Learning Transportation Modes from Smartphone Sensors Based on Deep Neural Network

Shih-Hau Fang, *Senior Member, IEEE*, Yu-Xaing Fei, Zhezhuang Xu, and Yu Tsao, *Member, IEEE*

***Abstract*—In recent years, the importance of user information has increased rapidly for context-aware applications. This study proposes a deep learning mechanism to identify the transportation modes of smartphone users. The proposed mechanism is evaluated on a database that contains more than one thousand hours of accelerometer, magnetometer, and gyroscope measurements from five transportation modes including still, walk, run, bike, and vehicle. Experimental results confirm the effectiveness of the proposed mechanism, which achieves approximately 95% classification accuracy and outperforms four well-known machine learning methods. Meanwhile, we investigated the model size and execution time of different algorithms to address practical issues.**

***Index Terms*—Transportation mode, big data, deep learning, mobile phone, sensors.**

I. Introduction

Transportation modes are essential for providing mobile users with various value-added services. For example, using the online mode of a phone, advertising companies can send messages to desired users in a specific state for behavioral targeting. The knowledge of individuals' mode of transport can further facilitate tasks such as urban transportation planning, physical activities, and health monitoring [1]–[3]. With increasing demand for various applications, effective transportation mode classification techniques have become necessary, though challenging [4], [5].

This issue has been studied extensively based on feedback from the Global Positioning System (GPS) [6]. However, GPS requires line-of-sight between devices and satellites. Thus, it is not suitable for indoor or urban environments [7] [8]. In addition, GPS-based methods consume considerable power and may not work effectively for activities such as running and walking [9] [10]. Although many works study transportation modes classification, most of them rely on the GPS data or wireless network information.

In recent years, several techniques have been developed based on sensors embedded in smartphones, such as accelerometer, magnetometer, gyroscope, and atmospheric pressure [11]–[14]. Using these sensors has been recognized as a good approach for determining body posture and motion modes [15]–[21]. However, studies of sensor-based approaches on classification of transportation modes are limited. Most existing sensor-based works focus on user behaviors such as walking, running, jumping, and the like [22], [23]. Nham et al. [24] obtained the accelerometer of an iPhone and used its magnitude to classify transportation modes. Hemminki et al. [25] proposed a kinematic motion classifier to detect five transportation modes including bus, train, metro, tram, and car. Yu et al. [26] extracted features from three sensors with minimum power consumption and proposed support vector machines (SVMs) as the best classifier. Similar conclusions and enhanced features can be found in [1], [27].

In recent years, the concept of deep learning has attracted considerable attention. Numerous studies have confirmed that a deep learning model, formed by stacking several layers of shallow structures, has better feature representation capability and accordingly could more effectively deal with nonlinear and high complexity tasks [28] [29] [30]. Modeling big data using a deep learning model has been performed extensively and is recognized in several applications such as traffic flow prediction [31], speech enhancement [32] [33], and vehicle type classification [34], In this study, we propose a deep learning framework to efficiently learn the transportation mode of a mobile phone from sequential sensing data. The data were obtained from three sensors, i.e., accelerometer, magnetometer, and gyroscope, which are available in most current smartphones. The large-scale database used in this study was built by HTC Corporation, containing more than one thousand hours of sensor data with ten transportation attributes including still, walk, run, bike, motorcycle, car, bus, metro, train, and high speed rail (HSR).

We first extract the temporal sequences of highly dimensional and heterogeneous sensor data into a feature domain. Then, by stacking several layers of neurons with optimized weights, the proposed deep neural network (DNN) can efficiently model the nonlinear function between sequential sensing data and labelled attributes. Thus, it can accurately learn the transportation mode of smartphones. Two types of features, feature sets A and B, were adopted to test recognition performance, where feature set B was designed for saving power and has a relatively lower dimension than that of feature set A [26], [27]. Experiment results first confirm the effectiveness of the proposed deep learning approach, which can achieve approximately 95% classification accuracy on the task. Moreover, the results show that when compared with four well-known methods, i.e., AdaBoost, decision trees (DT), K-nearest neighbors (KNN), and SVM, the proposed algorithm can achieve with detection accuracy improvements of 8.00%, 7.86%, 1.82%, and 4.17%, respectively, using feature set A, and 3.28%, 2.73%, 1.18%, and 1.06%, respectively, using feature set B. In addition to accuracy, this study investigated

Shih-Hau Fang and Yu-Xaing Fei are with the Department of Electrical Engineering and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Taiwan (Email: shfang@saturn.yzu.edu.tw and s1044638@mail.yzu.edu.tw). Zhezhuang Xu is with the School of Electrical Engineering and Automation, Fuzhou University (zzxu@fzu.edu.cn). Yu Tsao is with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan (yu.tsao@citi.sinica.edu.tw).

the model size and execution time of different algorithms to address practical issues.

II. Proposed Algorithm

A. Database Description

The data were provided by HTC company, and were collected since 2012 over two years, involving 224 volunteers and containing 8311 hours of 100 GB data. The data used in this study where a part of the raw data, approximately 20 GB in size, which HTC made public for academic use [26]. The group of participants sufficiently covered different genders (60% male), builds, and ages (20 to 63 years old). The transportation states included ten modes, i.e., still, walk, run, bike, motorcycle, car, bus, metro, train, and HSR. Compared to other studies that use small-scale data (several or dozens of hours), the use of big data in this study makes the results more convincing and general. The database for five transportation modes is given in Table I. We would like to clarify that we do not have the right to public the dataset directly because the owner is in fact HTC (a Taiwan company). Due to the big data contest (IEEE BigMM 2016 HTC Challenge), we have signed the contract which allowed us to use the data for academic research only.

TABLE I
Database Description

Transportation Mode		Collection Time (hour)
Still		158
Walking		141
Running		79
Biking		98
Vehicle	Motorcycle	163
	Car	208
	Bus	75
	Metro	132
	HSR	106

In this paper, we attempt to visualize the sensing measurements from different perspectives. Figs. 1 and 2 show the raw data from x-axis of three sensors during 10 seconds and 10 minutes, respectively. Fig. 3 shows the averaged distribution of Fig. 2, which is randomly selected 10 min, where the window size is 512. These figures show that the long-term statistic is slightly different from that of the raw data, verifying the importance of the temporal processing in the features.

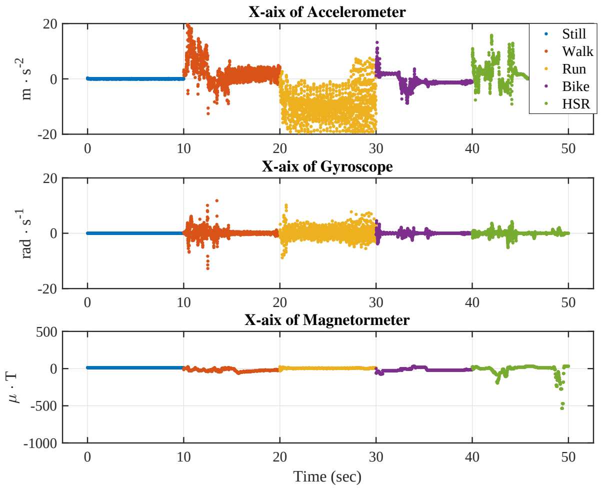

Fig. 1. Distribution of 10 seconds raw data from three sensors (x-axis).

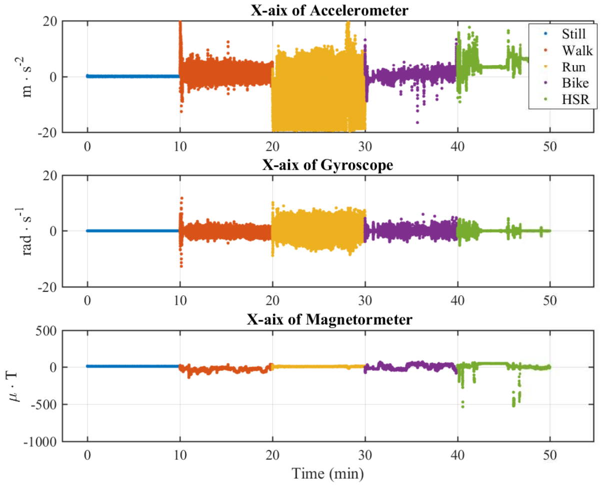

Fig. 2. Distribution of 10 minutes raw data from three sensors (x-axis)

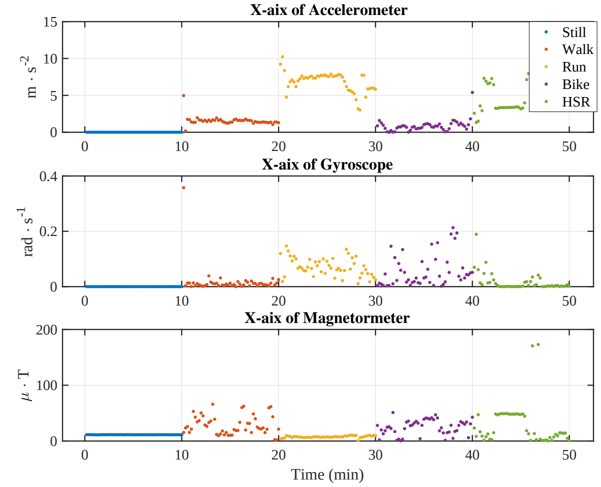

Fig. 3. Distribution of 10 minutes averaged data from three sensors (x-axis).

B. Conventional Neural Network with One Hidden Layer

We first review the neural network model with one hidden layer. The model contains an input layer, a hidden layer, and an output layer [35], [36]. Multiple interconnected neurons are involved in parallel to nonlinearly map sensor data (input layer) to the transportation mode (output layer) as

$$\hat{\boldsymbol{y}} = f\left(\boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}\right), \tag{1}$$

where $\hat{\boldsymbol{y}}$ denotes the output vector (estimated mode), $\boldsymbol{x}$ represents the input vector (sensor measurements), $f(\cdot)$ is the activation function, $\boldsymbol{b}$ is a bias vector, and $\boldsymbol{w}$ is a weight vector. The weight and bias vectors can be trained using a typical gradient-based algorithm. There are several neural network variations [37]. For example, the focused time delay neural network is a well-known solution that uses ordinary time delays to perform temporal processing. The primary idea is to transform a static neural network into a dynamic one, in which tapped delay lines are at the input layer [38]. In recent years, incorporating multiple layers has exhibited excellent performances in multiple big data and machine learning tasks. The concept is to stack more hidden layers to strengthen classification capability.

C. Proposed DNN-based Mechanism

This section introduces the proposed mechanism that can learn the transportation mode of a user from sequential sensing data. The motivation behind this study is to explore sequential information from multiple sensors and a DNN to design an enhanced transportation mode classification system. Fig. 4 shows the flowchart of the proposed mechanism, in which the estimation process can be divided into two phases, i.e., offline and online. During the offline phase, feature extraction is conducted to integrate sensor data into diversified features to train a DNN model. During the online phase, the proposed system transforms testing data to feature vectors, and feeds it to the previously trained DNN model to compute the most likely transportation mode.

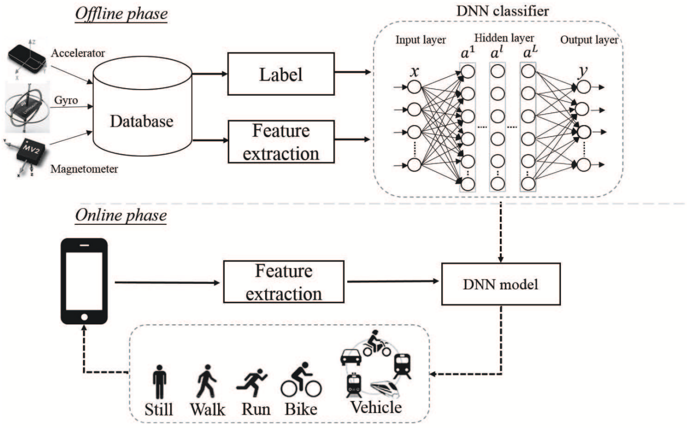

Fig. 4. Flowchart of the proposed DNN-based transportation mode learning system.

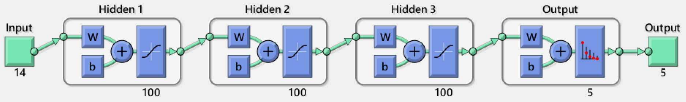

Fig. 5. Structure of the proposed DNN-based transportation mode classifier with three hidden layers.

Assuming that the temporal sequence, s_i, consists of p unit delay operators, then, the temporal sequence at time n is the signal vector, $\boldsymbol{s_i}(n) = [s_i(n), s_i(n-1), \cdots, s_i(n-p)]^T$, where i is the index of the input dimensions. The dimension of raw data is nine in this work because of three directional dimensions (x-axis, y-axis, and z-axis) from three sensors (accelerometer, magnetometer, and gyroscope). p depends on several factors such as sampling rate of devices and delay tolerance of an application.

In this study, we integrate p temporal samples into a frame to extract features, and use a slide-moving window with acceptable overlap to smooth data continuity and reduce system delay. Given the nine input signals consisting of the present value, $s_i(n)$, and the past values of order p, we follow the same procedure as that of [26] [27] to derive two distinct feature sets, $\boldsymbol{x}$, based on the statistics of the temporal sequence. For simplicity in notation, we use $\boldsymbol{x}$ to represent the feature vector in the following derivation. Different features have different limitations and advantages. For example, the most power-saving feature is proposed in [26], while [27] extracts better discriminant features. As $\boldsymbol{x}$ contains past sensor data of order p, the feature parameters reflect the temporal structures of the input signal and are regarded as input vectors to train the DNN model, as shown in Fig. 4.

A DNN model is a feed-forward artificial neural network model that consists of multiple hidden layers. Each hidden layer contains several neurons, which are fully connected to the neurons in the next hidden layer. Owing to the multiple hidden layers, a DNN can effectively characterize complex mapping functions between input feature vectors and output labels.

For a DNN model with L hidden layers, the output of a hidden layer can be described as

$$\boldsymbol{a}^l = f(\boldsymbol{w}^l\boldsymbol{a}^{l-1} + \boldsymbol{b}^l), \tag{2}$$

where $l = 1, ..., L$, $\boldsymbol{a}^0 = \boldsymbol{x}$ is the input layer with feature vector $\boldsymbol{x}$. $\boldsymbol{w}^l$ and $\boldsymbol{b}^l$ denote the neural weight matrix and bias of the l-th hidden layer, respectively. $f(\cdot)$ is a nonlinear active function (element-wise transform), such as the sigmoid function, tanh function, and the rectified linear units (ReLU) function [39]. In this study, the ReLU function is adopted as the activation function, which can be expressed as

$$f(z) = max(0, z) \tag{3}$$

Then, another function is placed on top of the L^{th} hidden layer to perform classification, for which the softmax function was adopted in this study as follows

$$\hat{y}_{i,j} = \frac{e^{a_j^L}}{\sum_d e^{a_d^L}} \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2017.2737825, IEEE Sensors Journal

where a_j^L and a_d^L are the jth and dth elements of $\boldsymbol{a}^L$, respectively, $\hat{y}_{i,j}$ denotes the jth element $\hat{\boldsymbol{y}}_i$, and $\hat{\boldsymbol{y}}_i$ is the output for the ith input data, $\boldsymbol{x}_i$.

To train the model parameters in the DNN model with a set of training data, $\boldsymbol{X} = [\boldsymbol{x}_1, .., \boldsymbol{x}_i, ..\boldsymbol{x}_I]$, and the corresponding output labels, $\boldsymbol{Y} = [\boldsymbol{y}_1, ..\boldsymbol{y}_i, ..\boldsymbol{y}_I]$, where I is the number of training samples, we formulate a cost function as

$$C(\boldsymbol{Y}, \hat{\boldsymbol{Y}}, \boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{IJ}\sum_i \sum_j y_{i,j} \log(\hat{y}_{i,j}), \tag{5}$$

where $\boldsymbol{\theta}$ denotes the model parameters, $\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, ..\hat{\boldsymbol{y}}_i, .., \hat{\boldsymbol{y}}_I]$ is the DNN output; $y_{i,j}$ and $\hat{y}_{i,j}$ denote the jth element of $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$, respectively. In this study, we adopt the back-propagation algorithm with the AdaGrad optimization to fine tune the parameters. The weight updates during back-propagation can be denoted as

$$w_{ij}(n+1) = w_{ij}(n) + \mu \frac{\partial C}{\partial w_{ij}} \tag{6}$$

where C is the cost function ($C(\boldsymbol{Y}, \hat{\boldsymbol{Y}}, \boldsymbol{X}, \boldsymbol{\theta})$ in (5)), and μ is the step size [40]. Fig. 5 shows the structure of the proposed DNN-based transportation mode classifier with three hidden layers (L=3). After training the DNN, the model parameters can be stored in a hand held smartphone or in remote cloud servers. When testing measurements are transformed into features and fed to the pre-trained DNN, the transportation mode can be obtained from the output layer of the model.

AdaGrad (for adaptive gradient algorithm) is a new family of subgradient methods that dy-namically incorporate knowledge of the geometry of the data observed in earlier iterations to perform more informative gradient-based learning [41], [42]. In this way, training procedure will not stuck in a saddle point and can more effectively fine tune the model parame-ters. Experimental results in [41] show that AdaGrad outperforms the traditional stochastic gra-dient descent (SGD) method. More specifically, traditional stochastic gradient descent (SGD) update model parameters by:

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \eta \mathbf{g}_t \tag{7}$$

where $\boldsymbol{\lambda}_{t+1}$ and $\boldsymbol{\lambda}_t$ denote the model parameters (in vectors form), η is a learning rate, and $\mathbf{g}_t$ denotes the gradient of the cost function w.r.t. to the parameters. On the other hand, AdaGrad update model parameters by

$$\mathrm{G}_{i,t} = \sum_{\tau=1}^{t} g_{i,\tau}^2 \tag{8}$$

$$\lambda_{i,t+1} = \lambda_{i,t} - \frac{\eta}{\sqrt{G_{i,t}}} g_{i,t} \tag{9}$$

where $\lambda_{i,t+1}$ and $\lambda_{i,t}$ denote the i-th component of $\boldsymbol{\lambda}_{t+1}$ and $\boldsymbol{\lambda}_t$, η is a scalar, $g_{i,\tau}$ denotes the gradient at iteration τ. More detailed descriptions and derivations of AdaGrad can be found in [41], [42].

III. Experimental Results

A. Experimental Setup

This study classifies the vehicular modes (motorcycle, car, bus, metro, train, and HSR) as a single mode, i.e., on a vehicle. Then, these data are divided into two independent sets, i.e., training (60%) and testing data (40%), for performance evaluation. We follow the same procedure as that of [26] and [27] to derive two distinct feature sets A and B. Specifically, feature set B include the average, standard deviation, highest FFT (Fast Fourier Transform) value and the ratio between the highest and the second-highest FFT value of the accelerometers magnitude. The FFT operation provides the frequency information from the temporal sequence of three sensors. Thus, the feature represents the short-term statistic values in the window. Feature set B also contains the average of the gyroscope and standard deviation of the magnetometer's and gyroscope's magnitudes. Feature set A fetched six notable features based on feature set B and append eight extra features, including the average of X, Y, and Z directions of acceleration, acceleration instantly changes, the magnetometers value, magnetic instantly changes, maximum of the accelerometers magnitude, and the standard deviation of acceleration instantly changes. The window size, p (temporal order), is 512 with 75% overlap to extract the sequential feature vectors of each mode into different machine learning algorithms, including AdaBoost, DT, KNN, SVM, and DNN. The parameters of each algorithm are tuned to maximize general accuracy.

B. Experimental Results

Table II compares the general accuracy, prediction time, and model size of each algorithm based on feature set A. The accuracy represents the the percentage of correct estimations/predictions, the prediction time is the duration for each prediction (microseconds), and the model size is the size of the training model (megabits). The results show that the proposed DNN-based approach outperforms the other four methods in terms of general accuracy, achieving approximately 95% classification accuracy. Table II shows that DT reports the lowest prediction time and the smallest model size. On the other hand, the DNN provides the best performance in terms of accuracy and an acceptable model size. The accuracy of KNN and SVM is comparable to that of DNN. However, they require a significantly increased model size.

Next, Fig. 6 shows the pairwise comparison of the based feature set A in the transportation mode classification tasks. This figure can visualize the ability of each feature in some sense. For example, it seems that the fourth feature outperforms the fifth one in transportation mode classification. It motivates us to use more features. In this study, we select and combine useful features from existing works to form an improved feature Set B. However, due to the constrained power and resources, the increased dimension should be minor.

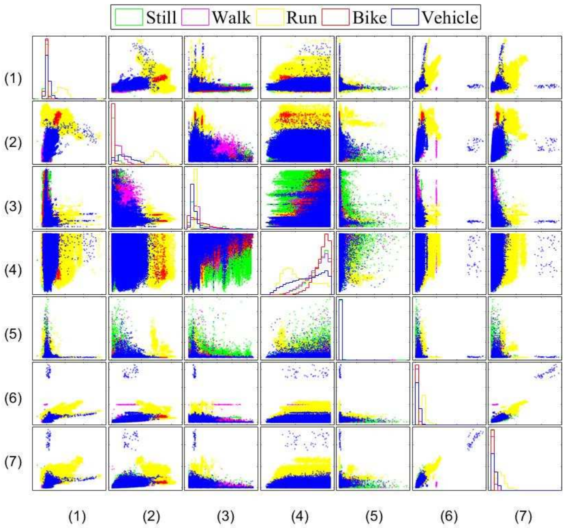

Fig. 6. Pairwise comparison of the traditional features.

Table X compares the performance using an alternative feature set B. This table shows a similar trend as Table II, demonstrating the effectiveness of the proposed mechanism. The results show that the proposed DNN-based mode estimation reports the highest general accuracy among the five algorithms. Note that the accuracy achieved using feature set B is slightly lower than that achieved using feature set A. This is because feature set B was originally designed for saving power, and has a relatively lower dimension (7) than that of feature set A (14). The results in both tables show that the proposed DNN-based approach outperforms AdaBoost, DT, KNN, and SVM, in terms of improving detection accuracy based on two existing features by 3.28% to 8%, 2.73% to 7.86%, 1.18% to 1.82%, and 1.06% to 4.17%, respectively.

TABLE II
PERFORMANCE COMPARISON USING FEATURE SET A.

Algorithm	Prediction	Prediction	Model
	accuracy(%)	time(us)	size(KB)
AdaBoost	87.70	22.3	463
DT (complex)	87.84	0.6	33
KNN (weighted)	93.88	107.4	88986
SVM (RBF)	91.53	7050.0	32154
DNN	95.70	67.2	82

TABLE III
PERFORMANCE COMPARISON USING FEATURE SET B.

Algorithm	Prediction	Prediction	Model
	accuracy(%)	time(us)	size(KB)
AdaBoost	85.15	11.7	137
DT (complex)	85.70	0.5	24
KNN (weighted)	87.25	93.0	88781
SVM (RBF)	87.37	11801.0	28672
DNN	88.43	62.1	81

To analyze the results in detail, the confusion matrices of each algorithm with feature set A were constructed in this study. In these matrices (Table IV to Table VIII), the header columns are the actual labels, and the header rows are the predicted labels. These matrices show the ratios of different mis-attributed errors. The matrices show that in several cases, the bike data were misjudged as walk and vehicle data. In addition, the running mode typically produces the most accurate result, achieving 98.61% and 97.87% accuracy for KNN and DNN, respectively. This may be because running makes the smartphone shake more vigorously, making the dynamic of sensor measurements is discriminative and the classification easier than that for other modes.

TABLE IV
CONFUSION MATRIX OF ADABOOST (GENERAL ACC. = 87.70%, η = 0.1, SPLITS # = 80)

AdaBoost	Still	Walk	Run	Bike	Vehicle
	(132,319)	(144,552)	(59,957)	(87,408)	(799,557)
Still	73.77	0.69	0.26	0.79	24.49
Walk	1.39	80.48	0.14	4.89	13.09
Run	0	4.93	93.50	0.11	1.46
Bike	0.07	21.10	0.04	55.17	23.62
Vehicle	15.63	1.84	0.00	2.16	94.44

TABLE V
CONFUSION MATRIX OF DT (GENERAL ACC. = 87.84%, SPLITS # = 100)

Decision	Still	Walk	Run	Bike	Vehicle
Tree	(132,319)	(144,552)	(59,957)	(87,408)	(799,557)
Still	79.05	0.44	0.26	0.54	19.70
Walk	1.63	80.15	0.18	4.83	13.20
Run	0.00	3.90	94.44	0.21	1.45
Bike	0.18	11.23	0.03	65.84	22.72
Vehicle	2.70	1.71	0.00	2.99	92.60

TABLE VI
CONFUSION MATRIX OF KNN (GENERAL ACC. = 93.88%, K=10.)

K-Nearest	Still	Walk	Run	Bike	Vehicle
Neighbor	(132,319)	(144,552)	(59,957)	(87,408)	(799,557)
Still	91.50	1.64	0.08	0.17	6.60
Walk	0.51	88.90	0.78	1.81	8.00
Run	0.35	0.20	98.61	0.29	0.55
Bike	0.49	2.45	0.10	84.17	12.80
Vehicle	1.20	1.45	0.17	1.32	95.86

TABLE VII
CONFUSION MATRIX OF SVM (GENERAL ACC. = 91.53%, KERNEL IS RBF.)

SVM	Still	Walk	Run	Bike	Vehicle
	(132,319)	(144,552)	(59,957)	(87,408)	(799,557)
Still	81.02	0.31	0.27	0.78	17.62
Walk	1.64	82.10	0.05	3.13	13.08
Run	0.01	2.28	95.89	0.06	1.76
Bike	0.14	4.08	0.01	77.31	18.46
Vehicle	1.68	0.90	0.00	1.21	96.21

TABLE VIII
CONFUSION MATRIX OF DNN (GENERAL ACC. = 95.70%, L=3, NEURONS # =100)

AdaBoost	Still	Walk	Run	Bike	Vehicle
	(132,319)	(144,552)	(59,957)	(87,408)	(799,557)
Still	95.25	0.39	0.26	0.16	3.94
Walk	1.11	91.99	0.25	0.75	5.90
Run	0.02	0.76	97.87	0.19	1.16
Bike	0.27	1.68	0.06	88.60	9.38
Vehicle	0.53	1.78	0.08	0.55	97.05

C. Impact of Different Parameters

This section describes the impact of different parameters of the proposed DNN-based approach for transportation mode recognition. First, from the feature perspective, experiments were conducted to test various window sizes (p) and overlapping ratios. Both parameters affect the general accuracy and the response time, i.e., the latency of each estimation. For example, because the sampling rate of the sensors is 30 Hz, the monitoring period of each frame (p=512) is 17.06 s in a default setup. 75% overlap implies that we reused the previous frame with period 12.8 s as the next frame to reduce system delay. Table IX shows the results of applying the DNN (3 layers with 100 neurons) with three overlapping ratios (25%, 50%, and 75%) on four different window sizes (256, 512, 1024, and 2048). These results show that increasing the window size dose not enhance performance. The highest accuracy is achieved when $p = 1024$. In this study, we set p to be 512 with 75% overlap because this setup balances the latency (4.26 s) and accuracy (95.7%).

TABLE IX
PARAMETERS SELECTION OF FEATURE EXTRACTION.

Performance	Overlap 25%		Overlap 50%		Overlap 75%	
Window size	Acc(%)	Latency(s)	Acc(%)	Latency(s)	Acc(%)	Latency(s)
256	92.87	6.4	93.07	4.26	93.20	2.13
512	93.63	12.8	94.26	8.53	95.70	4.26
1024	96.65	25.6	94.95	17.07	95.06	8.53
2048	93.87	51.2	94.55	34.13	95.31	17.07

Then, from the model perspective, experiments were conducted to test various layers (L) and the number of neurons of the DNN. Fig. 7 shows the impact of the parameters on accuracy and model size. This figure shows that the performance of a 3-layer DNN ($L = 3$) approximates to that of a 4-layer DNN if the number of neurons exceeds 100. However, the model size increases significantly with the number of layers, particularly for the cases where the number of neurons increases. This figure also compares the neural network with one hidden layer (100 neurons) with DNN. Results confirmed that the proposed deep learning framework, formed by stacking several layers of shallow structures, has better capability in dealing with nonlinear and high complexity transportation mode classification tasks. Thus, to balance the model size and accuracy, we select a 3-layer DNN with 100 neurons per layer in this study. To make a fair comparison, Fig. 8, Fig. 9, and Tab. X provide the results of both parameter tuning and prediction results from different machine learning algorithms. For the decision tree and Adaboost methods, we vary the number of splits and the learning rates η, as shown in Fig. 8. Figure 9 compares three kernel functions for SVM, including linear, radial basis function (RBF), the polynomial with order 3 under two features. Table X compares various K values for the K-NN method. These results show that parameter tuning indeed impacts the prediction results. More importantly, the proposed approach still outperforms these approaches in various parameters setup.

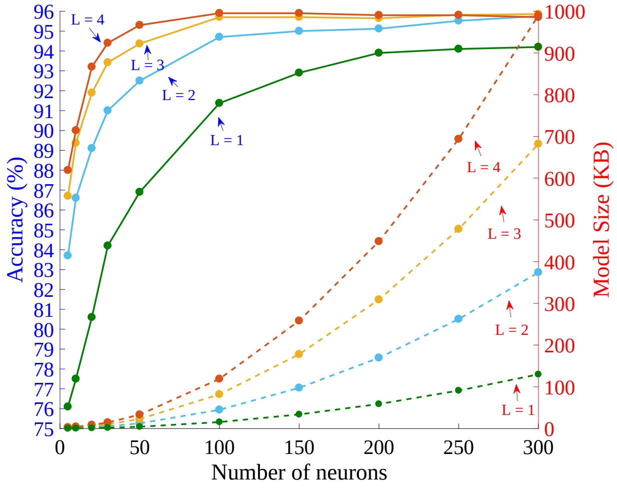

Fig. 7. Parameter selection for DNN model.

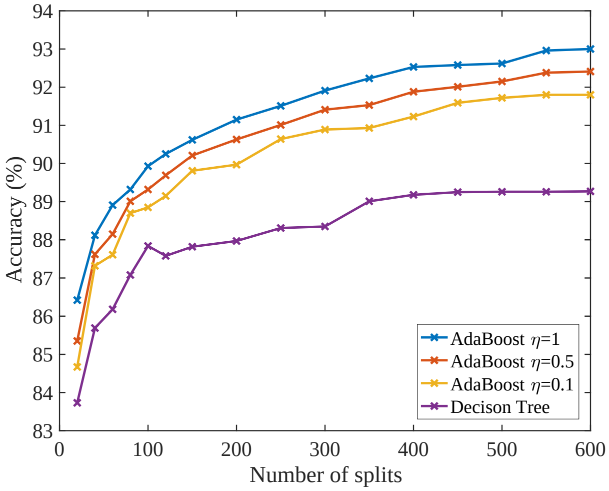

Fig. 8. Parameter selection for AdaBoost and Decision Tree model.

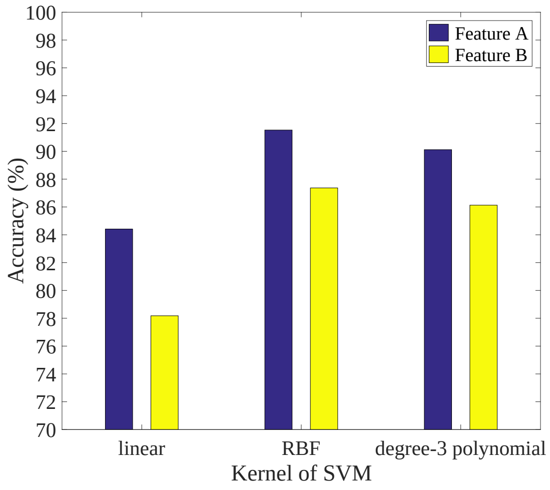

Fig. 9. Kernel selection for SVM model.

TABLE X
PARAMETER SELECTION OF KNN MODEL.

Number of K	Accuracy of set A (%)	Accuracy of set B (%)
1	93.55	84.34
3	93.75	85.94
5	93.85	86.65
7	93.88	86.98
9	93.88	87.18
10	93.88	87.25
20	93.94	87.47
30	93.88	87.46
50	93.91	87.36

IV. CONCLUSION

The aim of this study is to learn the transportation modes of users using a large database that contains more than one thousand hours of sensor data from five transportation modes including still, walk, run, bike, and vehicle. This study proposes a DNN-based approach that stacks several layers of neurons with optimized weights to efficiently recognize the transportation modes. We tested the proposed approach using two distinct feature sets, namely feature sets A and B. Feature set B was designed for saving power and thus has a fewer elements than that of feature set A [26], [27]. The experimental results first confirm the effectiveness of the proposed deep learning approach, which achieves approximately 95% classification accuracy on this transportation mode identification task. The results align with other pattern recognition tasks and suggest that deep learning can be a potential approach for the transportation mode classification task. In addition to accuracy, this study investigated different system parameters from feature and model perspectives to address practical issues including latency and model size. In the future, we will keep collecting training data for the vehicular mode classification task. Meanwhile, since the deep learning models have shown outstanding capability to extract discriminative features, we will also investigate deeper and more complex model to generate features.

ACKNOWLEDGMENT

The authors would like to thank the financial support provided by Ministry of Science and Technology, Taiwan (106-3114-E-007-005 and 105-2221-E-155-013-MY3), NSF of China (Grants 61673116 and 61304260) and NSF of Fujian Province of China (Grants 2017J01500 and 2014J05072).

REFERENCES

[1] 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.

[2] Q. Wei and B. Yang, "Adaptable vehicle detection and speed estimation for changeable urban traffic with anisotropic magnetoresistive sensors," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2021–2028, 2017.

[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] P. Widhalm, P. Nitsche, and N. Brändie, "Transport mode detection with realistic smartphone sensor data," in *International Conference on Pattern Recognition*, pp. 573–576, 2012.

[5] E. Aguirre, P. Lopez-Iturri, L. Azpilicueta, A. Redondo, J. J. Astrain, J. Villadangos, A. Bahillo, A. Perallos, and F. Falcone, "Design and implementation of context aware applications with wireless sensor network support in urban train transportation environments," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 169–178, 2017.

[6] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[7] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of international conference on World Wide Web*, pp. 247–256, 2008.

[8] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on GPS data for web applications," *ACM Transactions on the Web*, vol. 4, no. 1, 2010.

[9] S. Nath, "Ace: exploiting correlation for energy-efficient and continuous context sensing," in *Proceedings of international conference on Mobile systems*, pp. 29–42, 2012.

[10] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Building personal maps from GPS data," *Annals of the New York Academy of Sciences*, vol. 1093, no. 1, pp. 249–265, 2006.

[11] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," *APWCS*, pp. 44–46, 2010.

[12] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *International Conference on Pervasive Computing*, pp. 1–16, 2006.

[13] F. Sikder and D. Sarkar, "Log-sum distance measures and its application to human-activity monitoring and recognition using data from motion sensors," *IEEE Sensors Journal*, vol. 17, no. 14, pp. 4520–4533, 2017.

[14] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*, pp. 1–17, 2004.

[15] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver behavior analysis for safe driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3017–3032, 2015.

[16] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 386–403, 2017.

[17] M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg, "Motion mode recognition for indoor pedestrian navigation using portable devices," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 208–221, 2016.

[18] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[19] R. K. Shen, C. Y. Yang, V. R. L. Shen, and W. C. Chen, "A novel fall prediction system on smartphones," *IEEE Sensors Journal*, vol. 17, no. 6, pp. 1865–1871, 2017.

[20] T.-N. Lin, S.-H. Fang, W.-H. Tseng, C.-W. Lee, and J.-W. Hsieh, "A group-discrimination-based access point selection for wlan fingerprinting localization," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 8, pp. 3967–3976, 2014.

[21] S.-H. Fang and T. Lin, "Principal component localization in indoor wlan environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, 2012.

[22] B. Yamansavalar and M. A. Gvensan, "Activity recognition on smartphones: Efficient sampling rates and window sizes," in *IEEE International Conference on Pervasive Computing and Communication Workshops*, pp. 1–6, 2016.

[23] E. Bber and A. M. Guvensan, "Discriminative time-domain features for activity recognition on a mobile phone," in *IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 1–6, 2014.

[24] B. Nham, K. Siangliulue, and S. Yeung, "Predicting mode of transport from iphone accelerometer data," *Machine Learning Final Projects, Stanford University*, 2008.

[25] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of ACM Conference on Embedded Networked Sensor Systems*, 2013.

[26] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: the design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[27] 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. 1324, 2016.

[28] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[31] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[32] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder.,” in *Interspeech*, pp. 436–440, 2013.

[33] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “An experimental study on speech enhancement based on deep neural networks,” *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[34] Z. Dong, Y. Wu, M. Pei, and Y. Jia, “Vehicle type classification using a semisupervised convolutional neural network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, 2015.

[35] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiely & Sons, 2000.

[36] F. Keinosuke, *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[37] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 2006.

[38] S.-H. Fang, B.-C. Lu, and Y.-T. Hsu, “Learning location from sequential signal strength based on GSM experimental data,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 726 –736, 2012.

[39] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of International Conference on Machine Learning*, pp. 807–814, 2010.

[40] Peter, “How to implement a neural network.” http://peterroelants.github.io/posts/neural_network_implementation_intermezzo02/.

[41] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[42] Wikipedia. https://en.wikipedia.org/wiki/Stochastic_gradient_descent.

Shih-Hau Fang (M’07-SM’13) is a Full Professor in the Department of Electrical Engineering, and the Innovation Center for Big Data and Digital Convergence at Yuan Ze University (YZU). He received a B.S. from National Chiao Tung University in 1999, an M.S. and a Ph.D. from National Taiwan University, Taiwan, in 2001 and 2009, respectively, all in communication engineering. From 2001 to 2007, he was a software architect at Internet Services Division at Chung-Hwa Telecom Company Ltd. and joined YZU in 2009. Prof. Fang received the YZU Young Scholar Research Award in 2012 and the Project for Excellent Junior Research Investigators, Ministry of Science and Technology in 2013. His team won the third place of IEEE BigMM HTC Challenge in 2016. He is currently technical advisor to HyXen Technology Company Ltd. and serves as an Associate Editor for IEICE Trans. on Information and Systems. Prof. Fang’s research interests include indoor positioning, mobile computing, machine learning, data science and signal processing. He is a senior member of IEEE.

Yu- Xaiang Fei received the B.S. degree in Physics and Photoelctric from Fu Jen Catholic University, New Taipei City, Taiwan, in 2015. He won the third place of IEEE BigMM HTC Challenge in 2016. He received the M.S. degree in the Department of Electrical Engineering, Yuan Ze University, Taiwan, in 2017. His research interests include machine learning, pattern recognition, big data, and multivariable analysis.

Zhezhuang Xu received the Ph.D degree in Control and Systems from Shanghai Jiao Tong University, Shanghai, China in 2012. He is currently an Associate Professor with the School of Electrical Engineering and Automation, Fuzhou University, Fuzhou, China. Dr. Xus research interests include Mobile Ad-hoc Network, Wireless Sensor and Actuator Network, and their applications in Internet of Things. He has authored and/or coauthored over 30 referred international journal and conference papers. He is a Member of IEEE. He serves as a reviewer for several journals including IEEE Internet of Things Journal, IEEE Transactions on Vehicular Technology, IEEE Transactions on Industrial Electronics, and IEEE Communications Letters.

Yu Tsao received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2008. From 2009 to 2011, he was a Researcher with the National Institute of Information and Communications Technology, Kyoto, Japan, where he was engaged in research and product development in automatic speech recognition for multilingual speech-to-speech translation. He is currently an Associate Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His research interests include speech recognition, audio-coding, deep neural networks, biosignals, and acoustic modeling.