Detecting Transportation Modes with low-power-consumption sensors using Recurrent Neural Network

Hao Wang, Haiyong Luo, Fang Zhao, Yanjun Qin, Zhongliang Zhao and Yiqu Chen

September 8, 2018

Detecting Transportation Modes with low-power-consumption sensors using Recurrent Neural Network

Hao Wang
School of Software Engineering
Beijing University of Posts and Telecommunications
Beijing, China
wh1995@bupt.edu.cn

Haiyong Luo
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
yhluo@ict.ac.cn

Fang Zhao
School of Software Engineering
Beijing University of Posts and Telecommunications
Beijing, China
zfsse@bupt.edu.cn

Yanjun Qin
School of Software Engineering
Beijing University of Posts and Telecommunications
Beijing, China
qinyanjun@ict.ac.cn

Zhongliang Zhao
Institute of Computer Science
University of Bern
Bern, Switzerland
zhao@inf.unibe.ch

Yiqu Chen
School of Software Engineering
Beijing University of Posts and Telecommunications
Beijing, China
aoschen1996@gmail.com

***Abstract*—With the quick development of mobile Internet and the popularity of smartphones, smartphone-based transportation mode detection has become a hot topic, which is able to provide effective data support for urban planning and traffic management. Though the popular GPS based transportation mode detection method has achieved reasonable accuracy, this method consumes large power, thus limiting it to be used in smartphones. Here, we propose a novel transportation mode detection algorithm using the recurrent neural network. In order to identify transportation modes with low power consumption, this algorithm only uses four low-power-consumption sensors (namely accelerator, gyroscope, magnetometer, and barometer) which are embedded in the commodity smartphones. Furthermore, we exploited the good representative ability of Long Short-Term Memory (LSTM) and applied it to recognizing the transportation modes to achieve higher accuracy. To filter noises, a preprocessing is applied. After calculating features, we adopt the LSTM learning algorithm to train a model of transportation mode recognition and employ this model to predict transportation modes. Extensive experimental results indicate that our proposed approach outperforms the compared state-of-the-art transportation recognition methods with 96.9% accuracy to detect four transportation modes, namely buses, cars, subways, and trains.**

Keywords—deep learning; transportation mode detection; recurrent neural network

I. Introduction

In recent years, various mobile sensing applications based on sensor data has attracted much attention in ubiquitous computing research. As a special mobile sensing application, transportation mode detection based on smartphones can provide mobile users with various value-added services. As an example, using the knowledge of individuals’ mode of transport can improve traveling safety and enable many intelligent applications in the traffic, such as travel guidance [1], low-carbon travel promotion [2], targeted advertising, urban transportation planning, and health monitoring [3].

In order to meet the ever-increasing demand for various applications based on transportation mode detection techniques, a number of research efforts have been undertaken using various sensors. Most of the research on transportation mode methods rely on Global Positioning System (GPS) [4]. Though these methods have achieved reasonable accuracies, they suffer from high power consumption and are unsuitable for the power-limited handheld smartphones to consistently detect transportation modes for a long time. Furthermore, these approaches may not work well when GPS signals are blocked [5], such as in the indoor, underground environments, or in tunnels.

To reduce the power consumption for detecting different types of transportation modes, other studies [6][7][8] attempted to employ light-weighted sensors (such as accelerator, gyroscope, and magnetometer) to capture the features of various transportation patterns.

Most of the current transportation mode detection systems employ traditional machine learning algorithms, including Support Vector Machine (SVM) [6], Hidden Markov Model (HMM) [7], Adaptive Boosting (AdaBoost) [8]. The accuracies using these machine learning algorithms depend on the discriminability of artificial feature extraction. Thus the professional knowledge affects the accuracy of transportation mode detection [9].

As a machine learning method independent of artificially extracting features, deep learning has succeeded in many complex nonlinear classification problems by automatically extracting high-level features [10], such as computer vision [11], natural language processing [12], and text processing [13].

Most studies reported that the neural network is able to learn how to extract deep features from the large-scale data, and performs better than traditional machine learning algorithms [14]. However, few attempts of applying neural networks in the field of transportation mode detection. This kind of approach is still at primary stage and most of the current deep learning-based approaches are limited to common Deep Neural Network (DNN), while the other networks are rarely exploited.

In this paper, a transportation mode detection method without extra infrastructure is proposed, which is robust against device placement and environments. The main contributions of this paper are summarized as follows:

Our algorithm is based on multi-source sensor data fusion. Except for the most commonly used sensor for transportation mode detection—accelerometer, we have also gathered data from the gyroscope, magnetometer [15], a barometric pressure sensor and base station information. This kind of sensors proves positive effects on extending the diversity of the data level and helping us to improve the detection result.

The shallow feature vectors will be firstly calculated from the recorded sensor readings [16]. Raw measurements will be preprocessed through a mean filter. Particularly, the constant acceleration that results from the force of gravity should be erased through an algorithm proposed in [8]. Then we extract diversified features not only from the time domain, frequency domain, and statistics domain (which can capture information of high–frequency motion from vehicle's engine and contact between its wheels and road) but also several new properties we design. To reduce the dimension of the input attributes for the classification model, we also carry out feature selection and determine the feature with higher identification through the mapping of the CDF curve.

Under the idea that motorized transportation recordings during user's traveling are similar with context information, we innovatively introduce LSTM (Long Short-Term Memory)[17] learning algorithm as our training model, which can solve artificial long time lag tasks and demonstrate better performance on sequential measurements. Through the data normalization, gradient-based optimization and evaluation, the final results have been determined, which shows that our approach can obtain a robust model and a high accuracy.

II. Related work

GPS or GSM-based transportation mode detection has shown some limitations including modest accuracy in the places where the signals are generally severely shielded during daily activities, or higher power consumed [18][19]. Other approaches attempt GIS supplements [20] but it's prone to the noises brought by neighboring devices. In recent years, several studies suggest that the sensor-based approach is more appropriate and many related works have been developed. There is another approach that Hemminki [5] propose a hierarchical classifier (consisting of the kinematic motorized classifier, motion classifier, and stationary classifier) to identify five transportation modes relying only on accelerometer readings for minimizing the energy consumption of smartphones, which can achieve 85% accuracy typically. This strategy acquires better accuracy, but it takes too much time to calculate the optimal hand-crafted features.

Numerous studies have confirmed that a deep learning mechanism suggests better feature representation capability. Wang et al. [9] leveraged both manual features and deep features obtained from GPS trajectory data. A DNN-based mechanism is adapted to detect transportation and acquire a mediocre accuracy of 74%. Fang et al. [21] proposed a deep learning framework to determine ten transportation attributes based on sequential sensing data obtained from an accelerometer, gyroscope, and magnetometer. This DNN-based issue achieves a better performance through offline training and online testing, which indicates the effectiveness and robustness of the model with approximately 95% accuracy compared with four well-known machine learning algorithms. We believe that it's a convincing research but this work just adopt DNN-based deep learning technique, some other neural networks are worthy of further attempt as we concerned.

The purpose of this paper is to leverage recent advances in crowd sensing and deep learning techniques for a pilot study on transportation mode detection. We decide to adopt LSTM learning into our neural network, which can learn to bridge minimal time lags in great large time steps and store vehicle travel information over extended time intervals. For the purpose of accelerating the convergence speed, the task of data normalization has been measured. During the training process, the proposed networks can efficiently model the nonlinear function between labeled attributes and sequential data by stacking several layers with optimal parameters, which can be obtained through gradient descent and backpropagation [22]. At last, our algorithm computes the most likely transportation mode according to the prediction of the fine-grained classifier.

The approach we propose can meet the requirements of universality and stability since its higher accuracy and lower power consumption than existing representative methods.

III. Proposed method

A. Framework Overview

This section introduces the proposed approach in this study. Fig. 1 illustrates the flowchart of our mechanism. There are five main subtasks in our designed system, including data collection, data preprocessing and shallow feature selection as the data preparation stage, model training, and simulation tests as the classification stage.

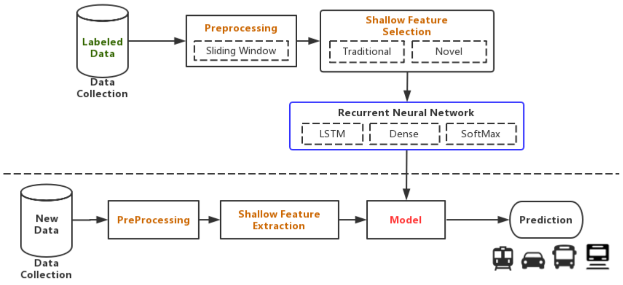

Fig. 1. The framework of the proposed detection system.

The motivation behinds this work is to explore characteristics from multiple sensors and appropriate neural networks to design a robust transportation detection algorithm. After the data collection, we apply the data preprocessing with noise elimination and acceleration decomposition, next we calculate the shallow feature including both traditional eigenvalues and several new properties. Finally, we output the selected measurements which symbolizes the end of the data preparation stage. In the classification stage, 169 shallow features are fed into our LSTM model. Owing to the sequentially of the attributes and different construction of multiple hidden layers, our model can effectively characterize complex mapping functions between input feature vectors and output labels. When testing measurements are transformed into features and fed to the pre-trained models, the detection result can be demonstrated by the confusion matrix and other evaluation indicators.

B. *Data Collection and Preprocessing*

Since Android smartphones perform easy programmability of built-in sensors, data collection is conducted with an Android application designed under a sampling frequency of 100Hz. The dimension of raw data is 11 because of 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, one dimension of barometer pressure and one dimension of base station signal strength. Meanwhile, we add a corresponding label (1, 2, 3 and 4) to each reading which represents four transportations.

To enrich the data dimension of raw measurements from sensors, we adopt a series of data preprocessing methods like windowing technique and noise filtering to calibrate the sensor data. Specifically, we estimate the gravity component of the accelerometer, transform the recorded data from a smartphone's coordinate system to a vehicle's coordinate system as Fig. 2 depicts, decompose the accelerometer measurements into horizontal and vertical representations without the impact of gravity. In order to erase the dirty data and smooth the Gaussian noise, we discard the zero-value data from several phones which miss fractional related sensors, split the recorded reading into a number of smaller data segments and integrate 256 contiguous samples into a frame to extract diversified features in the next step, then we use a sliding window [8] with 50% overlap to smooth data continually. Since the monitoring period of each frame is 2.56 s under a sampling rate of 100Hz, 50% overlap implies we reuse the previous frame with a period of 1.28s to reduce the detection delay and prevent noise interference.

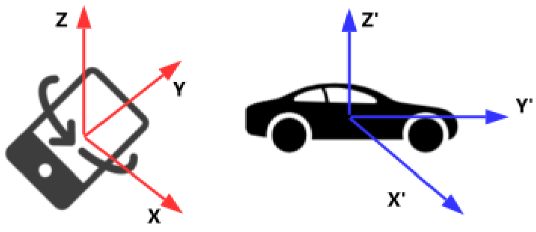

Fig. 2. The difference between a smartphone's coordinate system and a vehicle's coordinate system.

C. *Feature Selection*

In this task, we mainly focus on the behavior feature mining of transportation movement patterns without mobile posture influence. The feature calculation and selection process are undertaken on a frame-by-frame basis. The frame-based features are generated by a moving average strategy to remove the jitter and other noise from the initial measurements.

Firstly, we calculate a multitude of attributes of different levels from sensor data and integrate them into a feature domain, then the feature is filtered [23] by mapping the cumulative distribution function (CDF) which can extract effective feature and abandon useless feature that may lead to misleading detection. Through the analysis of several comparative experiments, we choose 169 features as the input of our neural network. All sets of features are summarized in TABLE I. which are used as independent samples for training or testing.

TABLE I. FEATURE LIST

Sensors	Domain	Features
Horizontal acceleration	Statistical	Mean, STD, variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS
	Time	Integral, Double integral, Auto Correlation, Mean-Crossing Rate
	Frequency	FFT DC, 1, 2, 3, 4, 5, 6 HZ, Spectral Energy, Spectral Entropy, Spectrum peak position, Wavelet Entropy
Vertical acceleration Gyroscope Magnetometer	Statistical	Mean, STD, variance, Median, Min, Max, Range, Interquartile range, Kurtosis, Skewness, RMS
	Time	Integral, Double integral, Auto Correlation, Mean-Crossing Rate
	Frequency	FFT DC, 1, 2, 3, 4, 5, 6 HZ
	Turn	Turn frequency
	Stationary	Stationary duration, frequency
Barometric pressure	Statistical	Mean, STD, variance, Median, Min, Max, Range, Interquartile range
Base station	Innovation	Signal strength

From the table we can see that: in addition to the popular features based on the domain including statistical, time and frequency, we also propose several new measurements.

In order to distinguish motorized transportations in the situation that the vehicle keeps stationary when we travel, two effective measurements have been extracted as stationary features, which is called stationary duration and frequency. Instead of adopting previous stationary identification with the speed information acquired by GPS, we calculate this two properties through a novel algorithm we design, which is realized through the judgment whether the variance value of vertical acceleration on each frame is less than the set threshold or not. Fig. 3 demonstrate that there is a threshold of 0.1 under a great number of experiments we conduct. When the value less than this threshold, the corresponding status and the reality of the marked stationary period is basically consistent.

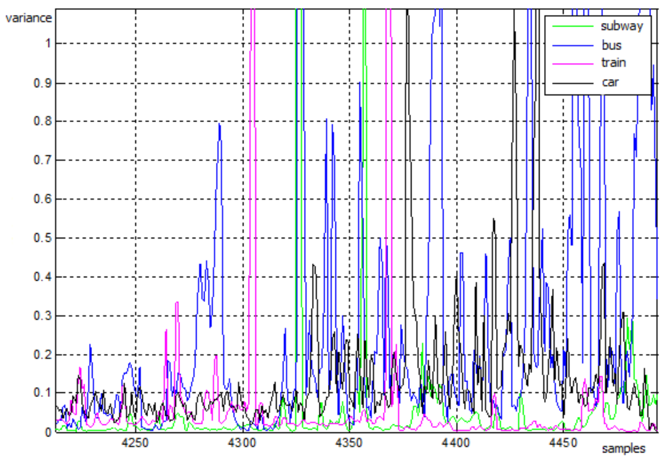

Fig. 3. The Variance of Vertical Acceleration from Different Vehicles.

The turn frequency of different transportation modes is also distinguishable since the degree of distinction is relatively obvious. This paper proposes a turn detection algorithm. Therefore, we can record the amount of turn behavior and capture the different turn frequency of those vehicles based on a fixed time step and regard this frequency as a feature. In this algorithm we calculate the turning angle by using accelerometer and gyroscope readings with the following formula:

$$angle = \frac{gyr \bullet gravity}{\sqrt{gravity_x^2 + gravity_y^2 + gravity_z^2}} * \Delta T \qquad (1)$$

We adopt the sum of turning angle attribute in a sliding window of 20 frames and compare it with a fixed threshold. If the calculated result is beyond the threshold, the corresponding frame is considered to be a turning status, and we conclude the current vehicle has resulted in a turn behavior. Meanwhile, we consider that the user's behavior, such as rotating the phone, will affect the outcome of our turn detection seriously. This issue can be solved through another threshold to filter user's interference behavior.

Moreover, we also introduce base station information since it can transmit a message to the mobile phone terminal in a range of radio coverage area. Since the signal strength can be obtained through the Android smartphone, and the switching frequency of base station strength is identifiable among different transportations (Especially trains), we take this characteristic into account by calculating the difference between the maximum and minimum signal strength of the base station between adjacent frames.

After all this process for our feature data, we still notice that the imbalance of data and consider that it may cause the deviation of classification. So the final step in the data preparation stage is to ensure the balance of our dataset for each vehicle by setting a threshold which is 80% of the smallest sample number for the four categories.

D. Normalization

In order to generalize the size and distribution of the feature measurements, speed up the convergence of the model through a gradient descent algorithm, we choose Z-score Normalization to standardize the selected feature vectors.

Before fed these features into the recurrent neural network, this issue raises an important point about the influence of serialized data processing. Since the LSTM layer needs a fixed time step which will be spanned through a sequence learning algorithm seeking to model the following measurement after this time block, we regard five samples as a time step and reshape the input data into a proper form. Our work with this novel sample construction implies a long-range sequential dependency behind the series of vehicle measurements.

E. Long Short-Term Memory Architecture

1) Network Structure

We take LSTMs into account because they are responsible for many state-of-the-art sequence modeling results. Fig. 4 illustrates our LSTM structure. Each input feature sample with a shape of 5×169 represents a series of sequential transportation attributes. We fed them into the model and stack an LSTM layer as our first layer, which consists of a set of memory blocks. Then we add a fully connected layer with 32 neurons to enhance the recognition accuracy of the model. The softmax [24] classifier is used as the top layer of the model to identify four kinds of transportation modes. Besides, there is a dropout layer stock between LSTM layer and dense layer so as to prevent overfitting.

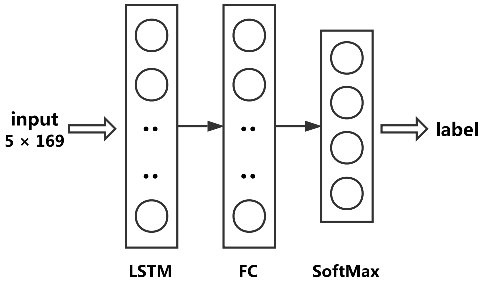

Fig. 4. The Network Structure.

2) Model Training and Optimization

To build our LSTM network with a set of training data and its corresponding output labels, we formulate a cross entropy cost function [25] as:

$$C = -\frac{1}{n}\sum_{x}[y\ln a + (1-y)\ln(1-a)] \qquad (2)$$

where *y* denotes the true value, and *a* denotes the predictive output of the neuron. Cross-entropy is commonly used in multi-class classification to find the error in prediction.

Unlike standard RNNs, there are three multiplicative units including the input gate, output gate and forget gates in each memory block, which prompt LSTM memory cells to store and access the vehicle 's travel information over long periods of time. Through this gating mechanism and self-connected memory cells, the network will decide how much of the input state should be "remembered" or "forgotten", which allows past information of motorized transportation modality to be reinjected at a later time [17].

In this study, we adopt the backpropagation with Adam [26] optimization algorithm to fine tune the parameters. Adam (Adaptive Moment Estimation) computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. In this way, the training process will not be stuck in a saddle, which can more effectively fine tune the model parameters and need little memory requirements. Besides, we adopt the dropout [27] mask (which randomly picks visible and hidden neurons to drop from the network) at each time step inside the LSTM layer. It's possible to obtain more characteristic expression and each neuron will not depend entirely on other neurons.

3) Complexity

The LSTM learning algorithm is local in time and space [28]. we assume that each memory blocks have the same size and gates operation do not possess outgoing connections, the LSTM's computational complexity on each time step and weight is $O(1)$. While storage complexity per weight is still $O(1)$ because it does not rely on the length of our input vector.

F. Algorithm Flow

Consider that our dataset is based on sequential data like language or speech, we choose the LSTM learning algorithm to model our vehicle measurements. The overall algorithm flow we implement according to our proposed approach is demonstrated in Fig. 5.

```
Alg. 1  Transportation mode Detection with LSTM Learning
Start sensor data sampling, frame size = $m$
Preprocessing with a sliding window
Repeat :
   calculate features for each frame with 50% overlap
   store the corresponding feature vector
sf ← form shallow feature vectors
Normalize sf measurements
Input sf, reshape_size = $time\_step \times n(\#feature)$
Repeat :
      **Forward Propagation:**
      lstm ← LSTM(sf)
      Dropout
      fc ← Fully_connected(lstm, sigmoid)
      class label ← softmax(fc)
      **Backward Propagation**
Until loss convergences
```

Fig. 5. Algorithm Flow of the proposed system.

Firstly, we trigger our whole algorithm with data sampling. Through preprocessing with a sliding window we acquire 169 feature measurements including several new attributes like stationary duration, turn frequency and signal strength of base station. These vectors have been normalized and reshaped before fed into the network so as to adjust the proper input form of the LSTM layer. At the starting point of LSTM learning, we employ Xavier to initialize trainable parameters. Next step is to gradually adjust these weights based on a training loop. In detail, we run our network on a batch with 128 training samples and obtain predictions through forward propagation, then compute the loss of the network on this batch. Finally, we update all weights of this network through back propagation in an exact way (Adam) to slightly reduce the loss. We eventually end up with the network until we find the proper parameters and the loss convergences.

IV. Experiment results and analysis

A. Dataset and Experimental Equipment

The datasets were collected from an Android-based application we have developed, then the feature selection is completed through a Matlab script. Finally, the model training tasks and simulation tests are both done offline on PC based on a Python project.

To obtain a sufficiently large, diverse and balanced dataset to train and test our model, we start our collection since 2016 over two years, involving 58 volunteers and containing more than 500 hours' data with four transportation modes from daily collection through different Android smartphones (Huawei mate8, Samsung S6, and MI Note2, etc.), which are both equipped with needed sensors including an accelerometer, gyroscope, magnetometer, a barometric pressure sensor and a SIM card to collect base station information. We recommend the volunteers to freely place the smartphones in multiple ways, such as putting it into a bag, trouser pockets, on the seat, or just being held in hand, which is aiming to evaluate whether our algorithm is sensitive to the placement of the sensor. The collection scene consists of several urban and suburban areas in Beijing, Tianjin, Shanghai, Shenzhen, Hangzhou and other cities, covering a variety of road tracks and different traffic condition so that we can ensure a strong scene universality of the training classifier we build and avoid the problem of poor generalizability caused by data sources from similar traffic tracks.

There are totally 230,192 samples and the specific number of each transportation modes are illustrated in TABLE II. The use of massive data in this work makes the result more general and convincing. We divide the dataset into a training set, a validation set and a testing set with a ratio of 6:2:2. The validation set is used to evaluate the performance of the model during the training process, not to participate in parameter adjustments and weight updating. For the purpose of avoiding the contingency of experimental results, we carry out totally 20 experimental runs on the samples for training, validating and testing with randomly selected, afterward we evaluate the average accuracy of the pre-trained model.

TABLE II. Dataset distribution

Label	**Sample Number**
Bus	53778
Car	56769
Subway	61603
Train	58042

Moreover, we prepare separate new datasets and we will use both internal and external data to assess the generalization capability of our framework.

B. *Model Training Process*

The experiments are performed using Keras [29] which is a lightweight deep learning framework built on TensorFlow and its highly modularized neural network programming with python provides considerable convenience. All the training and testing tasks like loading data, initialization, backpropagation and updating parameters are under CPU mode. Our experiments are completed on a PC with Python2.7, Ubuntu 16.04, Intel i5-4440, 8GB RAM.

The model training task is usually done with empirical approaches. We eventually set the batch size of 128 and 10 epochs during the training process.

In our model, we define the network structure by combining one LSTM layer with one fully connected layer and constructing the optimized network structure in terms of the performance under various numbers of neurons and results of applying different sets of parameters. Faced with a multi-class classification task in deep learning, we evaluate the Cross-Entropy Error (CEE) of the network with different parameters. Through the results in TABLE III. we stack an LSTM layer with 50 neurons and a fully connected layer with 32 neurons, a softmax function (which conducts a multinomial logistic regression that is paired with cross entropy) is placed on the top of the hidden layer to perform classification with four transportation modes.

TABLE III. CEE OF DIFFERENT LSTM STRUCTURE

Input	LSTM	Dense	Softmax	CEE
169	64	64	4	0.0612
169	50	64	4	0.0428
169	50	32	4	0.0367
169	50	None	4	0.0595
169	32	None	4	0.0912
169	32	32	4	0.0430

In addition, we set a dropout layer between the LSTM layer and Dense. The tanh function is adopted as the activation function in LSTM, while in the fully connected layer is sigmoid. We also utilize 10-fold cross-validation to tune the model and lead to a more reliable result. Fig. 6 and Fig. 7 manifest the accuracy and error rate in the training process. We can observe that the model becomes generally converged after 10 epochs. Although there is a slight fluctuation in the validation set, the overall accuracy is more than 98% during the training task.

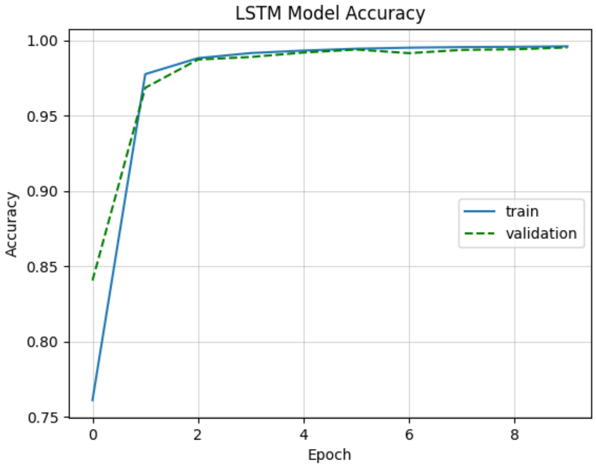

Fig. 6. Train and Validation accuracy of the model.

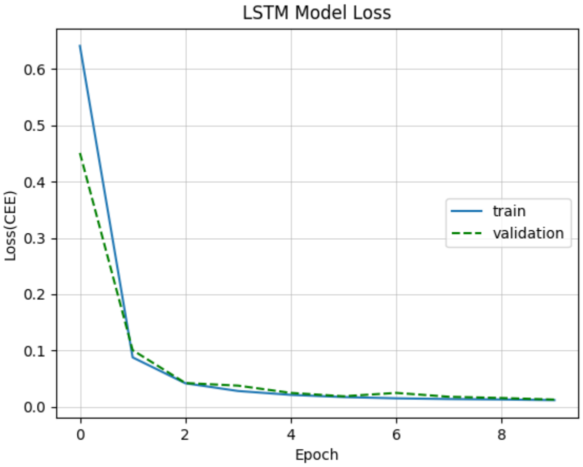

Fig. 7. Train and Validation loss of the model.

C. *Prediction Evaluation*

We measured the performance of our model with classification accuracy and confusion matrix. Furthermore, the F1 score [30] which represent a harmonic mean of precision and recall has also been estimated. TABLE IV. illustrates our result on this different evaluation indicators for the pre-trained LSTM model.

TABLE IV. MODEL EVALUATION USING PRECISION, RECALL, AND F1-SCORE

	precision	recall	F1-score
bus	0.93	0.99	0.96
car	0.97	0.98	0.97
train	0.99	0.97	0.98
subway	0.96	0.94	0.95

Fig. 8 demonstrate that our proposed model using LSTM outperforms the other two machine learning model (AdaBoost and SVM) in general.

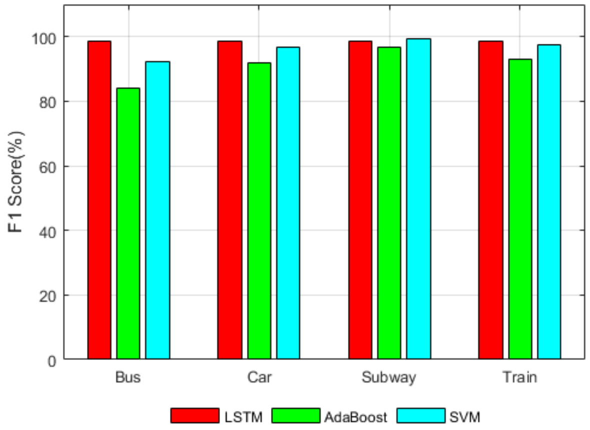

Fig. 8. Model Evaluation by F1 Score using LSTM, AdaBoost, and SVM.

To strengthen the classification ability of our proposed algorithm, we do not just shuffle all samples together because this kind of evaluation using single dataset deviates from the real-world situation and it will not reveal the generalizability and robustness of the model. In another way, we ask the system providing additional sensor data for the new sample, which is closer to real-world life and can help us identify the stability of our detector while facing with the new measurements the model has never seen.

Though the accuracy of our presented transportation mode detection method decreases by a small percentage, it still achieved nearly 92.3% accuracy on the test dataset, as TABLE V. depicts. From the confusion matrix, we can also observe that cars and buses are almost accurately differentiated, trains and subways are prone to be confused because of the high similarity of these modes. Moreover, there are a number of bus samples have been misjudged as subway, we consider that a portion of new bus data are collected from electric buses during recent collection and its motivation mode are almost as smooth as the subway.

TABLE V. CONFUSION MATRIX OF PREDICTIONS

Predicted / Actual	Bus	Car	Train	Subway
Bus	12051	160	881	6744
Car	16	32299	0	10
Train	58	7	39327	297
Subway	103	29	212	18376

This issue also makes an effort to GRU (Gated Recurrent Unit) model [31] which is another variant of the recurrent neural network. We compare AUC (Area Under the Curve) between LSTM and GRU by mapping ROC (Receiver Operating Characteristic) curve [32] generated to measure the sensitivity versus specificity. Fig. 9 shows the average ROC curve and the corresponding AUC value can both achieve more than 90% of these two RNN models, which reflects favorable generalization performance. We can also see that GRU get a better outcome and we will keep GRU-based experiment in our future work.

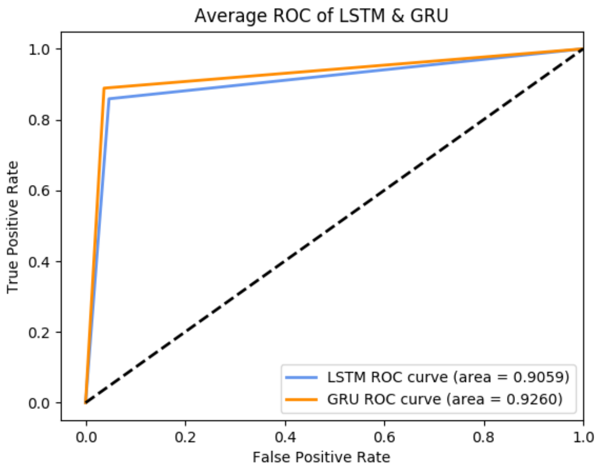

Fig. 9. Average ROC curve using LSTM and GRU.

D. Computation Complexity

In our experiment, the model training and simulation testing are undertaken offline on PC. On the basis of the same dataset, the computation cost of training among LSTM, AdaBoost, and SVM are shown in TABLE VI.

TABLE VI. COMPUTATION COST OF THE DIFFERENT MODEL

Model	*Training Time(s)*	*Testing Time(s)*
LSTM	106	5
AdaBoost	1124	57
SVM	118	5

We compare the execution time of this three algorithms so as to address practical issues. It can be observed that LSTM performs a better training speed, meanwhile, it also shows a higher detection accuracy.

V. Conclusion

This paper proposes a deep learning method for detecting transportation mode from smartphone sensor data and proves reliable and robust detection with fine-grained motorized transportation while mobile users are traveling. The extensive experimental results confirm the effectiveness of our sensor-based approach, which achieves 96.9% classification accuracy and outperforms traditional machine learning methods. In addition to average accuracy, this paper also evaluates the computation cost of the LSTM model to address practical measurements.

Our studies are able to meet the requirement of low-power consumption and high-accuracy, which will be used with great success in the challenge for urban planning and traffic management purpose.

There are still several issues for the future work. On one hand, we will focus on the transportation that is most often misclassified and enrich more transportation modes to further improve the diversification of our detection. On the other hand, we will look for other public datasets instead of our own collected data, because we consider that public datasets could be used to compare the results with other approaches much more extensive.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program (2016YFB0502004), the National Natural Science Foundation of China (61374214), and the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

REFERENCES

[1] Song C, Qu Z, Blumm N, et al. Limits of predictability in human mobility[J]. Science, 2010, 327(5968): 1018-1021.

[2] Froehlich J, Dillahunt T, Klasnja P, et al. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits[C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2009: 1043-1052.

[3] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, "A survey on approaches of motion mode recognition using sensors," IEEE Transactions on Intelligent Transportation Systems, 2016.

[4] Bloch A, Erdin R, Meyer S, et al. Battery-Efficient Transportation Mode Detection on Mobile Devices[C]//Mobile Data Management (MDM), 2015 16th IEEE International Conference on. IEEE, 2015, 1: 185-190.

[5] Hemminki S, Nurmi P, Tarkoma S. Accelerometer-based transportation mode detection on smartphones[C]//Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems. ACM, 2013: 13.

[6] Chen Shaomeng, Luo Haiyong, Zhao Fang, Yuan Weichao, Jiang Mengling. Transportation Mode Recognition Algorithm based on Multiple Support Vector Machine Classifiers. 2017 5th International Conference on Enterprise Systems. September 22-24, Beijing, China, pp. 253-259.

[7] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. ACM Transactions on Sensor Networks, 6(2):1–27, 2010.

[8] Qin Yanjun, Jiang Mengling, Yuan Weichao, Chen Shaomeng, Luo Haiyong. Transportation Mode Recognition Algorithm based on Bayesian Voting. 2017 5th International Conference on Enterprise Systems. September 22-24, Beijing, China, pp. 260-269.

[9] Wang H, Liu G J, Duan J, et al. Detecting Transportation Modes Using Deep Neural Network[J]. Ieice Transactions on Information & Systems, 2017:1132-1135.

[10] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436.

[11] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[12] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 160-167.

[13] Ruiz M E, Srinivasan P. Hierarchical text categorization using neural networks[J]. Information Retrieval, 2002, 5(1): 87-118.

[14] Um T T, Babakeshizadeh V, Kulić D. Exercise motion classification from large-scale wearable sensor data using convolutional neural networks[C]//Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE, 2017: 2385-2390.

[15] Chen K Y, Shah R C, Huang J, et al. Mago: Mode of Transport Inference Using the Hall-Effect Magnetic Sensor and Accelerometer[J]. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2017, 1(2): 8.

[16] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.

[17] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[18] Ashbrook D, Starner T. Using GPS to learn significant locations and predict movement across multiple users[J]. Personal and Ubiquitous Computing, 2003, 7(5): 275-286.

[19] Mun M, Estrin D, Burke J, et al. Parsimonious mobility classification using GSM and WiFi traces[C]//Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets). 2008.

[20] Stenneth L, Wolfson O, Yu P S, et al. Transportation mode detection using mobile phones and GIS information[C]// ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Acm-Gis 2011, November 1-4, 2011, Chicago, Il, Usa, Proceedings. DBLP, 2011:54-63.

[21] Fang S H, Fei Y X, Xu Z, et al. Learning Transportation Modes From Smartphone Sensors Based on Deep Neural Network [J]. IEEE Sensors Journal, 2017, 17(18): 6111-6118.

[22] Horikawa S I, Furuhashi T, Uchikawa Y. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm[J]. IEEE transactions on Neural Networks, 1992, 3(5): 801-806.

[23] Gong Yanyun, Zhao Fang, Chen Shaomeng, Luo Haiyong. A Convolutional Neural Networks based Transportation Mode Identification Algorithm. IPIN2017. September 18-21, Sapporo, Japan.

[24] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on. IEEE, 2013: 6645-6649.

[25] Blossom P. The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning[J]. 2006.

[26] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[27] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[28] D'Informatique D E, Ese N, Esent P, et al. Long Short-Term Memory in Recurrent Neural Networks[J]. Epfl, 2001, 9(8):1735 - 1780.

[29] Chollet F. Deep learning with Python[M]. Manning Publications, 2018.

[30] Powers D M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation[J]. 2011.

[31] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[32] Pencina M J, D'Agostino R B, Vasan R S. Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond[J]. Statistics in medicine, 2008, 27(2): 157-172.