($long_i$) of the device's location at the time of t_i. For two consecutive records G_i, G_{i+1}, we utilize Vincenty formula (Vincenty, 1975) to calculate the relative distance:

$D_i = \text{Vincenty}(lat_i, long_i, ; lat_{i+1}, long_{i+1})$ Indicating the time interval between G_i and G_{i+1} as Δt_i, based on the relative distance (D_i), we can calculate the first three motion features based on speed (S_i), acceleration (A_i), and jerk (J_i) of the R_i location using these equations:

$$S_i = \frac{D_i}{\Delta t_i}, \qquad 1 \le i \le N, S_N = S_{N-1} \tag{19}$$

$$A_i = \frac{S_{i+1} - S_i}{\Delta t_i}, \qquad 1 \le i \le N, A_N = 0 \tag{20}$$

$$J_i = \frac{A_{i+1} - A_i}{\Delta t_i}, \qquad 1 \le i \le N, J_N = 0 \tag{21}$$

According to Yazdizadeh et al. (2019), speed is computed using the distance between each two successive GPS points divided by their time interval, whilst acceleration is also defined as the derivative of speed or the rate of change of speed over time. On the other hand, jerk is the rate at which acceleration changes, and it is an important element in public transportation safety problems such as critical driver movements and passenger balance (Bagdadi & Várhelyi, 2013). In this work, we look at the relative distance, speed, acceleration, and jerk rate. These attributes comprise the channels of each GPS segment utilized to train the algorithms. Fig. 8 shows the features that constitute the channels of each segment.

6. Experimental setup

In this section, we analyze the model's performance and explain the outcomes of implementing the proposed model to a real-world open-access GPS trajectory dataset. We compare the performance of the proposed model with the centralized ML model. Again, we test the model's robustness by assessing its performance against i.i.d. and non-i.i.d. data. The mobility data is given and evaluated in order to evaluate the performance of the users' travel mode. We discuss the metrics that are used to evaluate the model's performance.

6.1. Evaluation metrics

This section describes the model performance based on the evaluation metrics for the classification problem. We adopt the accuracy score, precision, recall and F1 score for the model evaluation. As a classification problem, the basic measure for model evaluation is accuracy, which specifies the number of accurate predictions over all predictions. It assesses model performance by determining the ratio of true positives to true negatives based on all values predicted. The accuracy score is computed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative.

Precision is a measure of how often positive predictions are true. It is expressed as the relative ratio of accurately classified instances to expected positive instances. Precision is computed as follows:

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

where TP is true positive while FP is false positive.

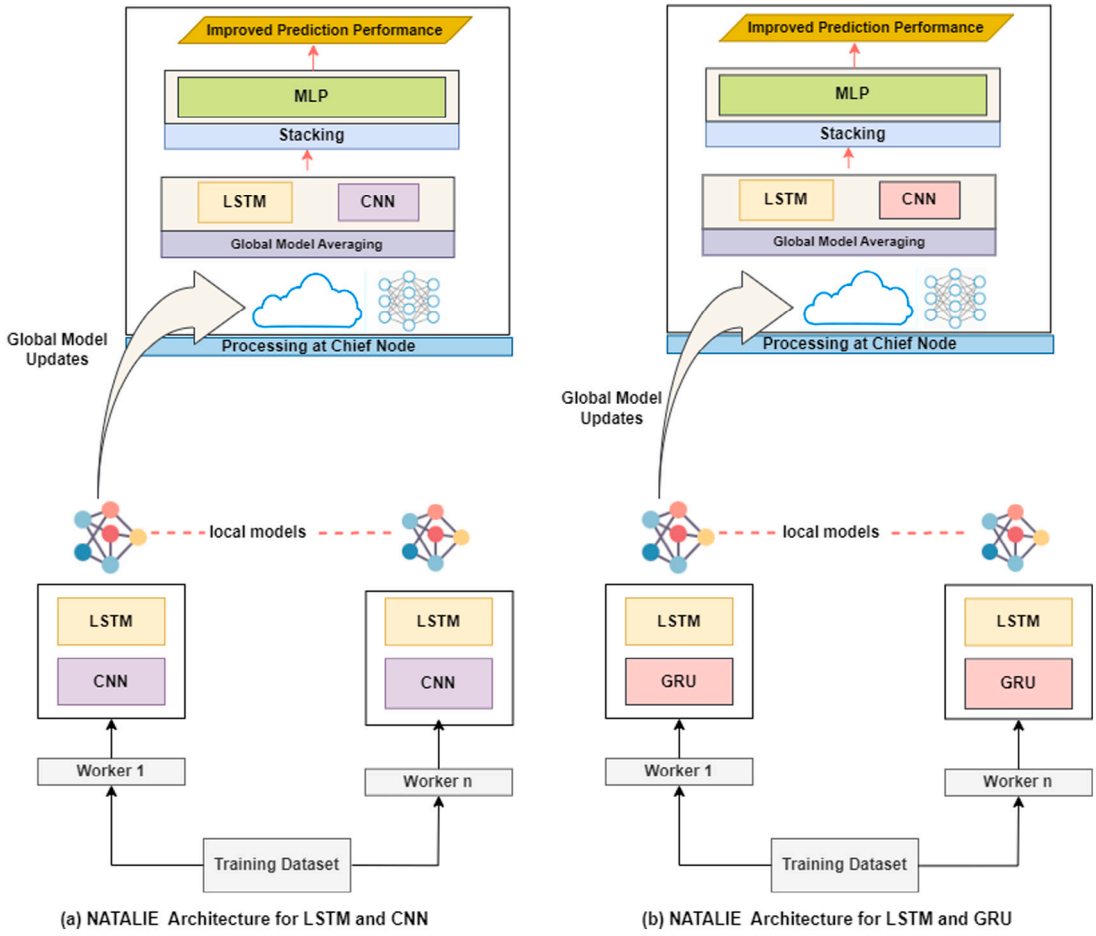


Fig. 7. NATALIE architecture for different classifiers.

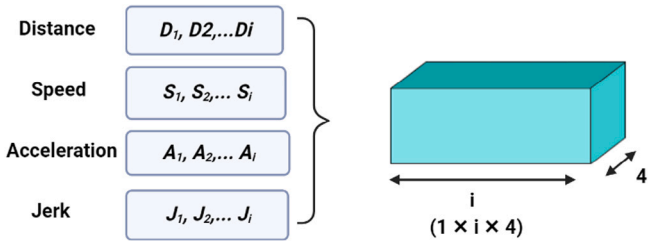


Fig. 8. The channel structure of a GPS segment.

Recall measures the number of the positive instances the model correctly predicted, over all the positive instances in the data. It is computed as follows:

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

where TP is true positive and FN is false negative.

F1-Score is a statistic that combines precision and recall and is sometimes referred to as the harmonic mean of the two. The F1-score is calculated as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{25}$$

6.2. Model parameter configurations

All simulations that require data pre-processing and model creation are done using Python programming language and PyTorch library with GPU support. Following the data planning phase, a search is conducted to identify the ideal network configurations. Dropout layers and their rates, the number of hidden units, the number of hidden layers, and the density of the layers are adjusted. A Core i7 4 GHz CPU and 16.0 GB of RAM are used to train the models.

We train a series of neural networks on each device during the experiment using 10 workers as a default. We separate the workers into three groups, one of which has four workers, to prevent overfitting. Three separate workers make up each part, which is trained using a different neural network design. We distribute the training data to 10 workers. To as closely as possible simulate how non-i.i.d. data are distributed in the real world, each worker has at least one travel mode available to them. The workers have access to the GPS datasets recorded on each device as part of a federated learning process. We develop ensemble base-learners made up of three deep learning models that can predict sequences: LSTM, GRU, and CNN. These based models are designed to accept speed, acceleration, and jerk estimated from time-stamped GPS logs of a mobile travel diary as input features of travel motion characteristics.

Each DNN model has two (2) hidden layers with a Rectified Linear Unit (ReLu) activation function. The output layer of the models has a linear dense layer with a SoftMax activation and is intended for multi-modal classification. Each DNN model has two optimizers in the NATALIE algorithm: the chief optimizer and the worker optimizer. The worker optimizer trains the local devices of the workers while the chief optimizer applies the averaged worker updates to the chief's global model. Adam optimizer was used for both worker and chief with two distinct learning rates: 0.0005 for worker optimizer, which is the same rate as the global model, and 0.001 for the chief optimizer. While the

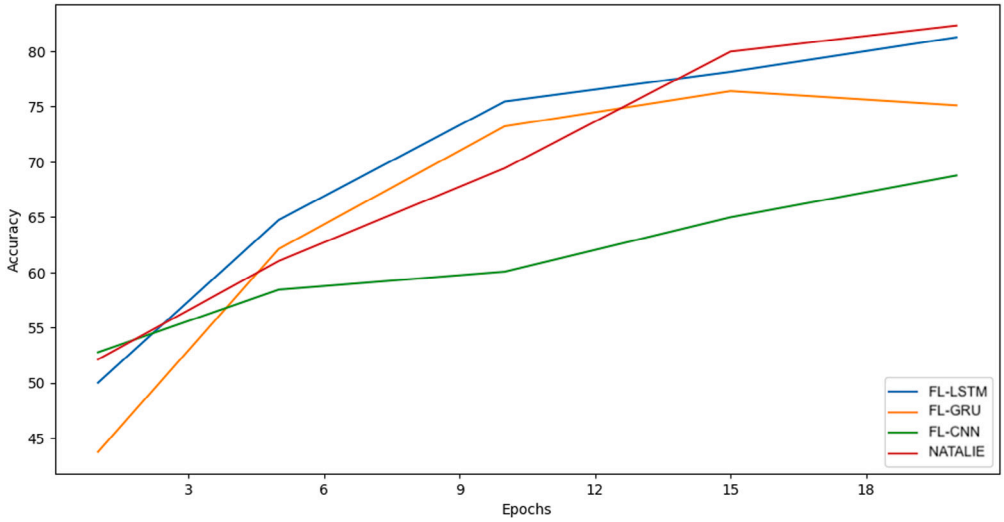


Fig. 9. Prediction accuracy of the baseline and ensemble models.

local batch size is set to 30, each worker trains the proposed framework locally for E (E = 20) epochs using the Adam optimizer. We use categorical cross-entropy as the loss function since mode inference is a multiclass classification problem. Then, we aggregate workers' updates from each device and produce a new global model.

7. Results and analysis

We first discuss the performance of the models using the MTL Trajét dataset. In identical settings, we evaluate the performance of the different base models, namely, federated learning-based LSTM, GRU, and CNN, to find the model with the best performance. For benchmarking purposes, we compare NATALIE to two state-of-the-art models on two different datasets. Detailed sensitivity analysis is performed. We also evaluate the proposed model's robustness and efficacy using i.i.d. and non-i.i.d. data at various sample sizes and its scalability based on various worker numbers.

7.1. Performance comparison with baseline models

In this section, we compare the performance of the proposed NATALIE model with that of the FL-based DNN models: LSTM, GRU and CNN in a similar simulation configuration. Fig. 9 shows the accuracy for the first 20 epochs of all the models. We observe that the LSTM model has a high predictive performance as it outperforms GRU and CNN by reducing the error of the mode of travel and improving prediction accuracy. The GRU and CNN achieve prediction accuracy of 75.1% and 68.75%, respectively, at the 20th epoch. At the same time, the LSTM achieves the highest predictive accuracy of 81.25% at the 20th epoch.

The travel modes in the GPS data change in various ways over time and space, for instance, for walking mode, the change in instantaneous acceleration can be very high, but the duration for which it can be maintained is relatively small. On the other hand, in motorized modes, such as bus and car, the acceleration can be maintained for a longer duration of time resulting in higher space-mean and time-mean accelerations. The variation and magnitude in speed of motorized vehicles may also differ depending on where they are operating (e.g., highway versus roads in downtown) and at what time (e.g., rush hour versus mid-night). The LSTM model is able to consistently perform better over epochs among the three as its architecture is specifically designed to capture the long-term dependencies in sequential data. The memory cell in LSTM with input, output, and forget gates is able to capture useful patterns and correlations in the GPS sequences (e.g., space-mean and time-mean acceleration, their rate of change, and the associated duration it can last) for each mode much better. On the other hand, CNN is designed to capture only the local-level dependencies (e.g., the instantaneous change in acceleration). When used on its own on GPS data, it may miss out on the long-term dependencies that are important to classify different modes. Like LSTM, GRU also has the memory cell, but with only two gates, i.e., reset and update gates. Using this structure, the GRU is able to capture the long-term patterns better than CNN, but this architecture is less robust than the one in LSTM, resulting is a lower performance than LSTM model. It though has lesser parameters and is faster to train than LSTM.

Using ensemble learning, NATALIE combines the benefits of LSTM, CNN, and GRU to achieve the prediction accuracy of 82.31% at epoch 20. The NATALIE architecture ensures that local-level patterns (e.g., instantaneous speed, acceleration, and jerk associated with a mode) are captured through CNN, simpler long-term patterns (e.g., the time-mean speed associated with a mode) are captured by GRU with lesser computational cost, and more complex long-term patterns (e.g., the relationship between magnitude, rate of change, and duration of change of features over sequence for a mode) are captured by LSTM. Note that NATALIE outperforms the LSTM only after the 14th epoch. The ensembling mechanism at the upper level of NATALIE takes a few epochs to learn the best weights for each of the base models and only after the 14th epoch, it seems to have found the better weighting.

Furthermore, to get a finer-grained idea of how well NATALIE and the individual FL models are doing, we evaluate the performance of the models based on the precision, recall, and F-score, see Table 1. The results show that NATALIE performed better than all the individual base classifiers. Specifically, the results show that NATALIE achieves a better result with 83.1% precision compared to the precision of LSTM (82.7%), GRU (75.1%) and CNN (70.4%). The findings reveal that NATALIE has a high prediction capacity to correctly infer users' travel mode and a low ability to incorrectly make predictions compared to LSTM, GRU and CNN. This means that minimizing the incorrectly predicted mode of travel is key to ensuring the reliability of the proposed model. Similarly, NATALIE achieves the best predictive performance for recall compared to the other models. The recall of 85.4% was obviously higher when compared to the recall of LSTM (83.1%), GRU (78.2%) and CNN (72.1%) indicating that users' travel mode can be inferred correctly using NATALIE model compared with the baseline models. Again, the best F1 predictive performance is the NATALIE model with an F1 score of 84.5%, followed by LSTM (82.9%), GRU (77.4%), and CNN (72.4%). Hence, based on the evaluation metrics, our proposed model outperforms all the other baseline models with high predictive performance.

Table 1
Performance measures of FL models.

Model type	Precision	Recall	F1-score
FL-LSTM	82.7	83.1	82.9
FL-GRU	75.1	78.2	77.4
FL-CNN	70.4	72.1	72.4
NATALIE	83.1	85.4	84.5

Table 2
Prediction accuracies of FL and ML models.

Architecture	Model type	Accuracy (%)
FL models	NATALIE	87.31
	FL-CNN	73.05
ML models	Ensemble CNN (Yazdizadeh et al., 2019)	83.09
	CNN	71.56

7.2. Performance comparison with centralized machine learning models

We benchmark the model performance by comparing the proposed model to the centralized ensemble CNN model developed by Yazdizadeh et al. (2019) for mode inference in MTL Trajét dataset. The reason we chose this particular model is that in the existing literature that used MTL Trajét dataset, Yazdizadeh et al. reported the highest prediction accuracy of 92%. However, their model was trained to infer only four modes (walk, bike, car, and bus). So, to be consistent with NATALIE, we implemented and trained their model for all 30 modes available in MTL Trajét dataset. Additionally, we also trained a stand-alone CNN on FL and in a centralized manner, for comparison. We used 200 ephocs for training all the models. Table 2 shows the prediction accuracies of NATALIE, FL-CNN, centralized ensemble CNN, and centralized CNN.

The performance of centralized ensemble CNN in terms of accuracy decreases from reported 92% to 83.09% when the trip modes are increased from four basic modes to 30 more complex modes that also include a combination of modes, e.g., walk and bus. NATALIE outperforms the centralized ensemble CNN, centralized CNN, and FL-CNN models with an accuracy of 87.31%. As mentioned before, the combination of LSTM, GRU, and CNN at workers ensures that both local and long-term patterns in GPS sequences are efficiently captured. At the same time, MLP at chief ensures that optimal weights are assigned to their contributions in the ensemble.

The confidentiality of users' private information in a real-world implementation is the basis for the proposed ensemble model's advantage over the centralized ensemble model. While the centralized ML model demands the upload of GPS data to a centralized location, infringing on users' privacy, the ensemble FL framework protects users' privacy by preventing data from being shared in a centralized platform. NATALIE not only provides higher travel mode predictive performance than the benchmark centralized ML model, but also guarantees that user privacy is preserved.

7.3. Sensitivity analysis

The aim of this analysis is to critically and systematically evaluate the performance of NATALIE at varying sample sizes and client numbers. The outputs are assessed based on the model's prediction accuracy.

7.3.1. Performance comparison with different number of workers

Here, we explore the effect of different numbers of workers on the performance of the proposed model by changing the number of workers for the training of NATALIE model. To evaluate the robustness and stability of NATALIE, a variety of devices are used as workers for the training of the model. Therefore, we set the worker numbers to as high as 100 as the data is enough to distribute to a large number of workers for modelling the FL based neural network.

It is evident from our experimentation that changes in the number of workers have an effect on the model performance. In Table 3, we observe that as the worker numbers remain relatively small, the predictive accuracy is high. Thus, a decrease in the number of workers relatively improves model performance. However, we notice that an expansion of the number of workers results in the oscillation of the performance of NATALIE, between 80.57% and 82.55%, with the highest performance achieved at 100 workers. We hypothesize that this phenomenon is due to the stochastic nature of the simulations used to develop the results. The mean accuracy is 81.91% and the standard deviation is 0.66%, which indicates strong stability in the performance with the change in the number of workers.

We observe that the model performance remained relatively the same when the number of workers increased from 60 to 100. Even though the expansion of the number of workers makes the model converge slowly, causing the prediction accuracy to reduce slightly, the consistent prediction accuracy after 60 workers shows that NATALIE is robust to huge worker numbers. In the real world, we expect large-scale workers to participate in NATALIE with different devices. The proposed model solves the scalability issue by achieving high predictive performance when huge numbers of workers are involved in the FL process. Furthermore, experimental results show the model's robustness to the number of workers, that is, the model performance is not influenced by a large number of participating workers. Thus, NATALIE can maintain good stability, robustness, and efficiency.

7.3.2. Performance comparison of i.i.d. and non-i.i.d. data with varying sample sizes

In this section, we explore the impact of i.i.d. and non-i.i.d. data distribution on the proposed model and compare the performances of the data distributions at varying sample sizes. In this approach, datasets are randomly selected from the original dataset with sizes at a percentage rate of 20, 50, 80 and 100. To evaluate the robustness and stability of NATALIE, the different data samples are trained as inputs to the model. Fig. 10 shows the performance of NATALIE when trained with varying data distributions.

It is evident from the results that the model's performance slightly lowers under non-i.i.d. data distributions when compared to the performance of the model under i.i.d. data distributions. With data distributed at 20% of the original data for both i.i.d. and non-i.i.d., we observe that the i.i.d. data outperforms the non-i.i.d. data with a prediction accuracy of 78.1%. Similarly, i.i.d. data performs better with a prediction accuracy of 84.4% compared to non-i.i.d. data with a prediction accuracy of 84.1% when the percentage of data distribution is 100%. Non-i.i.d. data are expected to perform poorly on FL models due to their slow convergence. The reason for the poor performance of non-i.i.d. could be attributed to the increase in differences in weights between the various participating workers and that makes it difficult for model aggregation and optimization (Zhu, Zhang, Liu, Niyato, & James, 2020). However, NATALIE is designed to be robust and effective against non-i.i.d. data distribution producing better performance. The results show that there is no significant difference in NATALIE performance under i.i.d. and non-i.i.d. data distribution. In summary, the proposed model has been shown to have a good convergence under i.i.d. data distribution. In the case of non-i.i.d. distribution, although the convergence guarantee is somewhat deficient, the experimental effect of it is relatively satisfactory and that makes NATALIE a robust enough model.

7.3.3. Performance under different ensembles configurations

The effect of possible combinations of base models, varying sample sizes, and voting techniques on the accuracy of NATALIE is explored.

a. Base model combinations and sample size. Instead of implementing a single LSTM, GRU, or CNN on worker nodes, two combinations

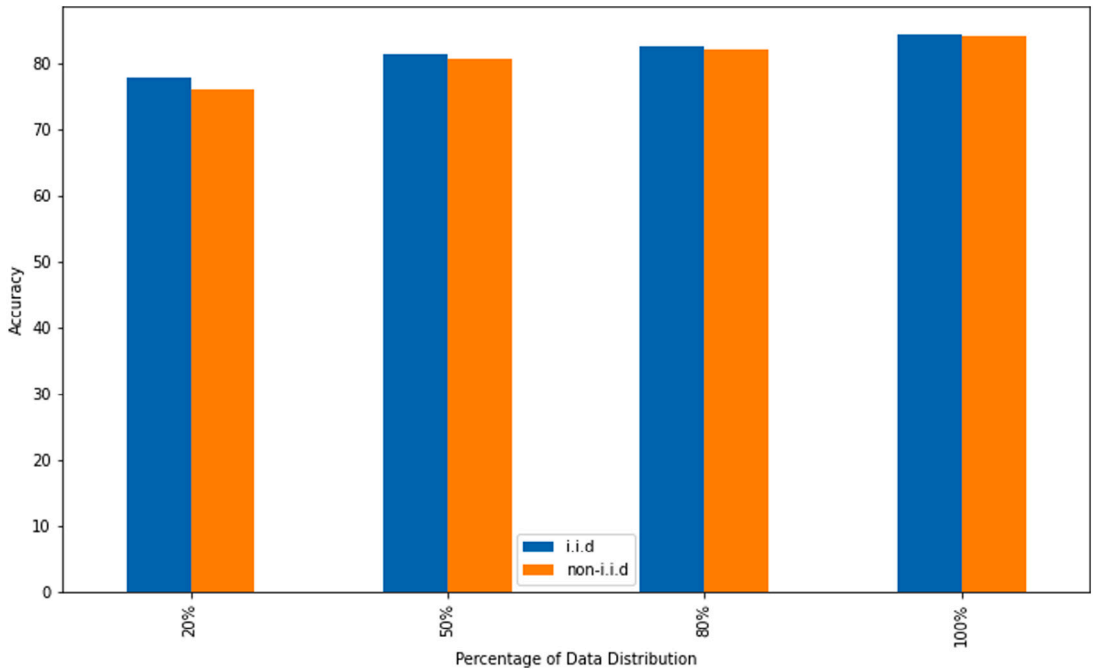


Fig. 10. Model performance between i.i.d. and non-i.i.d. data distribution.

Table 3
Prediction accuracy at different worker numbers.

Number of workers	10	20	30	40	50	60	70	80	90	100
Accuracy	81.25	80.57	82.15	82.34	81.95	82.52	82.15	82.33	81.27	82.55

are explored (see Fig. 7): (i) an LSTM feeding to GRU, and (ii) LSTM feeding to CNN. The rationale here is that LSTM first captures the long-term patterns and then GRU or CNN captures other relevant patterns. The sample size is also changed between 20% to 100% of the training data to evaluate the robustness of the model. The experimental results are shown in Fig. 11. We observe that the combination of LSTM and GRU performs better for all sample sizes than the LSTM and CNN combination. As mentioned before, LSTM and GRU are better suited for sequential data as they can capture the long-term patterns using the memory cell and gates architecture. CNN on the other hand can only capture local patterns that are not enough for classifying 30 different modes in the GPS sequence. The performance increases linearly with the increase in the sample size. However, none of these combinations are able to achieve the performance of the original NATALIE architecture (87.34%), where a worker implements an LSTM, GRU, or CNN and works with a meta-learner at the chief node to train in a distributed manner (see Fig. 11).

b. Voting technique. Furthermore, we analyze the performance of the two combinations with ensemble voting techniques, using accuracy, precision, recall and F1 score. Table 4 shows the model performance of LSTM-CNN and LSTM-GRU ensemble models. For LSTM-CNN ensemble model, the majority voting technique achieves the best performance with a precision of 84.8% compared to a precision of 83.9% and 83.6% for average voting and weighted voting, respectively. The results of recall also perform better for the majority voting technique when compared to average voting and weighted voting. In the case of F1 score, weighted voting performs slightly better. For LSTM-GRU model, weighted voting obtains the best predictive performance with a precision of 84.2% compared to a precision of 83.5% and 83.6% for majority voting and average voting, respectively. The results of recall also perform better for the weighted voting technique when compared to average voting and majority voting. In the case of F1 score, majority voting performs slightly better. LSTM-CNN ensemble model achieves better predictive performance for all the voting techniques when compared to LSTM-GRU ensemble model. These results show that all three voting techniques are robust and perform very closely to each other in terms of different performance indicators; therefore, they can be used interchangeably here. The voting techniques have proven effective for a set of comparable performing DNN models by enhancing their individual weak classifiers to obtain an ensemble model with high predictive performance.

Table 4
Predictive performance of different ensemble technique.

LSTM-CNN ensemble				
Voting technique	Accuracy	Precision	Recall	F1
Average voting	80.5	83.9	83.2	84.72
Majority voting	81.2	84.8	85.2	84.56
Weighted voting	81.7	83.6	84.0	85.1
LSTM-GRU ensemble				
Voting technique	Accuracy	Precision	Recall	F1
Average voting	82.6	83.6	84.2	83.3
Majority voting	82.3	83.5	82.3	84.4
Weighted voting	81.5	84.2	84.5	82.2

7.4. Performance on Geolife dataset

In addition to MTL Trajét dataset, we also train and test the performance of NATALIE on Geolife dataset (Zheng et al., 2011). Collected in five different cities in China from 2007 to 2012, it contains 2.5 million GPS points, representing 14,718 labelled trips from 69 individuals. The modes included walk, bike, bus, car, subway, train, airplane, boat, run, and motorcycle. To prepare this dataset, we use the same steps as the ones used on MTL Trajét dataset and described in Section 5.2. Compared to MTL Trajét dataset, it is a relatively small dataset, however, it is the most used open-access dataset in the literature.

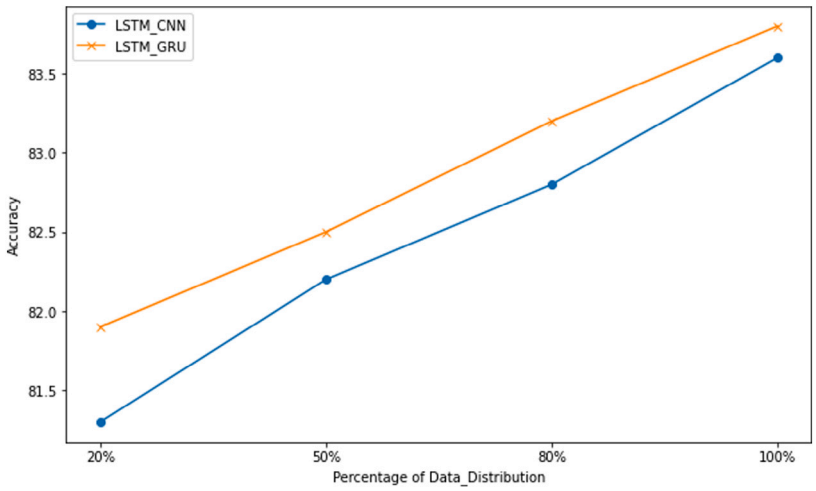


Fig. 11. Model performance at different ensemble FL approach.

Table 5
Prediction accuracies of FL and ML models.

Architecture	Model type	Accuracy (%)
FL models	NATALIE	87.25
	CNN	82.3
ML models	Ensemble CNN Dabiri and Heaslip (2018)	85.20

Dabiri and Heaslip (2018) that is considered a benchmark on the Geolife dataset, used an ensemble of CNNs for inferring a subset of available modes, i.e., walk, bike, bus, driving, and train. Here, we reimplement their proposed architecture to train and test for all the labelled modes in Geolife dataset. Table 5 shows the performance of NATALIE, centralized ensemble CNN, and federated CNN in terms of accuracy. The accuracy of the centralized ensemble CNN improves slightly from the original implementation of Dabiri and Heaslip (2018) that reported an accuracy of 84.8%. This slight improvement may be attributed to the dedicated hyperparameter tuning process (constrained random search) and longer epochs (200) used in our implementation. Their implementation took the hyperparameters from other models and used 62 ephocs to train the ensemble CNN model. The difference in the test sample may also be attributed to this difference. An implementation of CNN on the federated learning framework resulted in an accuracy of 82.3%. It is higher than the accuracy of 75.6% reported by Dabiri and Heaslip (2018) for a centralized CNN model. The accuracy of NATALIE was 87.25%, outperforming both centralized ensemble CNN and federated CNN. We attribute this performance gain to the robust architecture of the NATALIE, which is able to model both the local and long-term sequential relationship in the GPS data better. Furthermore, the ensemble structure ensures that the heterogeneity in the data is given specific attention by the best-suited individual model.

8. Implications for smart and sustainable cities and society

Based on the detailed experiments conducted in the previous section, using the MTL Trajét and GeoLife datasets, we discuss the implications of the adoption of NATALIE framework adoption for smart and sustainable cities and society.

8.1. Preserving privacy while accessing user-sensitive information

The proposed framework addresses issues related to information-based privacy by ensuring that travellers' sensitive information is not shared with third parties. In a real-world implementation, we anticipate that NATALIE for personal mobility inference is effectively deployed by using a large pool of smart devices that are periodically active. Since FL is concerned with travellers' privacy and sharing of private information to third parties, the proposed model can address the issues. As a distributed learning paradigm for privacy protection, NATALIE model attains travel mode inference by securely aggregating model parameters instead of accessing the real data of travellers to ensure privacy protection of user-sensitive data.

8.2. Public participantion

It is expected that NATALIE framework will be used by smart and sustainable cities to support the decision making processes in terms of planning, design, and operations of transportation infrastructure. Particularly, policies, regulations, and infrastructure investments will be the result of evidence provided by the NATALIE trained models. The privacy-by-design approach of NATALIE will attract greater confidence and interest from the citizen in these processes. They are expected to be much more willing to participate in the "democratic" development of decisions by voicing their priorities anonymously via their data's contribution to the model training process. Associated mechanisms like blockchain can then be used by citizens to track how their data contributed to the development process of certain policies, regulations, or investments (Lopez & Farooq, 2020).

8.3. High predictive performance

Empirical evidence from the previous subsections shows that NATALIE outperforms the state-of-the-art models proposed on both datasets used in the analysis. The ensemble CNN is a centralized machine learning model, which requires direct access to large amounts of raw GPS data that are aggregated to achieve high-precision travel mode inference. The proposed model differs from the centralized ML approach and learns the model via a secure parameter aggregation instead of directly accessing data from users. Moreover, NATALIE is an ensemble model which demonstrates its superiority by achieving high predictive performance over the single base FL classifiers. The reason is that combination of outputs of individual classifiers can minimize generalization errors and deal with high variance of the classifiers. Previous approaches such as discrete choice models and centralized ML algorithms for travel mode inference can achieve high predictive accuracy (Dabiri & Heaslip, 2018; Dalumpines & Scott, 2017; Yazdizadeh et al., 2019). However, travellers are required to share their

personal information for the implementation of the models. Hence, our proposed model addresses the trade-off between travel mode inference and privacy by ensuring high predictive performance while protecting user-sensitive information.

8.4. Robustness of the proposed model

We show that NATALIE is able to train and perform well on multiple datasets. Based on the experimental results, the proposed model demonstrates robustness and effectiveness to varying sample sizes and different worker numbers. While the performance oscillates to some extent with the change of worker numbers in the FL training process, we observe a minimal difference in performance. We expect that in practice, NATALIE will be training models using tens of thousands of workers. On the other hand, more input data for the FL training achieve high predictive performance compared to few input data for the FL training process. Furthermore, the proposed model accounts for issues arising from statistical heterogeneity. As the data generated from different smartphone GPS-enabled devices become more heterogeneous, the convergence of the training process slows down. Furthermore, it also causes instability in the training process. To handle the divergence and instability situation, the proposed model has been designed to ensure better performance of non-i.i.d. data to prevent the complexity of modelling and analysis. Not only does the proposed model handle the issue of statistical heterogeneity, but it ensures that global model updates from the chief are able to synchronize as some of the workers could drop out of the FL training process. Our proposed mode solves this issue by employing a random sampling technique for the selection of participant workers for the FL training process. The idea of randomly selecting participant workers ensures the functioning of the workers, thus ensuring syncing of global model updates.

9. Limitations

Even though the proposed model demonstrates some advantages over the centralized machine learning model, there are some challenges associated with its implementation. Based on empirical observations, the challenges of the proposed model can be discussed as follows:

9.1. Statistical heterogeneity

The statistical heterogeneity makes it challenging to analyze the convergence behaviour of the global models trained by NATALIE, as non-i.i.d. data across devices in the network would tend to diverge. Participants generate and collect data across the network with different devices in a completely different way, possibly increasing the complexity of the model and its evaluation (Li, Sahu, Talwalkar, & Smith, 2020). Convergence assurances are difficult to resolve when there are heterogeneous data and several constraints, even with k-fold validation. Therefore, techniques like bootstrapping must be employed to develop a confidence interval over the performance.

9.2. Scalability

In practice, NATALIE is expected to train the model with tens of thousands of workers. Convergence challenges are expected to be encountered when expanding to such a scale. This may result in a reduction in model performance. The heterogeneity of users could further exasperate the issue. NATALIE has shown stable performance for hundreds of users. However, a comprehensive analysis is required with tens of thousands of users to ensure robustness at such a scale. This will require access to very high computational power, which is not currently available to the authors.

9.3. Equity, diversity, and inclusion

In order to participate in the training process, the worker device must evaluate the potential model weights on their local data and return the gradient. This process can be computationally very expensive, especially for training models like LSTM. However, not every citizen will have access to a personal device that can provide the necessary computational power in order to participate in the FL training process. Thus, the resulting equity, diversity, and inclusion issues must be studied in more detail and innovative solution to overcome computational issues must be developed.

9.4. Cybersecurity and data poisoning

Even though cloud-access authentication has become extremely important in enabling public safety services to collect sensitive information about the city and citizens, there are concerns about security threats in smart cities that depend on digitally connected infrastructure (Habibzadeh, Nussbaum, Anjomshoa, Kantarci, & Soyata, 2019). The federated learning process is susceptible to data poisoning attacks from malicious entities that can result in the training process not converging or converging to a local optima desired by the malicious actors (Al Mallah, Badu-Marfo et al., 2021). To protect NATALIE against such attacks, cybersecurity threat detection, monitoring, and defence techniques designed for FL, such as Al Mallah, Lopez, Marfo and Farooq (2021) must be employed when deployed.

9.5. Freight modes

In a smart and sustainable city, it is important that all forms of transportation modes are given equal consideration. However, most of the current mode inference models, including NATALIE, are primarily focused on passenger modes. Unfortunately, modes related to freight/goods movement have historically been ignored, mainly due to the unavailability of open-access and labelled freight GPS data. The situation is becoming further complicated by the rise of on-demand services that use walk, bikes, cars, e-scooters, drones, and sidewalk robots for last-mile goods delivery. Therefore, there is a strong need to collect and label more inclusive data and develop inference models that include freight modes. This will also require the collection of freight mode specific features, for instance, land-use at origin and destination and frequency of trips by an individual from a certain origin.

9.6. Other motorized passenger modes

In many developing countries, modes like motorcycles and moped are important means of transportation. The current study does not have any labelled data for such modes, due to which such modes were not part of the modelling process. In the future, efforts must be made to include such modes. A dedicated data collection campaign in developing countries will be necessary to achieve incorporate such specific modes.

10. Conclusion

We propose an eNsemble federATed leArning for mobiLity InfErence (NATALIE) framework that can infer every human mobility behaviour. Meanwhile, this study focuses on travel mode inference based on NATALIE framework. Three neural networks that include LSTM, GRU, and CNN are used as base learners, whereas MLP is used as the meta learner for the stacking ensemble method. The model uses privacy mechanisms to train a global model in a distributed manner rather than allowing direct access to the user data. We constructed a set of Deep Neural Networks augmented by the stacking ensemble technique utilizing the FL framework to infer trip mode from GPS trajectories obtained from a large-scale smartphone travel survey. The FL approach involves

aggregating the local model updates uploaded by all the locally trained models from the workers to the chief to build a global one for travel mode inference. The datasets of GPS trajectories are handled in such a way that they may be given as an input layer to a series of DNNs in privacy-preserving manner. Each trip has fixed segments with four channels that include relative distance, speed, acceleration, and jerk rate.

With the stacking ensemble method, we combined the results of the LSTM, GRU, and CNN to obtain better prediction accuracy than the baseline models. Our ensemble library included several models with various hyper-parameter values and architectures. On the MTL trajét open-access dataset, we evaluated the performance of NATALIE and compared it to the vanilla FL and non-federated learning methods. The findings demonstrate that the proposed ensemble technique outperforms the baseline models. Among the three neural networks of the vanilla neural network-based FL, the LSTM model has the best accuracy compared with GRU and CNN. Besides, we evaluated the robustness and effectiveness of NATALIE to non-i.i.d. data distributions while training the model at varying sample sizes and under different worker numbers. The experimental results show the robustness of our proposed model since both data distributions show results that are comparable to each other. In summary, we have been able to infer mode of travellers while ensuring non-violation of their privacy. For a sustainable society, information of users on travel behaviour are required for transportation planning and policy-making. While attempts are made to get the information, the federated learning approach provides a solution to address the privacy of user-sensitive information. Hence, our proposed model offers a solution that can guide transport planners to achieve this purpose.

Declaration of competing interest

The authors declare that there are no competing interest.

Data availability

The two datasets used in the study, i.e., Montréal Trajet and GeoLife, are publically available datasets.

Acknowledgments

This study was funded by the Canada Research Chair Program (award id: 2017-00038) and the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Program (award id: RGPIN-2020-04492).

References

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M., & Inman, D. J. (2017). Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *Journal of Sound and Vibration*, *388*, 154–170.

Al Mallah, R., Badu-Marfo, G., & Farooq, B. (2021). Cybersecurity threats in connected and automated vehicles based federated learning systems. In *2021 IEEE intelligent vehicles symposium workshops (IV Workshops)* (pp. 13–18). IEEE.

Al Mallah, R., Lopez, D., Marfo, G. B., & Farooq, B. (2021). Untargeted poisoning attack detection in federated learning via behavior attestation. arXiv preprint arXiv:2101.10904.

Bagdadi, O., & Várhelyi, A. (2013). Development of a method for detecting jerks in safety critical events. *Accident Analysis and Prevention*, *50*, 83–91.

Bolbol, A., Cheng, T., Tsapakis, I., & Haworth, J. (2012). Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *Computers, Environment and Urban Systems*, *36*(6), 526–537.

Byon, Y.-J., & Liang, S. (2014). Real-time transportation mode detection using smartphones and artificial neural networks: Performance comparisons between smartphones and conventional global positioning system sensors. *Journal of Intelligent Transportation Systems*, *18*(3), 264–272.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Claici, S., Yurochkin, M., Ghosh, S., & Solomon, J. (2020). Model fusion with Kullback-Leibler divergence. In *International conference on machine learning* (pp. 2038–2047). PMLR.

Cottrill, C. D., Jacobs, N., Markovic, M., & Edwards, P. (2020). Sensing the city: Designing for privacy and trust in the internet of things. *Sustainable Cities and Society*, *63*, Article 102453.

Dabiri, S., & Heaslip, K. (2018). Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C (Emerging Technologies)*, *86*, 360–371.

Dalumpines, R., & Scott, D. M. (2017). Making mode detection transferable: extracting activity and travel episodes from GPS data using the multinomial logit model and Python. *Transportation Planning and Technology*, *40*(5), 523–539.

Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., et al. (2012). Large scale distributed deep networks. *Advances in Neural Information Processing Systems*.

Dostál, R., Přibyl, O., & Svítek, M. (2020). City infrastructure evaluation using urban simulation tools. In *2020 Smart city symposium prague (SCSP)* (pp. 1–6). IEEE.

Fahad, S. A., & Alam, M. M. (2016). A modified K-means algorithm for big data clustering. *International Journal of Science, Engineering and Computer Technology*, *6*(4), 129.

Feng, T., & Timmermans, H. J. (2016). Comparison of advanced imputation algorithms for detection of transportation mode and activity episode using GPS data. *Transportation Planning and Technology*, *39*(2), 180–194.

Fiosina, J. (2021). Explainable federated learning for taxi travel time prediction. In *VEHITS* (pp. 670–677).

Gonzalez, P., Weinstein, J., Barbeau, S., Labrador, M., Winters, P., Georggi, N. L., et al. (2008). Automating mode detection using neural networks and assisted GPS data collected using GPS-enabled mobile phones. In *15th World congress on intelligent transportation systems* (pp. 16–20). Citeseer.

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Habibzadeh, H., Nussbaum, B. H., Anjomshoa, F., Kantarci, B., & Soyata, T. (2019). A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities. *Sustainable Cities and Society*, *50*, Article 101660.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

James, J. (2020a). Semi-supervised deep ensemble learning for travel mode identification. *Transportation Research Part C (Emerging Technologies)*, *112*, 120–135.

James, J. (2020b). Travel mode identification with GPS trajectories using wavelet transform and deep learning. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 1093–1103.

Kalatian, A., & Farooq, B. (2021). A context-aware pedestrian trajectory prediction framework for automated vehicles. arXiv preprint arXiv:2104.08123.

Kim, J., Kim, J. H., & Lee, G. (2022). GPS data-based mobility mode inference model using long-term recurrent convolutional networks. *Transportation Research Part C (Emerging Technologies)*, *135*, Article 103523.

Kweon, Y., Sun, B., & Park, B. B. (2021). Preserving privacy with federated learning in route choice behavior modeling. *Transportation Research Record*, Article 03611981211011162.

Lari, Z. A., & Golroo, A. (2015). Automated transportation mode detection using smart phone applications via machine learning: Case study mega city of Tehran. In *Proceedings of the transportation research board 94th annual meeting, Washington, DC, USA* (pp. 11–15).

Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2019). On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189.

Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, *37*(3), 50–60.

Liu, Y., James, J., Kang, J., Niyato, D., & Zhang, S. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, *7*(8), 7751–7763.

Liu, Y., Kang, Y., Xing, C., Chen, T., & Yang, Q. (2020). A secure federated transfer learning framework. *IEEE Intelligent Systems*, *35*(4), 70–82.

Lopez, D., & Farooq, B. (2020). A multi-layered blockchain framework for smart mobility data-markets. *Transportation Research Part C (Emerging Technologies)*, *111*, 588–615.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

Mensah, D. O., Badu-Marfo, G., Al Mallah, R., & Farooq, B. (2022). eFedDNN: Ensemble based federated deep neural networks for trajectory mode inference. In *2022 IEEE international smart cities conference (ISC2)* (pp. 1–7). IEEE.

Montréal (2018). Déplacements MTL trajet - site web des données ouvertes de la ville de montréal. https://donnees.montreal.ca/dataset/mtl-trajet, (Accessed on 04/24/2023).

Moudra, K., Matowicki, M., Přibyl, O., & Foltýnová, H. B. (2019). Potential of a travel mode change in smart cities: a review. In *2019 Smart city symposium prague (SCSP)* (pp. 1–7). IEEE.

Nitsche, P., Widhalm, P., Breuss, S., Brändle, N., & Maurer, P. (2014). Supporting large-scale travel surveys with smartphones–A practical approach. *Transportation Research Part C (Emerging Technologies)*, *43*, 212–221.

Patil, V., Parikh, S. B., & Atrey, P. K. (2019). GeoSecure-O: A method for secure distance calculation for travel mode detection using outsourced gps trajectory data. In *2019 IEEE fifth international conference on multimedia big data (BigMM)* (pp. 348–356). IEEE.

Peel, K., & Tretter, E. (2019). Waterfront toronto: Privacy or piracy?.

Prelipcean, A. C., Gidófalvi, G., & Susilo, Y. O. (2017). Transportation mode detection–an in-depth review of applicability and reliability. *Transport Reviews*, *37*(4), 442–464.

Ramu, S. P., Boopalan, P., Pham, Q.-V., Maddikunta, P. K. R., Huynh-The, T., Alazab, M., et al. (2022). Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustainable Cities and Society*, *79*, Article 103663.

Ran, X., Shan, Z., Fang, Y., & Lin, C. (2019). An LSTM-based method with attention mechanism for travel time prediction. *Sensors*, *19*(4), 861.

Sharma, P. K., Park, J. H., & Cho, K. (2020). Blockchain and federated learning-based distributed computing defence framework for sustainable society. *Sustainable Cities and Society*, [ISSN: 2210-6707] *59*, Article 102220. http://dx.doi.org/10.1016/j.scs.2020.102220, URL https://www.sciencedirect.com/science/article/pii/S2210670720302079.

Stenneth, L., & Phillip, S. Y. (2010). Global privacy and transportation mode homogeneity anonymization in location based mobile systems with continuous queries. In *6th International conference on collaborative computing: Networking, applications and worksharing (CollaborateCom 2010)* (pp. 1–10). IEEE.

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 54–63).

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, *23*(176), 88–93.

Xiao, G., Juan, Z., & Gao, J. (2015). Travel mode detection based on neural networks and particle swarm optimization. *Information*, *6*(3), 522–535.

Yang, F., Yao, Z., & Jin, P. J. (2015). GPS and acceleration data in multimode trip data recognition based on wavelet transform modulus maximum algorithm. *Transportation Research Record*, *2526*(1), 90–98.

Yazdizadeh, A., Patterson, Z., & Farooq, B. (2019). Ensemble convolutional neural networks for mode inference in smartphone travel survey. *IEEE Transactions on Intelligent Transportation Systems*, *21*(6), 2232–2239.

Yazdizadeh, A., Patterson, Z., & Farooq, B. (2021). Semi-supervised GANs to infer travel modes in GPS trajectories. *Journal of Big Data Analytics in Transportation*, *3*, 201–211.

Yue, Y., Lan, T., Yeh, A. G., & Li, Q.-Q. (2014). Zooming into individuals to understand the collective: A review of trajectory-based travel behaviour studies. *Travel Behaviour and Society*, *1*(2), 69–78.

Zheng, Y., Fu, H., Xie, X., Ma, W.-Y., & Li, Q. (2011). Geolife GPS trajectory dataset - user guide. In *Geolife GPS trajectories* (1.1 ed.). URL https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/.

Zheng, Y., Liu, L., Wang, L., & Xie, X. (2008). Learning transportation mode from raw gps data for geographic applications on the web. In *Proceedings of the 17th international conference on world wide web* (pp. 247–256).

Zhu, Y., Zhang, S., Liu, Y., Niyato, D., & James, J. (2020). Robust federated learning approach for travel mode identification from non-IID gps trajectories. In *2020 IEEE 26th International conference on parallel and distributed systems (ICPADS)* (pp. 585–592). IEEE.