Transportation Type Identification by using Machine Learning Algorithms with Cellular Information

Yi-Hao Lin, Jyh-Cheng Chen, Chih-Yu Lin, Bo-Yue Su, Pei-Yu Lee
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

Email: {yhlin1377, jcc, cylin, bysu8902, pylee}@cs.nctu.edu.tw

***Abstract*—It is crucial for future 5G networks to intelligently understand how users move so that the networks can allocate different resources efficiently. In this paper, we try to find practical features to identify four common types of motorized transportations, including High-Speed Rail (HSR), subway, railway, and highway. We propose a system architecture that can provide accurate, real-time, and adaptive solution by using cellular information only. Because we do not use GPS as that in most of the prior studies, we can reduce energy consumption, size of log data, and computational time. Around 500-hour data are collected for performance evaluation. Experimental results confirm the effectiveness of the proposed algorithm, which can improve well-known machine learning algorithms to approximately 98% classification accuracy. The results also show that battery consumption can be reduced about 37%.**

***Index Terms*—Transportation Type Identification, Machine Learning, Cellular Information, Classification, 5G**

I. Introduction

One of the important features in future 5G networks is *network slicing*. It allows a mobile operator to provide dedicated virtual networks with function-specific resources to different types of users over a common network infrastructure. Thus, the networks are able to support numerous and various services envisaged in 5G. As shown in Fig. 1, a single physical network denoted as next generation network will be sliced into multiple virtual networks that can support different mobility types of users. Different network resources and specific mobility policies will also be allocated to different types of users across a common core network. It is critical to enhance intelligence in the 5G era to automatically recognize a service type, infer the appropriate provisioning mechanisms, and establish the required network slices.

In order to incorporate intelligence into next-generation networks, the service requirements of users need to be investigated first. In this paper, we focus on analyzing user mobility as shown in the red box in Fig. 1 to identify user's transportation type. We apply machine learning algorithms into Transportation Type Identification (TTI). We focus on identifying motorized transportation types because the moving speed of non-motorized types such as walking, jogging, and biking are not fast enough for mobile operators to allocate specific resources. Moreover, the non-motorized types could be detected by using Inertial Measurement Unit (IMU) data such as accelerometer, gyroscope, and rotation vector in the smartphone [1], [2].

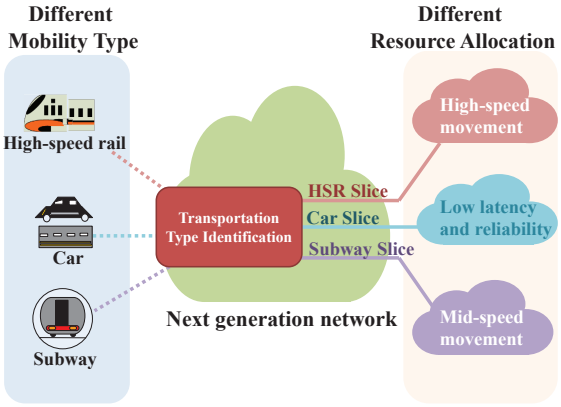

Fig. 1. Overview of next generation networks.

In this paper, we try to find practical features to identify four common types of motorized transportations, including *High-Speed Rail (HSR), subway, railway, and highway*. We use various machine learning algorithms to train the classification models. Many well-known learning algorithms are evaluated and the results are presented in this paper. The main contributions of this paper include:

1) We propose a system architecture that can provide accurate, real-time, and adaptive solution for TTI by using *cellular information only*. The architecture could be further applied to different applications such as carbon footprint, intelligent navigation, elderly tracking, smart city, and transportation traffic analysis.
2) We propose to extract Base Station (BS) identification as the location information for TTI. We are the first to integrate cellular data and location information and propose Cellular Information with Sliding Window (CISW) algorithm for machine learning algorithms. The proposed CISW is not limited to smartphone but also can be used for Internet of Things (IoT) devices.
3) Around 500-hour labeled data for offline training are collected by our prior work [3], [4]. We release the dataset, source code, and pre-trained models to anyone who is interested in exploring TTI problem more [5].

II. Challenges

A smartphone can provide many information, ranging from GPS, accelerometer, magnetometer, barometer, gravity sensor,

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

light sensor, to cellular/wireless radios. Although these information could help TTI, there are still some challenges for using these data.

- **Environmental limitations:** GPS is useful for TTI because it provides the physical position of the user. However, GPS requires line-of-sight between devices and satellites. Thus, it is not suitable for tunnels or the urban environment due to obstacles. Especially, GPS is not available in the subway which is underground. For magnetometer, barometer or light sensor, they would be affected easily by electronic devices, humidity/temperature, and sunlight.
  Our solution: Nowadays, base stations are widely deployed along public transportation no matter on the ground or under the ground. As described in Sec. V-B, we use cellular data instead of GPS signals for positioning.
- **Energy consumption:** Computing resources are the main limitations for long-term sensing when using smartphones. High sampling rate would drain battery fast. GPS also consumes considerable energy [6].
  Our solution: We don't use GPS because we use cellular information instead. As discussed in Sec. V-C, we propose a new algorithm which only needs BS information and the frequency of handover. Thus, *event-based* sampling can be used. It can reduce energy consumption, size of log data, and computational time.
- **Privacy:** The more sensor readings from a smartphone, the more possibility that we can detect where the user is and what physical activity the user is performing.
  Our solution: Because we only use cellular information, the granularity of location information we collect is coarser than that using GPS. Also, we don't need other sensor readings.

III. Related Work

The TTI problem has been considered as an activity recognition problem. Traditionally, *rule-based* algorithms are proposed for TTI problems. They seek the best thresholds and are designed with if-else patterns. However, the algorithms would be complicated for different transportation types. Therefore, *learning-based* algorithms become a promising paradigm. Recently, many studies apply machine learning techniques into TTI by using the sensed data from smartphones. Nowadays, more and more studies pay attention to *supervised learning* algorithms and aim to improve accuracy by extracting more discriminative features from the data sensed by smartphones. Here, we categorize prior studies into four types of sensed data that are commonly used for TTI problem.

- GPS/GIS information: These algorithms apply *map-matching technique* to match the GPS/Global Information System (GIS) trajectories of the user with coordinates of transportation in the referenced database. GIS data can be used to create more distinguished features as detailed in [7], [8]. In addition to GPS/GIS information, the authors of [9], [10], [11], [12] use smartphone sensors to collect accelerometer data of different transportation types, which achieve a better performance than those using GPS data only. Zheng et al. [13] use common statistical features such as mean velocity, expectation of velocity, top three velocities, and top three accelerations to identify four different transportation types (bike, bus, car, walking). In addition, Zheng et al. [14] further introduce more advanced features including the Heading Change Rate (HCR), the Stop Rate (SR), and the Velocity Change Rate (VCR), which achieve a more accurate result.
- Inertial Measurement Unit (IMU) data: These algorithms use accelerometers, gyroscopes, rotation vectors, and magnetometer to collect more discriminative information for the behaviors of smartphone users [1], [2].
- Other sensors: The authors of [15], [16] use body temperature, heart rate, light intensity obtained from other sensors as features, and are able to build a model to predict user activities, including walking, running, rowing and cycling.
- Wireless signals: Movement can also be detected by using the change in cellular, Wi-Fi, or bluetooth signals. For example, the authors of [17], [16], [18] use fluctuation in cellular signal strength to detect whether the user is stationary, walking, or driving.

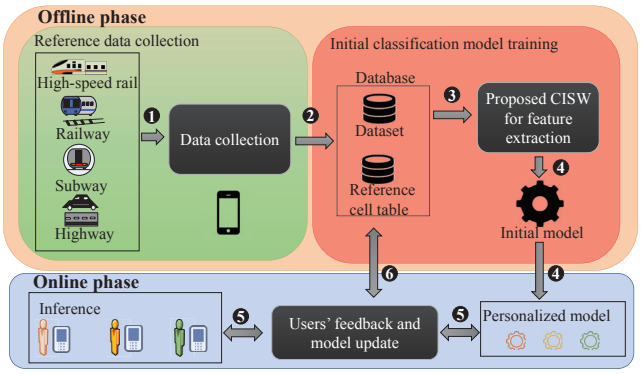

Fig. 2. Overview of the proposed system.

In this paper, we focus on cellular information (i.e., the category of wireless signals). Instead of taking the signal fluctuations as features in the prior studies, we propose a new algorithm to integrate location information and cellular data that have not been used in prior works. Due to the wide deployment of cellular towers (aka BS) along public transportation, cellular information is more reliable than GPS/GIS data, especially for subways. Our algorithm only focuses on which cellular tower the smartphone connects to and the frequency of handover. Thus, it can reduce energy consumption, data size, and CPU power.

IV. System Overview

A. Proposed Architecture

Fig. 2 depicts the overview of the proposed system architecture. There are two phases. The *offline phase* generates a classification model by the baseline dataset. The *online phase*

updates (fine-tunes) the model with user's feedback to improve the results adaptively. In the end, different personal model is generated for each user and the new dataset can be stored in the database for further analysis. These new datasets are helpful for a new version of the initial model training. Next, we present the detail of each procedure.

1) Initial data collection: The *baseline* data such as GPS and cellular information are collected by ourselves. Noted that GPS information is not needed for our algorithm. We collect it because we may use it to verify the correctness of our labeling in some cases. With the APP we developed [3], [5], we have collected data from each type of transportation with time stamps, which are used for marking correct label for each data sample. So far, we have collected over 500-hours data as shown in Table I.
2) Data upload: The collected data in Step 1 are uploaded to our server through the Internet and saved in the SQL-based database.
3) Model training: Through cleaning and normalizing, multiple features are extracted (original data) by CISW. We can select the best performance model with various validations and tune the best hyper-parameters as shown in Fig. 3.
4) Model deployment: The pre-trained model is generated in Step 3. The model can be deployed in the APP or in the type of application interface such as RESTful APIs for users to query.
5) Online training: The feedbacks of the users are inputed to the initial model that can be updated adaptively for each user as shown in 3. During *online phase*, the sensed data from mobile users are transformed to feature vectors and fed into the pre-trained model developed on the *offline phase* to generate the personalized transportation mode classifier.
6) Feedback collection: We adopt the online learning algorithm as described in Sec. VI-B such that the pre-trained model can be fine-tuned by user's feedbacks. The feedback can be represented as a period of time with a label the user just marked. We only evaluate the feasibility of the proposed system. Due to space limit, how to verify the correctness of user's feedbacks is out of the scope of this paper.

B. Typical Workflow

The workflows of training and inference are shown in Fig. 3, which includes offline phase and online phase. In the offline phase, we first divide the original collected data as shown in Table I into test and training sets with proportion of 3:7. The test dataset is used for final accuracy evaluation. The training dataset can be further separated into *validation data* and *training data* with proportion of 1:4 for 5-fold cross-validation. In other words, we split training dataset into 5-fold datasets evenly and randomly. During training, we use the 4-fold datasets and we use the dataset in the remaining fold as the new data we have never used before. In data pre-processing,

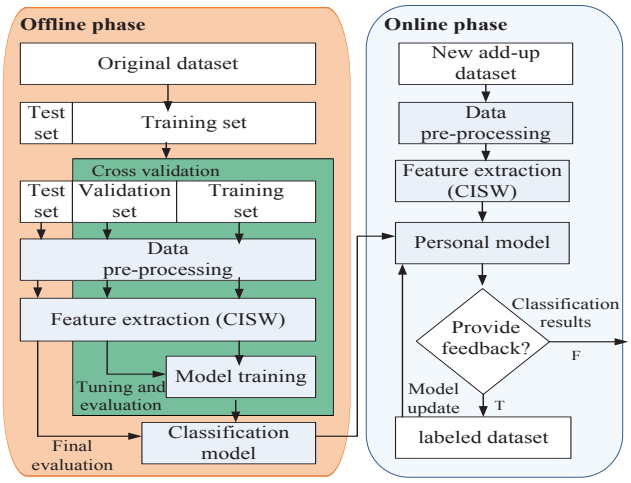

Fig. 3. Typical workflow.

TABLE I
BASELINE DATA DESCRIPTION

Type	Duration	Sample number	GPS availability (%)
HSR	142 hours	12012	69%
Subway	120 hours	8426	0%
Railway	206 hours	42681	68%
Highway	116 hours	38160	86%
Others	56 hours	20804	61%

we remove some outliers, which are deemed abnormal. Next, the proposed CISW is applied for feature extraction from the dataset. With the grid search technique, we can tune the different hyper-parameters among various learning algorithms. Finally, the best model with highest accuracy will be selected and evaluated with the test dataset. In the online phase, the pre-trained model which is generated in the offline phase will be deployed for general users. The add-up dataset are collected from the user and fed into pre-trained model after data pre-processing and CISW. While the user provides the feedbacks in the form of a series of labels, the pre-trained model will get a chance to update (fine-tune) the model. It can improve the accuracy adaptively for personal model.

V. Proposed Cellular Information with Sliding Window (CISW)

In this section, we first introduce the terminologies. Next, the reference cell table establishment and the proposed CISW are presented.

A. Terminologies in cellular information

Fig. 4 shows the cellular information retrieved from a smartphone when the user keeps moving. The smartphone connects to the BS with Cell-ID_i (in $Cell_i$) in the beginning. We can get the BS information through an Android Application Interface (API) called *onSignalStrengthsChanged()*. The BS information contains: (1) mobile country code (MCC) and mobile network code (MNC) to identify a mobile network operator (carrier) globally, (2) location area codes (LAC) or tracking area codes (TAC) used in 3G and 4G to identify the

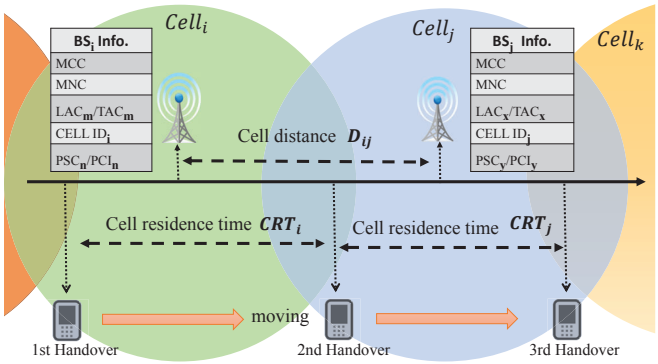

Fig. 4. Cellular information retrieved from smartphone.

location area (the coverage of a group of BSs), (3) Cell-ID, a generally unique number used to identify each BS, and (4) primary scrambling codes (PSC) and physical cell ID (PCI) used in 3G and 4G to identify a cell at physical layer. After handover, the serving cell of the smartphone changes from Cell-ID_i to Cell-ID_j when the user moves to the boundary of $Cell_i$ and $Cell_j$. Cell distance D_{ij} is the distance between BS_i and BS_j. Cell residence time CRT_i is the time from Cell-ID_i to Cell-ID_j.

B. Reference cell table

Most of the previous proposed cellular-based algorithms leverage only the number of handovers and received signal strength from serving and neighboring BSs for TTI [18], [19]. However, the received signal strength fluctuation is easily affected by the environment. We further use the location information of BSs because we can estimate where the user roughly is by tracking the serving cell of the smartphone. In other words, we can identify transportation type by tracking the sequence of BSs which are near the transportation trajectories. However, two challenges need to be addressed: (1) How to get the location information of BSs, and (2) how to know a cell is located near the transportation trajectory? We propose to use *reference cell table*, which is a database maintaining the cell location information and different types of cell-IDs near the four transportation types (HSR, subway, railway, and highway). Next, we elaborate how to use reference cell table to address the two challenges.

- *Location information of BSs:* Ideally, we can get the location information of BSs from mobile operators. However, many operators won't release such information. Fortunately, location information of BSs can be collected through crowd-sourcing. For example, Google [20] and OpenCellId [21] collect such information and release them to public. We also recruited volunteers to collect the location information of BSs [3] for the top-five operators in Taiwan. The cell location can then be calculated by weighted centroid-based approach [22]. The idea is that a good signal strength corresponds to a close proximity of the BS. The estimated location of the BS can be calculated using the following formula with w_K being the weight of the kth measurement:

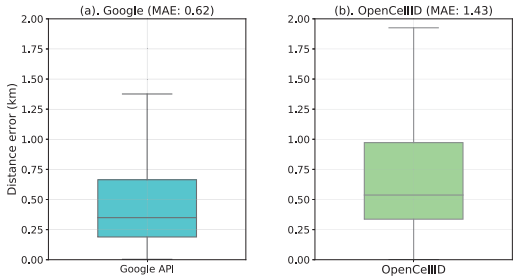

Fig. 5. The position errors of cell location with Google and OpenCellId.

$$P = \frac{1}{\sum_{K=1}^{n} w_K} * \sum_{K=1}^{n} * w_K * P_K \qquad (1)$$

In Fig. 5, we compare the Mean Absolute Error (MAE) of the location of each BS we collected between Google and OpenCellId. Figs. 5(a) and (b) show the distance errors between Google and OpenCellId with the database we created. The distance errors of Google are smaller than OpenCellId with 300 m in average. This means that Google database is more accurate in Taiwan. Thus, we adopt Google database if a BS is not found in our database.

- *Cell-IDs:* With the software we developed, we can collect the Cell-IDs through the whole journeys of different transportations [3], [5].

In short, we create a reference cell table that contains the location information and Cell-IDs that can be used for TTI.

C. Cellular Information with Sliding Window (CISW)

The cellular features are extracted by a group of samples, called segment, in the log files as shown in Fig. 6. Noted that we integrate samples into one segment with length t seconds. In addition to the number of handovers, we present two cellular features which have not been used before. (1) The cell distance during one segment, and (2) the matched cell number for each transportation type. More details about the proposed cellular features are described in the following.

- The handover rate during one segment: Ideally, the faster the user moves, the more handovers are incurred. In real world, due to the noisy nature of the wireless signal propagation and the load of the BS, the serving BS would switch back and forth between different nearby serving cells. This phenomenon is called ping-pong effect [23]. It means that the number of handovers with ping-pong effect cannot represent the speed of the user. In order to address this phenomenon, we only count the dominate (unique) serving cell-IDs in one segment.
- The cell distance during one segment: In order to estimate the moving speed of a user, we can calculate the cell distance D between first and last serving BS in one segment (it means the displacement). The BS location can be retrieved from the *reference cell table* discussed earlier.
- The numbers of matched cells for each transportation type: This feature contains five numbers, each one for

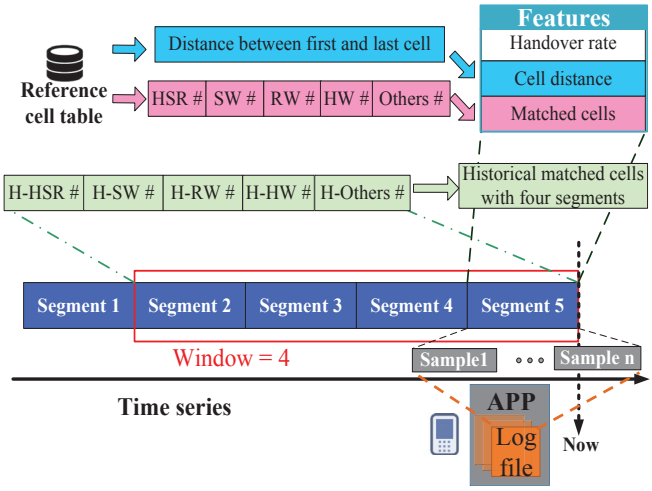

Fig. 6. Proposed CISW.

one transportation type, including HSR, subway, railway, highway, and others. We count the numbers of matched cell-IDs for each transportation type in one segment. The more matched numbers of a certain type, the more possible that the user is on the trajectory of that transportation type.

However, we still cannot make sure that our reference cell table contains all cell-IDs. It is still possible to encounter unknown cell-IDs which are not maintained in our reference cell table. To address this problem, we propose three solutions: (1) ignore the unknown cell-ID, (2) count unknown cell-ID as other type (if our reference cell table contains entire cell-IDs, the probability that the unknown cell-ID belongs to other type is high), and (3) estimated by historical information. The third solution can be considered as a new feature presented in the following.

Because the continuity characteristic of taking transportation, it is important to take historical information as a feature for TTI. As depicted in Fig. 6, if we set the window size T as 4, a new feature, which contains the numbers of historical matched cells with four segments, is extracted.

VI. Experimental Results

In this section, we first raise a list of questions that select suitable parameters in CISW. Second, we review the performance metrics to verify CISW accuracy. We then elaborate the performance of CISW by comparing it with some well-known machine learning algorithms. Finally, we discuss the resource consumption of the smartphone.

A list of questions that we aim to find the answers are:

1) In CISW, we propose three solutions to overcome unknown cell-IDs in real-world data. Which one will improve the accuracy more?
2) How to set the segment length S and window size T in CISW?
3) Will the online update from user's feedback achieve better results than that of offline algorithms in terms of accuracy, training time, model size, and prediction time?
4) Will CISW reduce resource consumption of smartphone compared to the baseline algorithm?

Next, we will answer the questions.

A. Evaluation

1) Dataset: We conduct our experiments using the dataset we collected as shown in Table I. The dataset contains over 500-hours for different transportation types and has two categories of training data:

- Baseline dataset: This type of data is used for pre-training model and to verify accuracy in offline phase.
- Add-up dataset: In order to verify the accuracy in online phase, we collect data from the same user with: (1) same smartphone, (2) same network operator, (3) same trajectory of his/her journey. We denote these data as add-up dataset when we consider the training data for each trip of the user in online phase.

2) Performance metrics: We employ two metrics *precision* and *recall* to evaluate the data retrieval performance of CISW. Taking the transportation label as a groundtruth, precision is the fraction of the retrieved data that are correct, while recall is the fraction of the correct data that are successfully retrieved. Both of them are measured by our dataset, and the larger the better.

In order to measure the overall performance of the model, the F1-score, F, is considered, which is a weighted average of the precision and recall [24]. F1-score is defined as following:

$$\frac{2}{F} = \frac{1}{Precision} + \frac{1}{Recall} \tag{2}$$

B. Performance of CISW

1) Classification results in offline phase: The classification results are evaluated by popular algorithms, such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest (RF), XGBoost (XGBT), Multiple Layer Perception (MLP), and Stochastic Gradient Descent (SGD) under SVM. We implement the learning algorithms with python scikit-learn library [25] and tensorfolw [26]. First, the comparisons between the three proposed solutions are presented. Next, the comparisons between different segment lengths S and windows size T are elaborated. Due to page limit, we only show the results of average F1-score for each algorithm.

- Accuracy with three proposed solutions: The results with different segment lengths S are shown in Fig. 7. For solution 1, as shown in Fig. 7-(a), XGBT outperforms others and achieve 97% F1-score when S is set to 240 seconds. Generally, the larger the S, the more information can be considered for the learning algorithms. Therefore, if we ignore unknown cell-IDs, we cannot have sufficient information for classification in shorter segment length. With longer segment length, more information can be considered to overcome unknown cell-IDs. However, it will decrease if S is set too long. This is because two or more transportation modes switching in one segment will cause prediction error easily. For solution 2, as shown

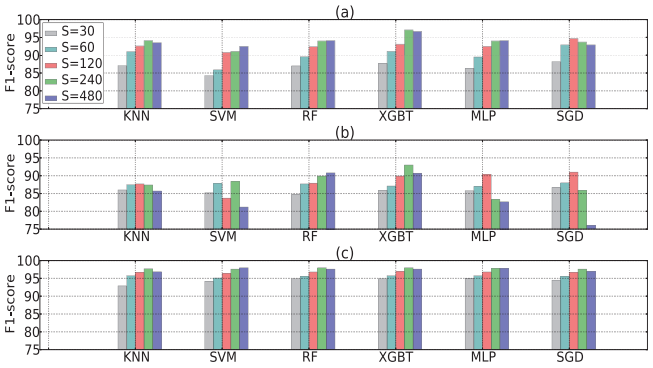

Fig. 7. Comparisons with the three proposed solutions. (a) Solution 1: ignoring the unknown cell-IDs. (b) Solution 2: count an unknown cell-ID as other type. (c) Solution 3: counted by historical information with window size 4.

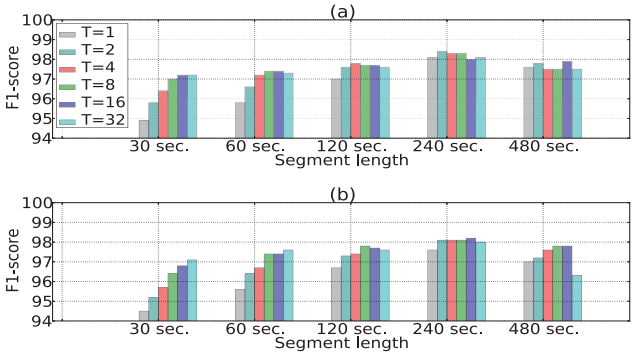

Fig. 8. Comparison of segment S and window size T. (a) offline, XGBT. (b) online, SGD.

in Fig. 7-(b), the accuracy of each learning algorithm is not high enough compared to solution 1. Because we consider the unknown cell-ID as the type of others, it will dramatically interfere the learning algorithms (i.e., decrease recall). For solution 3, as shown in Fig. 7-(c), the accuracies are higher than those of the other two solutions in average. With the help of historical information (sliding window), we can alleviate the impact of unknown cell-IDs.

In summary, solution 3 with sliding window is more suitable for TTI. In the following, we choose XGBT with segment size 120 as the offline learning algorithm and SGD as the online learning algorithm. The reason is that we can reduce one-half of segment time but only degrade 1% accuracy.

- Accuracy with different window sizes: Fig. 8 shows the results between different segment S and window size T in offline (XGBT) and online (SGD) learning algorithms, respectively.

  Generally, the longer the S, the more information feed into the learning algorithms that can achieve higher accuracy as shown in Fig. 8. On the contrary, when T and S are small, the results are less accurate. The is because the amount of information is not sufficient for the learning algorithms. With longer T, however, the algorithms consider more historical information that can improve accuracy even if S is small. With the help of the proposed CISW, SGD still can achieve 94.5% accuracy if T is 1 and S is 30 seconds as shown in Fig. 8-(b).

  The best performance is accomplished when the S is set to 240 seconds. The main reason is that the duration between two stations of the subways in Taiwan is shorter than 4 minutes. Thus, it improves the accuracy of subway, and thus, also increases the overall accuracy. XGBT can achieve 98% accuracy in average as shown in Fig. 8-(a), and 97.8% accuracy for SGD as shown in Fig. 8-(b).

  In summary, when the segment is longer, the system will wait longer to extract cellular information and input the data to the algorithms. Therefore, we tend to choose shorter segment length and window size, and maintain acceptable accuracy at the same time.

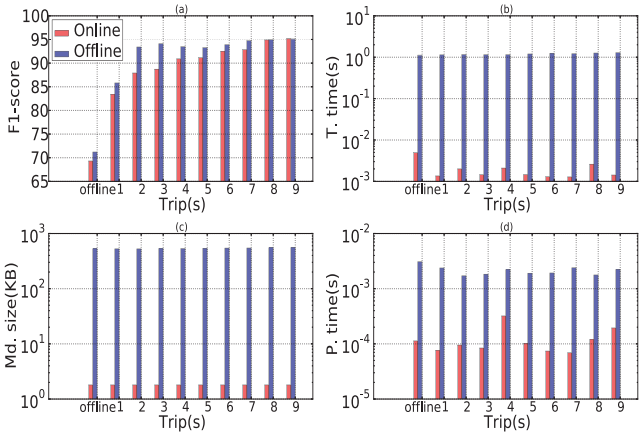

Fig. 9. Performance comparison with add-up dataset. (a) accuracy, (b) training time, T, (c) model size, and (d) prediction time, P.

2) Classification results in online phase: The size of training data is critical for traditional learning algorithms. It implies that the smaller dataset usually results in less accurate models. Although CISW considers some features for classification, we cannot claim that we have collected data for all scenarios. Therefore, we prefer online learning algorithms that can update models with new dataset (i.e., user's feedback). Next, we evaluate CISW in both offline and online learning algorithms with add-up dataset. The accuracy, training time, data size, and prediction time are shown in Fig. 9. We found that with feeding more dataset, we can achieve higher accuracy in both offline and online algorithms as shown in Fig. 9-(a). The accuracy of online algorithm is lower than that of offline algorithm in the beginning, but with more training data, the accuracies of them are close. The model size of online algorithm is only 1 KB compared to 500 KB in offline algorithm which is shown in Fig. 9-(c). As shown in Fig. 9-(b)(d), the trend are similar for training time and prediction time. It only takes 10 ms for online algorithm to train the model and 1 ms for prediction.

C. Resource Consumption

Here we evaluate the battery consumption for CISW and GPS-based learning. Smartphones Xiao-mi note 4 (3080 mAh)

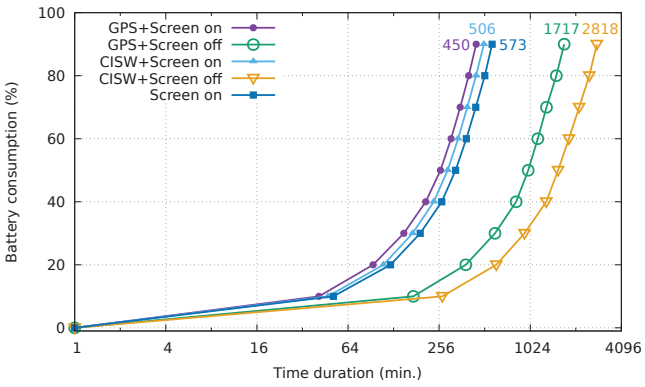

Fig. 10. Comparison of battery consumption.

are used. We compare GPS-based learning (turn on GPS module) and CISW (cellular information) with screen on and off. The results are shown in Fig. 10. If the screen keeps on, CISW can run about one more hour. When the screen is off, CISW can save 18 hours.

VII. Conclusion

It is urgent for 5G networks to identify the service types of users to allocate resources intelligently. Thus, we analyze user mobility pattern in different motorized transportation types, including HSR, subway, railway, and highway. In this paper, we elaborate how to apply machine learning algorithms to realize transportation type identification. We also propose a system architecture that can provide accurate identification for TTI using cellular information only. Around 500-hour dataset are collected for performance evaluation. Experimental results demonstrate the effectiveness of the proposed algorithm, which can improve well-known machine learning algorithms to approximately 98% classification accuracy. The results also show that battery consumption can be reduced about 37%. We will continue to collect more data, and release the data to everyone. For future work, we plan to apply deep learning with CISW to investigate the performance.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology under grant numbers MOST 106-2221-E-009-046-MY3, MOST 106-2221-E-009-047-MY3, and MOST 107-2218-E-009-049.

References

[1] S.-H. Fang, H.-H. Liao, Y.-X. Fei, K.-H. Chen, J.-W. Huang, Y.-D. Lu, and Y. Tsao, "Transportation modes classification using sensors on smartphones," *Sensors*, vol. 16, no. 8, 2016.

[2] X. Su, H. Caceres, H. Tong, and Q. He, "Online travel mode identification using smartphones with battery saving considerations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2921–2934, 2016.

[3] Y.-H. Lin, J.-C. Chen, C.-Y. Lin, B.-Y. Su, and P.-Y. Lee, "Poster: SensingGO - Toward Mobile/Cellular Data Measurement with Social and Rewarding Activities," *in Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom'18)*, pp. 765–767, Oct. 2018.

[4] "SensingGO," http://sensinggo.nctu.me/, 2018.

[5] "Dataset and pre-trained models," http://sensinggo.nctu.me/projects/, 2018.

[6] S. T. Pasha, "Energy efficiency in smartphones: A survey on gps energy conservation," *Imperial Journal of Interdisciplinary Research*, vol. 3, no. 6, 2017.

[7] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gis information," *in Proc. of ACM International Conference on Advances in Geographic Information Systems (GIS'11)*, pp. 54–63, Nov. 2011.

[8] F. Biljecki, H. Ledoux, and P. Van Oosterom, "Transportation mode-based segmentation and classification of movement trajectories," *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 385–407, Feb. 2013.

[9] T. Nick, E. Coersmeier, J. Geldmacher, and J. Goetze, "Classifying means of transportation using mobile sensor data," *in Proc. IEEE International Joint Conference on Neural Networks (IJCNN'10)*, pp. 1–6, Jul. 2010.

[10] A. Jahangiri and H. Rakha, "Developing a support vector machine (svm) classifier for transportation mode identification by using mobile phone sensor data," *in Proc. of 93rd Annual Meeting Transportation Research Board*, no. 14-1442, Jan. 2014.

[11] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2406–2417, 2015.

[12] P. Widhalm, P. Nitsche, and N. Brändie, "Transport mode detection with realistic smartphone sensor data," *in Proc. of IEEE International Conference on Pattern Recognition (ICPR'12)*, pp. 573–576, Feb. 2012.

[13] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw gps data for geographic applications on the web," *In proc. of ACM International Conference on World Wide Web (WWW'08)*, pp. 247–256, Apr. 2008.

[14] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," *In proc, of ACM International Conference on Ubiquitous Computing (UbiComp'08)*, pp. 312–321, Sep. 2008.

[15] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, Jan. 2006.

[16] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. De Lara, "Mobility detection using everyday gsm traces," *in Proc. of ACM International Conference on Ubiquitous Computing (UbiComp'06)*, pp. 212–224, Sep. 2006.

[17] I. Anderson and H. Muller, "Practical activity recognition using gsm data," Technical Report Department of Computer Science, University of Bristol, CSTR 06-016, 2006.

[18] A. M. AbdelAziz and M. Youssef, "The diversity and scale matter: Ubiquitous transportation mode detection using single cell tower information," *in Proc. of IEEE Vehicular Technology Conference (VTC'15)*, pp. 1–5, May 2015.

[19] G. Li, C.-J. Chen, S.-Y. Huang, A.-J. Chou, X. Gou, W.-C. Peng, and C.-W. Yi, "Public transportation mode detection from cellular data," *in Proc. of ACM on Conference on Information and Knowledge Management (CIKM'17)*, pp. 2499–2502, Nov. 2017.

[20] "Google Maps Geolocation API," https://developers.google.com/maps/.

[21] "Unwiredlabs Location API," https://unwiredlabs.com/?ref=ocid.

[22] E. Neidhardt, A. Uzun, U. Bareth, and A. Küpper, "Estimating locations and coverage areas of mobile network cells based on crowdsourced data," *in Proc. of IEEE Wireless and Mobile Networking Conference (WMNC'13)*, pp. 1–8, Jun. 2013.

[23] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," *Computer Networks*, vol. 52, no. 14, pp. 2690–2712, 2008.

[24] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," *in Proc. of European Conference on Information Retrieval*, pp. 345–359, 2005.

[25] "Scikit-learn tool for python," http://scikit-learn.org/stable/.

[26] "TensorFlow framework," https://www.tensorflow.org/.