Power-saving transportation mode identification for large-scale applications

Yuren Zhou*, Wang Jin†, Shi Peng‡, Daniel Dahlmeier‡, Nils Tippenhauer*, Erik Wilhelm*
*Singapore University of Technology and Design, 8 Sompah Road, 487372 Singapore
Email: yuren_zhou@nmail.sutd.edu.sg; {nils_tippenhauer, erikwilhelm}@sutd.edu.sg
†Zhejiang University, 38 Zheda Road, Xihu, Hangzhou, 310027 China
Email: impavidity@zju.edu.cn
‡SAP, 1 Create Way, #14-01 & 02, 138602 Singapore
Email: {jin.wang02, d.dahlmeier}@sap.com

Abstract—Transportation mode detection with personal devices has been investigated for over ten years due to its importance in monitoring people's activities, understanding human mobility, and assisting traffic management. However, two main limitations are still preventing it from large-scale deployments: high power consumption, and the lack of high-volume and diverse labeled data. In order to reduce power consumption, existing approaches are sampling using fewer sensors and with lower frequency, which however lead to a lower accuracy. A common way to obtain labeled data is recording the ground truth while collecting data, but such method cannot apply to large-scale deployment due to its inefficiency. To address these issues, we adopt a new low-frequency sampling manner with a hierarchical transportation mode identification algorithm and propose an offline data labeling approach with its manual and automatic implementations. Through a real-world large-scale experiment and comparison with related works, our sampling manner and algorithm are proved to consume much less energy while achieving a competitive accuracy around 85%. The new offline data labeling approach is also validated to be efficient and effective in providing ground truth for model training and testing.

I. INTRODUCTION

Identifying individuals’ transportation modes, such as stationary, running, walking, driving, riding metro, riding buses, using wearable sensors has been studied for more than ten years [1]. Researchers have shown its great advantages in many applications, such as monitoring one’s daily physical activities [2], estimating personal CO2 emission [3], conducting large-scale transportation surveys more efficiently and accurately [4, 5].

Same as other classification tasks, high accuracy of transportation mode identification is the focus of most of the previous works. However, while the accuracy has reached a very high level, there are still two issues which present itself from practical large-scale applications like transportation surveys. The first issue is large power consumption. Mode identification with high accuracy requires several types of sensors and high sampling frequencies, and thus draws a lot of power from the host device. As a result, battery life of the host device can be shortened dramatically, which makes it either unacceptable for users or unrealistic for long-duration applications.

The second and not well discussed issue is the difficulty in obtaining large and diverse data sets with accurate ground truth labels (true transportation modes). Such data are called labeled data and are crucial for model training and testing. The data sets used in mode identification research are people’s commuting data with location and Inertia Measurement Units (IMU) values. Their size refers to length of duration or distance of the data. Their diversity refers to the number of different data collectors, because same person tends to travel in a certain pattern everyday. A large and diverse training data set prevents over-fitting problem, while a large and diverse testing data set makes the model evaluation more convincing. If the training and testing data sets are collected by the same group of people, there is little variance between the two data sets. As a result, the accuracy can be higher than its real level, i.e., the accuracy is often overestimated [6]. Table VII lists the data size and number of data collectors of related work. It is clear that even though some of them do have a large set of training and testing data, they all suffer from a limited number of data collectors.

The bottleneck of increasing number of data collectors is the typical way of obtaining ground truth label (referred as data labeling in the following), in which data collectors need to report their true transportation modes through either questionnaires or smart phone APPs. This approach requires specific training to the collectors and tedious post-processing procedures, thus consuming lots of time and energy. It also introduces errors to labels as people cannot focus all the time during their commuting trips.

In this paper, we adopt a low-frequency sampling manner and develop a corresponding mode identification algorithm to achieve much lower power consumption compared with the state-of-art works. Three types of sensors are used, namely accelerometer, magnetometer and Wi-Fi scanner. The device samples the sensors and performs a local feature extraction process every 15 seconds. A hierarchical structure and three separate classifiers form the mode identification algorithm. The proposed manner and algorithm can be implemented in many platforms, including smartphones, smartwatches, and any other portable devices with necessary sensors.

Another contribution of our work is that we propose an offline data labeling approach and implement it in both manual and automatic versions. “offline” here means that we don’t
need the data collectors to record or report their true modes during or after the data collection. This approach allows us, the researchers, to label the collected data partially without any information from the collectors, which releases the data labeling workload and enables large-scale data collection of thousands of participants.

To demonstrate the effectiveness of our approaches, we implement them in a real world large-scale application. The results are then compared with other works. It is validated that our sampling manner and mode identification algorithm consume much less energy while maintaining a good level of accuracy, and meanwhile, our offline labeling approach can efficiently help provide large and diverse data sets for training and testing.

II. RELATED WORK

Related works trying to solve the above two issues can be summarized as four types. The first three are aimed at reducing power consumption of mode identification while the last one tries to deal with the conventional tedious and error-prone data labeling method.

Reducing the types of sensors used. Due to high power consumption of GPS, [7], [8] only use accelerometer, but the results show their weakness in differentiating different some vehicle types. Meanwhile, [9], [10] conduct mode identification using only GPS data. Their accuracy is unfortunately not well balanced across different classes. Also, since GPS doesn’t work indoor and underground, they cannot detect indoor walking and metro trips. Other than accelerometer and GPS, [11] tries to use barometer only. It surely saves more energy but it does not differentiate vehicle types and has a low accuracy in identifying walking. Therefore, if we want to identify transportation modes in a fine precision with well balanced and high accuracy, both of IMU and location sensor are necessary.

Using alternative location sensors. Because GPS consumes very high power, its alternatives have been tried in mode identification. [12] senses the geographical movement of user by extracting features from serving cell tower ID and received signal strength. The resulted overall accuracy is above 85%, but it only identifies three modes: stationary, walking and driving. [13] replaces GPS with Android’s Network Location Provider which localizes the device using cell tower and Wi-Fi signals. Their method can identify vehicle types and achieves an overall accuracy above 80%. This work proves the success of GPS alternatives in mode identification.

Decreasing sampling frequency. Sampling less frequently can also help reducing power consumption. In Table VII related work with their sampling frequency and performance are listed. [14] adopts the lowest sampling frequency, every 1 minute, but it cannot identify vehicle types. Among those which identify more modes and achieve an accuracy above 80%, 1 Hz is the lowest sampling frequency.

Adopting semi-supervised or unsupervised learning. There are only a few works trying to deal with the difficulties in obtaining large-quantity and diverse labeled data set for identifying travel modes. [15] proposes a semi-supervised learning approach to deal with this issue, in which they apply Virtual Evidence mechanism to fill missing labels in partially labeled data sets. This approach only requires data collectors to label each mode chunk once at a random moment and is showed to produce good accuracy. However, engagement of collectors is still needed. [16] tries a completely unsupervised learning approach to skip the data labeling step, but unfortunately didn’t report very good performance.

Compared with previous works, the proposed power-saving solution achieves a good balance between consumption and accuracy. We replace GPS with Wi-Fi based localization and further reduce the sampling frequency to every 15 seconds. Although the data quality is restricted, the specifically designed algorithm helps to maintain mode identification accuracy on a competitive level.

Meanwhile, unlike semi-supervised and unsupervised learning cases, our offline data labeling approach release the involvement of data collectors while still being able to provide useful labels. With a small amount of manual engagement from researchers, it can provide full-scale labeled data sets for training and testing models.

III. DATA DETAILS

A. Background

This work is motivated by a nationwide project “National Science Experiment” (NSE) in which over 50,000 students from Primary, Secondary and Junior Colleges were volunteered to each carry a specially designed device equipped with various sensors, as shown in Figure 1a and 1b, for one week in 2015 and 2016. During the week, these devices keep sensing and uploading part of their data to the server when being connected to specific hotspots equipped in the participants’ schools. The project was initially designed as a participatory experiment for youth to experience crowd-sensing techniques, and data from the project after anonymization is open to interesting research questions. Given such a large spatial-temporal data set with Inertia Measurement Units (IMU), identifying the transportation modes of participants is the basic task before doing higher applications such as understanding mobility patterns and evaluating city traffic. On the way to finish this basic task, this work was developed to overcome the data quality limitations and lack of ground truth labels in NSE, but it can also be extended to other platforms like smartphones.

B. Power-Saving Sampling Manner

The NSE device is designed separately before this work, whose details can be found in [17]. Its main design goal is to have a low-cost and power-saving device for a one-week crowd-sensing experiment without charging. To optimize the cost and power consumption, Wi-Fi based localization is used to replace GPS and a low-energy sampling manner is designed through trials and errors.
As shown in Figure 1c, the sampling manner works with a 13-second duty cycle. First part of the cycle is an idle waiting period which significantly extends the lifetime in the field. If there is movement detected from 9-axis IMUs, the waiting period lasts only 10 seconds, otherwise its maximum duration can be one hour. After idle waiting, the device spends two to three seconds scanning surrounding Wi-Fi hotspots and checking uploading availability. Finally, it uses the last one second for data collection and feature extraction. Within this second, the environment sensors are sampled once while the IMUs are sampled 100 times, because motion varies much faster than environment. To minimize the size of uploaded data, we don’t upload the raw data of 9-axis IMUs, namely 3-axis accelerometer, magnetometer and gyroscope. Instead, statistical features calculated from them are uploaded together with the environmental readings. The raw data and features used for mode classification will be elaborated in next section. To make use of NSE data, we adopt this preset sampling manner of the device and develop the corresponding mode identification algorithm. However, depending on different applications, the sampling manner can be adjusted accordingly, and the developed algorithm will still be suitable.

C. Collected Data and Features

As shown in Figure 1c, the NSE device is equipped with one Wi-Fi module, several environment sensors and 9-axis IMUs. Data it uploads to the server include:

- Environment sensor readings sampled every 13 seconds, including light intensity, sound pressure, relative humidity, ambient temperature, pressure, and infrared temperature;
- Statistical features from one-second raw IMU data extracted every 13 seconds, including the standard deviation of magnitude of accelerometer readings, maximum of magnitude of accelerometer readings, mean of magnitude of accelerometer readings in y-axis, the standard deviation of magnitude of gyroscope readings, mean of magnitude of gyroscope readings, and mean of magnitude of magnetometer readings.

Once the data is uploaded to the server, the server calls the Skyhook API to convert the Wi-Fi hotspots information to estimated latitude and longitude.

To make our work extendable to other applications, we limit the data types used in our algorithm. Table I lists the types of data used in the following sections. All these values are sampled every 13 seconds during the data acquisition part of each working cycle.

| Data Types  | Descriptions                                      |
|-------------|---------------------------------------------------|
| TS          | UNIX time stamp recorded each time the device     |
| LAT         | Latitude returned by Skyhook API using hotspots   |
| LON         | Longitude returned by Skyhook API using hotspots  |
| STD_ACC     | Standard deviation of 100 samples of 3-axis       |
| MEAN_MAG    | Mean of 100 samples of 3-axis magnetometer        |

IV. IDENTIFICATION ALGORITHM

A. System Overview

Five different modes are identified in our work: stationary, walking, riding a metro, riding a bus, and riding a car. While the Wi-Fi based localization and low sampling frequency help to save a lot of power consumption, they also cause poor data quality, namely data sparsity, inaccurate and incomplete localization. To overcome these issues, the mode identification algorithm adopts a hierarchical structure which consists of three separate classifiers as shown in Figure 2.

After necessary data cleaning and feature calculation process, more features will be obtained for each working cycle, which are listed in Table II. Together with variables in Table I, they form up one data sample standing for one working cycle of the device. These data samples are then classified one by one into vehicle samples and non-vehicle samples by a sample-wise vehicle-or-not classifier. Vehicle samples are those which belong to vehicle trips, while non-vehicle samples are those which can either be walking or stationary. A heuristic smoothing procedure is then applied to the classified samples to further improve the classification accuracy. After
smoothing, vehicle samples and non-vehicle samples are processed separately. Adjacent vehicle samples are accumulated into different vehicle segments, and each of these segments is further classified into different vehicle types by a segment-wise classifier. The samples belonging to the segment are then assigned with the same vehicle mode as the segment. One the other hand, non-vehicle samples are classified into either walking or stationary individually by a sample-wise walking-or-not classifier.

The key idea behind this overall structure is to conduct vehicle type classification on segments instead of individual data samples. Traditional approaches classify data sample into different modes one by one and then smooth the results. In our case, due to the data sparsity and poor-quality localization, performing vehicle type classification on each sample directly is very hard. On the other hand, if combining vehicle data samples into an entire vehicle segment, high-level features such as average velocity and total travel distance can then be extracted and used to perform a much easier vehicle type classification on segments.

B. Vehicle Segment Detection

As the first part of the system, catching vehicle segments accurately is the basis to achieve a good overall identification accuracy. Therefore vehicle segment detection is conducted by two steps: a) Classifying each data sample using the sample-wise vehicle-or-not classifier, b) Smoothing the classified data sample sequence to remove unrealistically short vehicle or non-vehicle segments.

The sample-wise vehicle-or-not classifier is an Adaptive Boosting (AdaBoost) classifier which consists of 30 decision tree classifiers. The features used to train this classifier are shown in Table II. According to the usage ratios of each feature inside the weak classifiers (decision trees in our case), AdaBoost training algorithm is able to tell importance ratios of all involved features. The ratios are added up to one.

As shown by the importance ratios values in Table II, MOV_AVE VELOCITY is the most effective feature as expected because people travel faster in vehicles. The other four features share the rest importance evenly. STD_ACC is a indicator of physical movement. When it is high, the participant is very likely to be walking and thus not in a vehicle. MEAN_MAG and STD_MEAN_MAG_5WIN are good indicators for riding metros. When they are high, the participant should be riding metros and thus in a vehicle. NUM_AP is useful here because there are more Wi-Fi hotspots when people are sitting or walking compared with riding vehicles.

After classification, there could be a lot of noises existing in the results. For example, a few samples could be classified as vehicle while all the other samples adjacent to them are classified as non-vehicle, which is obviously false because vehicle segment must last for a certain duration. To deal with these noises, a structure prediction algorithm such as Hidden Markov Model (HMM) can be adopted here to perform a smoothing process. Due to easiness however, here a heuristic smoother is used to simply remove unrealistically short non-vehicle segments among vehicle segments and short vehicle segments among non-vehicle segments.

C. Walking-or-Not Classification

After vehicle segment detection, all the non-vehicle data samples are then further classified into either walking or stationary modes by a sample-wise walking-or-not classifier.

Same as the vehicle-or-not classifier, walking-or-not classifier is an AdaBoost classifier. It contains 10 decision trees as its weak learners and uses only three features as shown in Table II. According to the importance ratios, STD_ACC and MOV_AVE VELOCITY are the dominant features because physical motion and geographical movement are very good indicators to differentiate walking and stationary. TIME DELTA is an indicator of whether the device just awakes from sleeping mode. When no motion or movement is detected, the device’s idle waiting time is increased and it falls asleep. Once any motion is detected, the device wakes up and TIME DELTA sampled at that moment is much larger than normal values. Including TIME DELTA in the features of walking-or-not classifier makes sure that the sleeping period is classified as stationary.

D. Vehicle Type Classification

All the detected vehicle segments are classified into different vehicle types by using a Random Forest classifier. The classifier has 30 decision trees as the weak learners and uses totally 17 segment-wise features as shown in Table II. These features can be summarized into three types:

- Features which describe how close to public transportation lines the recorded travel track is, including features from BUS_DIST_BEGIN to METRO_DIST_STD in the table. They are calculated from the recorded location values and known locations of bus stops and metro stations.
- Features related to how long and how fast the vehicle segment is, including features from START_VELO to
| Classifier   | Features       | Descriptions                                                                 | Importance | ratios |
|--------------|----------------|------------------------------------------------------------------------------|------------|--------|
| Sample-wise  | STD_ACC        | Standard deviation of 100 samples of 3-axis accelerometer magnitude          | 0.1333     |        |
| Vehicle-or-not| MEAN_MAG       | Mean of 100 samples of 3-axis magnetometer magnitude                        | 0.1333     |        |
| Classifier   | STD_MEAN_MAG_5WIN | Standard deviation of 5 recent samples of MEAN_MAG          | 0.1        |        |
|              | MOV_AVE VELOCITY | Average of 5 recent samples of geographic velocity calculated from LAT and LON values | 0.5333     |        |
|              | NUM_AP         | Number of access points in the surrounding                              | 0.1        |        |
| Sample-wise  | STD_ACC        | Standard deviation of 100 samples of 3-axis accelerometer magnitude          | 0.4        |        |
| Walking-or-not| MEAN_MAG       | Mean of 100 samples of 3-axis magnetometer magnitude                        | 0.5        |        |
| Classifier   | MOV_AVE VELOCITY | Average of 5 recent samples of geographic velocity calculated from LAT and LON values | 0.1        |        |
|              | TIME_DELTA     | Time span between current data sample with the previous one               | 0.1        |        |
| Segment-wise | BUS_DIST_BEGIN | Distance between segment’s start location to the nearest bus stop            | 0.019      |        |
| Vehicle Type | METRO_DIST_BEGIN | Distance between segment’s start location to nearest metro station       | 0.008      |        |
| Classifier   | BUS_DIST_END   | Distance between segment’s end location to nearest bus stop                 | 0.028      |        |
|              | METRO_DIST_END | Distance between segment’s end location to nearest metro station           | 0.028      |        |
|              | BUS_DIST_AVE   | Average distance between each point of the segment to the nearest bus stop  | 0.059      |        |
|              | BUS_DIST_STD   | Standard deviation of distance between each point to the nearest bus stop   | 0.051      |        |
|              | METRO_DIST_AVE | Average distance between each point to the nearest metro station           | 0.088      |        |
|              | METRO_DIST_STD | Standard deviation of distance between each point to the nearest metro station | 0.035     |        |
|              | START_VELO     | The beginning velocity of the segment                                       | 0.025      |        |
|              | AVE_VELOCITY   | Average velocity in the entire segment                                      | 0.171      |        |
| TEPPER       | 85_PERC_VELO   | 85th percentile of all velocity values in the entire segment               | 0.094      |        |
|              | 95_PERC_ACC    | 95th percentile of all acceleration values in the entire segment            | 0.082      |        |
|              | TOT_DIST       | Total travel distance of the segment                                        | 0.049      |        |
|              | TOT_DURA       | Total duration of the segment                                               | 0.063      |        |
|              | WAITING_TIME   | Time spent by the participant in the same location before the vehicle segment | 0.125     |        |
|              | STD_MEAN_MAG   | Standard deviation of MEAN_MAG in the entire segment                        | 0.039      |        |
|              | PERC_FAIL_LOC  | Percentage of points with failed localization among all points of the segment | 0.036     |        |

TOT_DURA in the table. They are simply calculated from the recorded values of TS, LAT and LON.

- Features which help to distinguish one specific mode, including WAITING_TIME, STD_MEAN_MAG and PERC_FAIL_LOC. If WAITING_TIME is large, it means that the participant waits for a long time before starting the vehicle segment and thus he/she is very likely to be riding a public transportation. If the participant is riding a metro, the value of STD_MEAN_MAG is much higher than usual due to the magnetic effect from big electric motor of the metro train. Because we use Wi-Fi based localization, when the participant is riding a metro or a car, lack of hotspots can cause a failure of localization. As a result, low PERC_FAIL_LOC value tells a high potential of traveling by buses.

Similar as AdaBoost classifiers, Random Forests are also able to provide the importance ratios of used features. From above values, it is clear that the second and third types of segment-wise features are more effective in classifying the segments into different vehicle types.

V. EXPERIMENT DATA PREPARATION

A. Data Collection

There are two different data sets used for training and testing purpose in this work: NSE data set and Lab Experiment data set. Both of them were collected using the NSE sensing devices described in Section III.

The NSE data set used is only one small portion of the entire data set collected during the National Science Experiment (NSE) in 2016. The entire NSE 2016 data set contains data of 4388 volunteered students who wore the NSE sensing devices on their necks with lanyards during daytime for one week. All data are anonymous and securely handled. Total duration of this data set is around 250,000 hours, and total distance of traveling is around 189,000 km.

Since the first data set doesn’t have any ground truth labels originally, it is labeled afterwards by the proposed offline data labeling approach. However, due to its nature, the offline labeling can only provide four types of labels: non-vehicle, riding a metro, riding a bus, and riding a car. Therefore, to provide labels of walking and stationary, and to conduct an evaluation of the offline labeling approach, the Lab Experiment data set was collected in an additional small-scale experiment. In this experiment, we recruited 15 volunteers to collect data using the NSE device. While wearing the devices, they were also required to record their real-time true modes through a specially developed Android APP. The total duration of collected data is around 200 hours, and the total distance is around 3000 km.

The entire NSE data set is used as training data. One portion
of the Lab Experiment data set is used as training data, and the rest is used for testing.

B. Data Labeling

In order to train and test the mode identification algorithm described in Section IV, it is necessary to obtain ground truth labels (true transportation modes) of the two data sets. An offline labeling approach is proposed to label the NSE data set, while traditional label recording approach is used to obtain labels of Lab Experiment data set.

1) Offline Labeling Approach: Because the participants in NSE are mostly primary and secondary school students, it is not possible to train so many students to record or report their ground truth labels for us. Therefore, the data collected in NSE have no ground truth labels from the collectors. On the other hand, to make use of the data, we need the labels. An offline data labeling approach is then implemented to label the NSE data set after it is collected and stored in the cloud. The main idea is that traveling between same end points using different transportation normally results in different geographical tracks/trajectories (see Fig. 3). By comparing the recorded travel track with all possible routes of different transportation tools, one can tell the true transportation modes which the commuter experiences.

Based on this idea, a web portal is firstly implemented for researchers to manually label the true modes of each data sample along the trip. The web portal, as shown in Figure 4, displays all the data samples with valid locations on the map for each trip. Through the User Interface (UI), researchers can choose to display routes of different transportation tools returned by Google Directions API on the map. Supplementary information such as total travel distance and duration is also provided. For each trip, the user compare the geographical shape, duration and distance between the real track and different Google routes. If there is one Google route which is almost the same to the real track in terms of all the attributes, the user can then label the data samples along the real track accordingly. All the points without valid positions but within a commuting segment labeled will also be assigned the same type of travel mode as its surrounding points. If no best-fit Google route is found, this single trip is skipped, left as unable to label offline.

One trip, mentioned above, is defined as a sequence of data samples between two significant stays, such as students’ home and school. Because the offline labeling approach deals with one trip at each time, the data of each device on each day are firstly processed to catch the trips, following two steps: a) Applying velocity-based filtering and Density-based spatial clustering of applications with noise (DBSCAN) to detect significant stays of the participant on that day; b) The data segment between each two significant stays is then a trip.

To further simplify the process, the above offline labeling approach is also automated. For each trip, we take the first and last points with valid location as the origin and destination of a query sent to Google Directional API. The API will then return one set of all possible routes of different transportation tools between those end points. For each type of transportation, the returned routes are generally more than one. To select the most similar Google route with the real route in terms of both location and time, we design two features to describe their similarity:

- Geographical Similarity, which describes the level of trajectory similarity between candidate route and real route. $S_{route} = 1 - \frac{A_{diff}}{A_{real}}$, where $A_{real}$ is area of the convex hull for all the real routes points, $A_{diff}$ is the total area of polygon(s) encircled by the candidate routes and real route. If there is intersection between the two routes, there will be more than one polygon generated. In most cases, the closer two routes are to each other, the larger $R_{overlap}$ will be as $A_{diff}$ is smaller.
- Duration Similarity, which describes the closeness in duration between candidate route and real route.
\[ S_{\text{duration}} = 1 - \frac{|T_{\text{real}} - T_{\text{test}}|}{T_{\text{real}}}, \]

where \( T_{\text{real}} \) is the time duration the entire real route, \( T_{\text{test}} \) is the time duration the entire candidate route.

We compute a similarity score based on these two features and select the Google route with the highest score which is the most similar to the real route. If the highest score is still below a pre-set minimum score, the automatic labeling process fails to find a qualified ground truth route and thus the current trip is skipped. Otherwise, we treat the selected Google route as the ground truth route, and assign the travel modes suggested by the ground truth route to corresponding data samples along the real route by a sample-wise mapping process.

Due to the nature of this offline data labeling approach, it has three drawbacks:

- It is only able to provide four types of labels, namely non-vehicle (walking or stationary), metro, bus and car.
- It fails occasionally because of failure in finding a qualified reference Google route.
- The labels it provides are not 100% accurate.

Despite of these drawbacks, the proposed offline data labeling approach is still attractive because it avoids the traditional tedious and time-consuming data labeling way in which data collectors need to record their true modes during the collection. With its help, we can now make use of existing unlabeled data sets for mode identification, or conduct large-scale crowd-sensing experiments without participants' involvement in labeling the data.

2) Labeling the Lab Experiment data set: As lab experiment is more controlled, we want to obtain more accurate ground truth to evaluate the offline labeling approach as well as to provide labels of walking and stationary. An Android APP to collect the ground truth label was developed. During the data collection, each volunteer in the lab experiment was asked to bring an Android phone and record their true modes in the APP, by changing the status of travel mode whenever he/she entered a different travel mode. The APP recorded the UNIX timestamp and correct travel mode. We synchronized the recorded labels with sensing data uploaded by the NSE device to assign the ground truth label to each data sample.

3) Evaluation of Offline Labeling Approach: To evaluate the performance of our portal-based manual labeling method, we use one portion of the Lab Experiment data set. The data is firstly labeled manually using the method. Then these labels are compared with the labels obtained from the Android app. First half of Table III shows the confusion matrix and it reports an overall accuracy of 98%. Also, 82.1% of the data samples are successfully labeled by the manual labeling method. These results indicate that our portal-based manual labeling method is applicable to provide ground truth labels for the data.

To evaluate the performance of automatic labeling system, we compare the labels it provides with those provided by the manual labeling portal. Second half of Table III shows the comparison confusion matrix. An overall accuracy of 86% is reported and the labeling success rate is 9.4%. As expected, the automated process performs worse than the manual labeling portal with a poorer accuracy and a much lower success rate. However, because NSE data set is too large to be labeled manually thoroughly, automated labeling process is still adopted to label all of NSE data set. Results and discussions presented in Section VI show that it is proper to do so.

VI. RESULTS AND DISCUSSION

A. Performance of the Vehicle-or-Not and Walking-or-Not Classifiers

The sample-wise vehicle-or-not classifier is trained and tested using one portion of the NSE data set which is successfully labeled by the portal-based manual labeling method. The used data are collected by 76 students and contain 321 trips with a total duration of 175 hours and a total distance of 2215 km. 2/3 of these data are used for training and the rest are used for testing.

First two parts of Table IV show the testing results of the Vehicle-or-Not classification before and after smoothing respectively. An overall accuracy of 94% is reported. By comparing the two sets of results, one can see that the smoothing helps to improve the accuracy of sample-wise vehicle-or-not classification.

Because the offline data labeling approach doesn’t provide walking and stationary labels, 1/3 of the Lab Experiment data set is used to train the sample-wise walking-or-not classifier. Its evaluation is conducted by a 10-fold cross-validation whose results are shown in the last part of Table IV. The reported overall accuracy is 79%, which is a middle-class performance compared with others in the literature. The relatively low accuracy is mainly caused by three things:

- The NSE device is hung on one’s neck. This makes it harder for the device to sense people’s motion compared with smartwatch worn on the wrist and smartphone put in the pocket.
- The Wi-Fi based localization cannot catch people’s walking paths. When the sensor is carried by a walking person, its location values just jump between different buildings in the surroundings.

| TABLE III | CONFUSION MATRIX OF PORTAL-BASED MANUAL LABELING METHOD |
|-----------|----------------------------------------------------------|
|           | Precision | Recall | F1-score | Counts   |
| Manual Labeling |           |         |          |          |
| Walking/Stationary | 0.94 | 0.90 | 0.92 | 785  |
| Metro | 0.99 | 1.00 | 0.99 | 4999 |
| Bus | 0.93 | 0.99 | 0.96 | 1265 |
| Car | 1.00 | 0.97 | 0.99 | 3671 |
| Sum | Accuracy: 0.98 | 10720  |
| Automatic labeling |           |         |          |          |
| Walking/Stationary | 0.87 | 0.76 | 0.81 | 1596 |
| Metro | 0.91 | 0.89 | 0.90 | 2565 |
| Bus | 0.82 | 0.82 | 0.82 | 2343 |
| Car | 0.84 | 0.95 | 0.89 | 1751 |
| Sum | Accuracy: 0.86 | 8255  |
• Data collectors make mistakes when recording their true modes. Because walking and stationary change to each other rapidly and frequently, it is hard to record them correctly.

| TABLE IV | ACCURACY OF EACH SAMPLE-WISE CLASSIFIER |
|----------|---------------------------------------|
|          | Precision | Recall | F1-score | Counts  |
| Vehicle-or-not Classifier |          |        |          |         |
| Vehicle   | 0.86      | 0.89   | 0.87     | 2682    |
| Non-vehicle | 0.95     | 0.94   | 0.94     | 6143    |
| Sum       |           |        |          | 8825    |
| Vehicle-or-not Classifier (smoothed) |          |        |          |         |
| Vehicle   | 0.89      | 0.91   | 0.90     | 2682    |
| Non-vehicle | 0.96     | 0.95   | 0.96     | 6143    |
| Sum       |           |        |          | 8825    |
| Walking-or-not Classifier |          |        |          |         |
| Walking   | 0.75      | 0.83   | 0.78     | 1476    |
| Non-walking | 0.84     | 0.77   | 0.80     | 1772    |
| Sum       |           |        |          | 3248    |

B. Performance of Vehicle Type Classification

Unlike the previous two binary classifiers, sample-wise vehicle type classifier identifies three vehicle classes: metro, bus and car. In real world applications, the conditions of these three vehicle classes vary a lot, which requires the classifier to handle large-variety data sets. Therefore, over-fitting problem must be avoided when training the classifier, which requires a large-size and diverse training data set. As a result, we perform the automatic labeling process on all NSE data set to label it. Due to the low success rate of automatic labeling as shown in Section V-B3, only around 10% of the NSE data set is successfully labeled. But the labeled data are still very large in size and diverse: they are collected by 2200 different students and contains 4504 different trips with a total duration around 1600 hours and a total distance around 24700 km.

A 10-fold cross-validation is conducted to evaluate the performance of the vehicle type classifier. Table V shows the resulted confusion matrix and an overall accuracy of 87% is reported. As can be seen from the table, the counts of three classes are not even, so the true accuracy could be less. Moreover, because the automatic labeling itself has an accuracy around 86%, the accuracy values shown in Table V therefore cannot be used as an absolute criteria to the classifier’s performance. But it does show that the sample-wise vehicle type classifier works. More convincing evaluation is conducted in the following section where Lap Experiment data with Android APP collected labels are used to test the performance of the entire algorithm.

C. Performance of Entire Algorithm

Because the above three classifiers are trained and tested separately using different data sets, a performance test of the entire algorithm is conducted using the other 2/3 of the Lap Experiment data set. The testing data therefore are labeled by data collectors using the Android APP. Because the labels collected by the Android APP are more accurate and cover all five types of transportation modes, the results of this test are convincing.

Table VI shows the confusion matrix of the test. It shows that our algorithm achieves a total accuracy of 81% for all the travel modes. Stationary mode has the lowest precision and many walking samples are classified as "stationary". Similar situation occurs for Walking mode. Reasons for poor performance of the sample-wise walking-or-not classifier have been discussed in Section VI-A. Errors of the vehicle-or-not classifier are also obvious in the table. Many non-vehicle samples are classified as vehicle modes, and vise versa. Among the three separate classifiers, segment-wise vehicle type classifier shows the best performance. Metro mode has the highest recall and precision, given the most effective features we identified.

Generally, despite that our data quality is poor as a cost of saving power, the developed mode identification algorithm overcomes the issues well and yields a competitive performance compared with literature.

| TABLE VI | CONFUSION MATRIX OF ENTIRE SYSTEM |
|----------|----------------------------------|
|          | Actual                               | Classified as | Stationary | Walking | Metro | Bus | Car | Recall |
|          |                                      |               |            |         |       |     |     |        |
| Metro    | 907                                  | 218            | 34         | 21      | 14    | 0.76|
| Walking  | 285                                  | 1003           | 46         | 100     | 72    | 0.67|
| Metro    | 5                                    | 5              | 1734       | 0       | 0     | 0.99|
| Bus      | 176                                  | 83             | 17         | 1110    | 191   | 0.70|
| Car      | 20                                   | 46             | 0          | 110     | 1468  | 0.89|
| Precision| 0.65                                 | 0.74           | 0.95       | 0.83    | 0.84  |      |

D. Comparison with other work

Table VII compares our work with other 9 related ones in terms of six features. We adopt the lowest sampling frequency among all. The data size used in our work is among the highest while the number of data collectors is much larger than others, which benefits from the efficient offline data labeling approach. Among those which identify same types of modes, our work has one of the best accuracy. Moreover, we use Wi-Fi based localization instead of GPS. Together with the lowest sampling frequency, this helps to save much more power consumption.

VII. Conclusion

In this paper, we presented a comprehensive transportation mode identification solution to large-scale applications, includ-
ing a power-saving data sampling manner, a corresponding mode identification algorithm, and an offline data labeling approach.

The sampling manner replaces commonly used GPS with Wi-Fi based localization and adopts a very low sampling frequency to minimize the power consumption. The mode identification algorithm is designed based on a hierarchical structure and successfully overcomes the restricted data quality with an overall accuracy of 81%. The novel offline data labeling approach avoids typical tedious, time-consuming data labeling methods and enables the usage of unlabeled data.

Our solution can be extended in different platforms and in different parts of the world. It deals with the issues of current transportation mode identification, namely high power consumption and difficulties in obtaining large-size and diverse labeled data. With its help, large-scale applications of mode identification become practical.

REFERENCES

[1] S.-W. Lee and K. Mase, “Activity and location recognition using wearable sensors,” IEEE pervasive computing, vol. 1, no. 3, pp. 24–32, 2002.
[2] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in International Conference on Pervasive Computing, Springer, 2004, pp. 1–17.
[3] V. Manzoni, D. Maniolfi, K. Kloeckl, and C. Ratti, “Transportation mode identification and real-time co2 emission estimation using smartphones,” SENSEable City Lab, Massachusetts Institute of Technology, nd, 2010.
[4] W. Bohte and K. Maat, “Deriving and validating trip purposes and travel modes for multi-day gps-based travel surveys: A large-scale application in the netherlands,” Transportation Research Part C: Emerging Technologies, vol. 17, no. 3, pp. 285–297, 2009.
[5] A. Ghorpade, F. C. Pereira, F. Zhao, C. Zegras, and M. Ben-Akiva, “An integrated stop-mode detection algorithm for real world smartphone-based travel survey,” in Transportation Research Board 94th Annual Meeting, no. 15-6021, 2015.
[6] M. A. Shafique and E. Hato, “Formation of training and testing datasets, for transportation mode identification,” Journal of Traffic and Logistics Engineering Vol, vol. 3, no. 1, 2015.
[7] S. Wang, C. Chen, and J. Ma, “Accelerometer based transportation mode recognition on mobile phones,” APWCNS, vol. 2010, pp. 44–46, 2010.
[8] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems. ACM, 2013, p. 13.
[9] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, “Understanding transportation modes based on gss data for web applications,” ACM Transactions on the Web (TWEB), vol. 4, no. 1, p. 1, 2010.
[10] Q. Zhu, M. Zhu, M. Li, M. Fu, Z. Huang, Q. Gan, and Z. Zhou, “Identifying transportation modes from raw gps data,” in International Conference of Young Computer Scientists, Engineers and Educators. Springer, 2016, pp. 395–409.
[11] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, “Using mobile phone barometer for low-power transportation context detection,” in Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems. ACM, 2014, pp. 191–205.
[12] A. M. AbdelAziz and M. Youssef, “The diversity and scale matter: Ubiquitous transportation mode detection using single cell tower information,” in 2015 IEEE 81st Vehicular Technology Conference (VTC Spring). IEEE, 2015, pp. 1–5.
[13] D. Shin, D. Aliaga, B. Tunçer, S. M. Arisona, S. Kim, D. Zünd, and G. Schmitt, “Urban sensing: Using smartphones for transportation mode classification,” Computers, Environment and Urban Systems, vol. 53, pp. 76–86, 2015.
[14] G. Bieber, T. Kirste, and M. Gaede, “Low sampling rate for physical activity recognition,” in Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments. ACM, 2014, p. 15.
[15] A. Subramanya, A. Raj, J. A. Bilmes, and D. Fox, “Recognizing activities and spatial context using wearable sensors,” arXiv preprint arXiv:1206.6869, 2012.
[16] M. Lin, W.-J. Hsu, and Z. Q. Lee, “Detecting modes of transport from unlabelled positioning sensor data,” Journal of Location Based Services, vol. 7, no. 4, pp. 272–290, 2013.
[17] E. Wilhelm, S. Siby, Y. Zhou, X. J. S. Ashok, M. Jayasuriya, S. Foong, J. Kee, K. Wood, and N. O. Tippenhauer, “Wearable environmental sensors and infrastructure for mobile large-scale urban deployment,” IEEE Sensors, December 2016.

TABLE VII

| Group         | Sampling Freq. (Hz) | Sensor types          | Data size  | Number of data collectors | Accuracy (%) | Mode types                          |
|---------------|---------------------|-----------------------|------------|---------------------------|--------------|------------------------------------|
| Abdelaziz et al[12] | N/A                  | phone cell info.      | 135 hours  | 4                         | 89           | stationary, walking, driving       |
| Bohte et al[14]          | 0.0167               | accelerometer (ACC)   | 1500 hours | 13                        | 82.5         | physical and transportation        |
| Ghorpade et al[5]        | 1                   | ACC+GPS+GSM+Wi-Fi     | 500 hours  | 10                        | 90           | physical activities only           |
| Hemminki et al[8]        | 60-100               | ACC                    | 150 hours  | 16                        | 80.8         | physical and transportation        |
| Sankaran et al[11]       | 1                   | barometer              | 178 hours  | 13                        | 69           | idle, walking, vehicle             |
| Shin et al[13]           | 1                   | ACC+GSM+Wi-Fi         | N/A        | 30                        | 82.05        | physical and transportation        |
| Wang et al[7]            | 35                  | ACC                    | 12 hours   | 7                         | 70           | physical and transportation        |
| Zheng et al[9]           | 0.5                 | GPS                    | 2000 hours | 65                        | 71           | physical and transportation        |
| Zhu et al[10]            | 0.2-0.5             | GPS                    | N/A        | 73                        | 82.85        | physical and transportation        |
| Our Work                | 0.08                | ACC+megnetometer+Wi-Fi | 1600 hours | 2200                      | 81           | physical and transportation        |