

---

# Deep Convolutional Bidirectional LSTM Based Transportation Mode Recognition

**Jeya Vikranth Jeyakumar**

University of California, Los Angeles  
vikranth94@ucla.edu

**Sandeep Singh Sandha**

University of California, Los Angeles  
sandha@cs.ucla.edu

**Eun Sun Lee**

University of California, Los Angeles  
eunsunlee@g.ucla.edu

**Nathan Tausik**

The Buckley School  
ntausik@buckley.org

**Zhengxu Xia**

University of California, Los Angeles  
zxxia@ucla.edu

**Mani Srivastava**

University of California, Los Angeles  
mbs@ucla.edu

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

*UbiComp/ISWC'18 Adjunct*, October 8-12, 2018, Singapore, Singapore.  
©2018 Copyright is held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-5966-5/18/10...\$15.00  
<https://doi.org/10.1145/3267305.3267529>

**Abstract**

Traditional machine learning approaches for recognizing modes of transportation rely heavily on hand-crafted feature extraction methods which require domain knowledge. So, we propose a hybrid deep learning model: Deep Convolutional Bidirectional-LSTM (DCBL) which combines convolutional and bidirectional LSTM layers and is trained directly on raw sensor data to predict the transportation modes. We compare our model to the traditional machine learning approaches of training Support Vector Machines and Multilayer Perceptron models on extracted features. In our experiments, DCBL performs better than the feature selection methods in terms of accuracy and simplifies the data processing pipeline. The models are trained on the Sussex-Huawei Locomotion-Transportation (SHL) dataset. The submission of our team, *Vahan*, to SHL recognition challenge uses an ensemble of DCBL models trained on raw data using the different combination of sensors and window sizes and achieved an F1-score of 0.96 on our test data.

**Author Keywords**

Transportation Modes Classification; Machine Learning; Deep Learning; Mobile Sensing

**ACM Classification Keywords**

1.2.6 [Learning]: Supervised machine learning; 1.5.1 [Models]: Neural network; 1.5.4 [Applications]: Transportation

modes

## Introduction

The rise of smartphone's sensing capabilities has enabled complex activity recognition applications like fitness tracking [24], user activity recognition [30] and transportation modes prediction [23, 29, 9, 15, 8]. Prediction of transportation modes helps to understand the user mobility patterns [29, 33] and has applications including travel time reduction, traffic planning, targeted advertisements, etc.

Multiple machine learning approaches to predict transportation modes have been tried in the past. To name a few, traditional methods like Support Vector Machines (SVM), Decision Tree, Random Forest, K-Nearest Neighbor, Naive Bayesian and Multilayer Perceptron (MLP) are used by researchers [14, 9, 34, 26, 8] on hand-crafted features extracted from the raw data. Yanyun et al. [31] use Convolutional Neural Networks (CNN) for transportation modes prediction by using statistical features as input. The accuracy reported depends on the method, complexity of the classification task and dataset used.

However, these approaches require careful feature selection and domain knowledge. Feature selection also has a trade-off between using a highly specific discriminative feature set and a more generic less discriminative feature set [4]. To avoid feature selection, modern deep learning techniques like CNN [32, 10], Deep Recurrent Neural Networks [18], ConvLSTM [21, 20] are used for activity recognition which directly takes raw data as input. Wang et al. [27] present a survey of deep learning based sensor-based activity recognition methods. Song et al. proposes DeepTransport [25] that applies deep learning to model human mobility and transportation modes using GPS (Global Positioning System) traces.

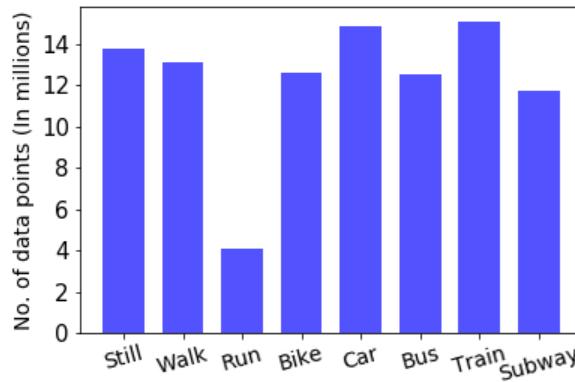
Inspired by the recent success of deep learning for activity recognition, our team, *Vahan*, proposes a deep learning model: Deep Convolutional Bidirectional LSTM (DCBL) to predict the transportation modes using raw sensor data as input. We use the SHL dataset [12] to classify the user's locomotion and transportation modes. We compare our model with the traditional machine learning approaches of SVM and MLP using the feature extracted from the data. We consider different window sizes for the data and extract multiple features both in time and frequency domain to train SVM and MLP classifiers. Our experiments show that DCBL performs better than the traditional approach using features, and it requires no feature selection or domain knowledge thereby making the pipeline for training and predictions much simpler. Finally, to make the test predictions for SHL recognition challenge, we use an ensemble of DCBL classifiers. We train seven different DCBL classifiers on raw data using the different combination of sensors and window sizes and ensemble them to get F1-score of 0.96 on our test dataset.

## SHL Dataset Description

We use SHL dataset [12] which includes sensor data recorded by a smartphone, Huawei Mate 9 [11]. A single participant collected data with the phone worn inside the front right pocket with no fixed orientation over a period of four months using an Android application called DataLogger [6]. Along with the smartphone, a body-worn camera captured photos every 30 seconds to label the data. The dataset has seven sensors: accelerometer, gyroscope, magnetometer, linear acceleration, gravity, orientation (quaternions), and ambient pressure. It consists of 271 hours of training data, 95 hours of test data from eight modes of locomotion and transportation activities: Car, Bus, Train, Subway, Walk, Run, Bike, and Still. The samples were split into 62 days for training and 20 days for testing. After the split, the data were seg-

Window Size	SVM	MLP
3s	0.783	0.776
5s	0.755	0.761
6s	0.795	0.777
8s	0.764	0.734
10s	0.783	0.749

**Table 1:** F1-scores for different window sizes using 1000 data samples. Support Vector Machine (SVM) and Multilayer Perceptron (MLP) achieve the best F1-score when the window size is 6 second. The numbers reported are the average of 5 experiments.



**Figure 1:** Distribution of labels in the SHL dataset. X-axis represents 8 different types of transportation modes. Y-axis represents the number of data points in each transportation mode.

## Procedure and Workflow

### Data Pre-processing:

The data samples (16310 samples) from the training dataset of the challenge are split into train, validation and test set in the ratio of 75:10:15. This split is used in both models on extracted features (SVM and MLP) and deep learning models on raw data (DCBL). We use the validation set to explore the different window sizes and to tune the hyper parameters of our models. We split the 1-minute window samples into smaller windows to get input sequences. In our evaluation, we try different window sizes and different overlaps across the windows while training our models. The input sequences are fed directly as input to the DCBL

mented into samples of 1-minute non-overlapping sliding window and permuted randomly. Each sample has 6000 data points. The distribution of class labels for all the data samples is shown in Figure 1.

model. To evaluate the traditional approach we extract a set of features from input sequences and use them as input to train the SVM and MLP classifiers.

### Evaluation Metric:

**F1-Score:** We use F1-score as our evaluation metric. The F1-score can be interpreted as a weighted average of precision and recall, where an F1-score reaches its best value at 1 and worst at 0. The calculation of F1-score is presented next in Equations 1 - 5.

$$\text{true positive} = tp, \quad \text{true negative} = tn \quad (1)$$

$$\text{false positive} = fp, \quad \text{false negative} = fn \quad (2)$$

$$\text{precision} = \frac{tp}{tp + fp}, \quad \text{recall} = \frac{tp}{tp + fn} \quad (3)$$

$$\text{accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (4)$$

$$F1 - score = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

### Implementation:

The entire code is developed using Python, Sklearn [22], and Keras [5] with Tensorflow backend [1]. We use the server with 60GB RAM, a single 12 GB GeForce GTX TITAN X GPU and 12 cores each clocked at 3GHz. In the next section, we discuss the details of the feature extraction and training of SVM and MLP classifiers. The details of our DCBL model are presented in the section following that.

Layers	No. of Units	Act
Den 1-2	256	relu
Den 3-4	128	relu
Den 5-12	64	relu
Den 13-20	32	relu
Den 21	8	softmax

**Table 3:** Multilayer perceptron (MLP) Model Architecture. Den 1-2 refers to the dense layer 1 and dense layer 2 both having 256 output nodes. Act refers to the activation function. relu and softmax are the activation functions. No. of Units refers to the number of output nodes.

Time Domain			Frequency Domain
Maximum	Minimum	Mean	Energy
Median	Kurtosis	Skewness	Entropy
Absolute Area	Absolute Mean	Coefficient of Variation	Mean Frequency
Interquartile Range	Standard Deviation	Root Mean Square	Band Power
Signal Magnitude	Mean Crossing Rate	Mean Absolute Deviation	Domain Frequency Ratio
Range	Percentiles	Variance	

**Table 2:** List of features extracted on the raw data. The feature set contains statistical values from time domain and frequency domain signal analysis.

## Models based on Feature extraction

### Feature Extraction:

We split each 1-minute data sample into smaller windows to generate input sequences. For each input sequence, the list of features in Table 2 is extracted from the raw data. These features are selected based on the prior work done by Bao et al. [3], Fang et al. [9] and Kwapisz et al. [16]. We compute 18 time domain features and 5 frequency domain features. The frequency domain features are based on prior work by Erdacs et al. [7]. We extract the feature values from each axis of sensors separately. Except for pressure, three axes (x, y, z) from accelerometer, gyroscope, magnetometer, linear acceleration, and gravity, and four axes (x, y, z, w) from orientation sum up to 20 sensor streams. In total, we have 720 features of all sensors within a sample.

From the previous studies of Banos et al [2], we generate input sequence using different window sizes to extract features and then normalize features with zero mean and unit

standard deviation. Before training the SVM, we apply Principal Component Analysis(PCA) to reduce the dimensionality to 50. MLP classifier uses all the 720 features as input.

### Classifiers trained using the features: SVM and MLP

We consider a small dataset of 1000 1-minute samples out of total 16310 1-minute samples to experiment with different window sizes to generate input sequences. The evaluation of classifiers on 1000 1-minute samples is used to determine the best window size and hyper parameter values. Table 1 shows F1-score of SVM and MLP trained models at different non-overlapping window sizes: 3, 5, 6, 8, and 10 seconds. SVM performs the best with 6-second window. We use GridSearchCV function from sklearn to optimize the C and gamma hyper parameters of SVM.

In addition to SVM classifiers, we also train MLP classifiers using the features extracted from 1000 samples. Similar to our approach to SVM classifiers, we train MLP classifiers with features from 3-second, 5-second, 6-second, 8-second, and 10-second non-overlapping windows. The architecture of the MLP classifier used by us is shown in Table 3. It has a total of 21 dense layers along with dropout layers. The architecture used here is generally adapted into many wide range of application by changing the number of layers and size of output nodes. We initially experimented with different architectures before picking this one. Table 3 only shows layers having trainable parameters and dropout layers are not shown.

## Model on Raw Sensor Data

There are a couple of limitations to the traditional machine learning approach. First, it requires hand-crafted feature generation as seen from the previous section. One has to first generate time-domain and frequency-domain statistics for each training window and hence there is a question

of which feature has to be generated. This is usually decided by experiments and experience and requires a lot of domain-specific expertise. Second, these traditional algorithms do not usually generalize. The accuracy of common algorithms tends to drop when we don't take into account the change mobile phone deployment location on human body (possible locations: hand, left and right pockets, etc.) or the orientation of the device once placed (the sensors axes change direction as well). Hence, we use deep learning models to make our activity predictions as these models do not require the extraction of features [21] from the sensor values and also generalize better when compared to the classical machine learning approaches [19, 17].

#### ***Building Blocks of our Model***

*Convolutional Layer (one dimensional):* It is a biologically-inspired variant of fully connected layer which is designed to use minimal amounts of preprocessing. Convolutional layers exploit spatially-local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. The main operations in Convolution layers are Convolution, Activation(ReLU), Batch normalization and Pooling or Sub Sampling.

*Long Short-Term Memory (LSTM) layer:* It is a type of Recurrent Neural Network which takes current inputs and remembers what it has perceived previously in time. A LSTM layer has a chain-like structure of repeating units and each unit is composed of a cell, an input gate, an output gate and a forget gate working together. It is well-suited to classify, process, and predict time series with time lags of unknown size and duration between important events. Because LSTMs can remember values over arbitrary intervals, they usually have an advantage over alternative RNNs, Hidden Markov models, and other sequence learning methods in numerous applications.

*Bi-Directional LSTM:* It has two LSTM layers operating in parallel. The input to the first layer is provided as-is and the input to the second layer is a reversed copy of the input sequence. This helps to preserve information from both past and the future by combining the hidden states of these two layers.

*Model Ensemble by Majority Voting:* Ensemble is the art of combining diverse set of learners (individual models) together to improve the stability and predictive power of the model or reduce the likelihood of an unfortunate selection of a poor one. The aggregate opinion of multiple models is less noisy than an individual model as ensembling reduces generalization error. The ensembled models usually have at least one of the following: difference in population, difference in modeling technique, difference in hypothesis or difference in the initial seed. In majority voting based ensemble, individual classifiers are combined by taking a simple majority vote of their decisions in the end. For any given instance, the class chosen by the most number of classifiers is the ensemble decision.

#### ***Deep Convolutional Bi-Directional LSTM (DCBL)***

Using the building blocks we designed a hybrid model:

*Deep Convolutional BiDirectional-LSTM (DCBL).* This model is a deep learning framework composed of Convolutional layers and Bidirectional LSTM recurrent layers, that is capable of automatically learning feature representations and modeling the temporal dependencies between their activation. Table 4 shows the architecture of the DCBL model. The convolutional layers act as feature extractors and provide abstract representations of the input data in feature maps. The recurrent layers model the temporal dynamics of the activation of the feature maps. Our model consists of three Convolutional layers followed by a single Bi-Directional LSTM layer. The convolution layers are followed

by batch normalization and maxpooling layers to help in faster convergence. The final layer is a dense layer with softmax activation which gives the probability distribution of the different classes. We also incorporated model ensembling, which is another powerful technique, to further increase the accuracy of our predictions. So we train seven different models with the same DCBL architecture but with differences in the training data and implement a simple majority voting based ensemble to make the final predictions. The differences in the seven DCBL models used in the ensemble are explained in Table 5.

Each network is trained for 50 epochs with a batch size of 512. We try different regularization techniques (dropout, L2-norm) and optimizers and finally decide to use 30 percent dropout and adam optimizer. The first three models, which use a window size of 10-second without overlap, yield three transport mode detections every 30 seconds. The next four models, which use a window size of 5-second with fifty percent overlap, give 23 transport mode detections every 30 seconds. These are aggregated into a per-30-second transport label and then the final labels are assigned based on majority voting.

## Results

*Models on Extracted Features:* The best F1-score for MLP classifier and SVM is for 6-second window as shown in Table 1. We choose the 6-second window to extract features from the entire SHL dataset and train MLP and SVM classifiers which achieves an average F1-score of 0.933 and 0.873 respectively on our test set as shown in Table 6.

Layers	No.ofUnits	Size	Stride	Activation
Conv1	32	(1,5)	(1,1)	relu
MaxPool		(1,2)	(1,2)	
Conv2	64	(1,3)	(1,1)	relu
MaxPool		(1,2)	(1,2)	
Conv3	128	(1,3)	(1,1)	relu
BiD-Lstm	64			
Dense	8			softmax

**Table 4:** Deep Convolutional and Bidirectional-LSTM (DCBL) Model Architecture (ignoring batch normalization and dropout layers)

*Models on Raw Sensor data:* We compare and evaluate our DCBL model with other deep learning models [13] like CNN [32], LSTM, MLP and ConvLSTM [21] which operate on raw sensor data and find that our model performs significantly better than the other existing models. Table 6 summarizes the results of using the vanilla deep learning models and DCBL on raw sensor data.

Table 5 reports the final performance achieved by an ensemble of DCBL models and also lists the performance of each individual model in the ensemble. The confusion matrix of an ensemble of 7 DCBL models is shown in Figure 2. We can see clearly from the confusion matrix that our model has no trouble in distinguishing six of the eight categories but struggles slightly when it has to distinguish between train and subway. We believe this is because train and subway have similar sensor patterns and can be mitigated if it's possible to use location information such as GPS while making the predictions.

S.No	Sensor data used in training	Window size	Overlap	F1-Score
1	Linear Acceleration, Gravity, Orientation, Pressure	10 sec	-	0.923
2	Accelerometer, Gyroscope, Magnetometer	10 sec	-	0.891
3	All Sensors	10 sec	-	0.925
4	Linear Acceleration, Gravity, Orientation, Pressure	5 sec	2.5 sec	0.925
5	Accelerometer, Gyroscope, Magnetometer	5 sec	2.5 sec	0.915
6	All Sensors	5 sec	2.5 sec	0.935
7	All (Model with Timedistributed Dense layer)	5 sec	2.5 sec	0.942
<b>Ensembled Model</b>				<b>0.964</b>

**Table 5:** This table shows the details of seven Deep Convolutional and Bidirectional-LSTM (DCBL) models which are trained using the different input of sensors and window sizes. The first 3 model windows have no overlap. The last 4 model windows have an overlap of 2.5 sec. For the ensembled model, the individual classifiers are combined by taking a simple majority vote of their decisions.

Approach	Model	Window Size	F1 Score
On Extracted Features	SVM	6 sec	0.873
	MLP	6 sec	<b>0.933</b>
On Raw Sensor Data	CNN	5 sec	0.901
	MLP	5 sec	0.702
On Raw Sensor Data	LSTM	5 sec	0.662
	ConvLSTM	5 sec	0.930
	DCBL	5 sec	0.942
	Ensembled DCBL	30 sec	<b>0.964</b>

**Table 6:** The F1-scores of individual models on the entire SHL dataset are compared in this table. The numbers reported are an average of five experiments.



**Figure 2:** The normalized confusion matrix of the final ensembled Deep Convolutional Bidirectional LSTM

## Conclusion

In this paper, we presented our approach to predict the modes of transportation using the SHL dataset. We compared the traditional approach of feature extraction to train the classifier with our modern deep learning method: Deep Convolutional Bidirectional LSTM (DCBL), of using the raw data directly as input. We show that DCBL performs better than the traditional feature extraction methods. Also using DCBL simplifies the processing pipeline and does not require tedious hand-crafted feature selection process. The recognition result for the testing dataset will be presented in the summary paper of the challenge [28].

## Acknowledgement

This research was supported in part by the NIH Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K) under award 1-U54EB020404-01, the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001, the National Science Foundation under awards # CNS-1636916 and 1640813, and the King Abdullah University of Science and Technology (KAUST) through its Sensor Innovation research program. Any findings in this material are those of the author(s) and do not reflect the views of any of the above funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

## References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, and others. 2016. Tensorflow: a system for large-scale machine learning.. In *OSDI*, Vol. 16. 265–283.
- [2] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. 2014. Window size impact in human activity recognition. *Sensors* 14, 4 (2014), 6474–6499.
- [3] Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*. Springer, 1–17.
- [4] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 33.
- [5] François Chollet and others. 2015. Keras: Deep learning library for theano and tensorflow. (2015).
- [6] Mathias Ciliberto, Francisco Javier Ordoñez Morales, Hristijan Gjoreski, Daniel Roggen, Sami Mekki, and Stefan Valentin. 2017. High reliability Android application for multidevice multimodal mobile data acquisition and annotation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 62.
- [7] Ç Berke Erdaş, İşıl Atasoy, Koray Açıci, and Hasan Oğul. 2016. Integrating features for accelerometer-based activity recognition. *Procedia Computer Science* 98 (2016), 522–527.
- [8] Shih-Hau Fang, Yu-Xaing Fei, Zhezhuang Xu, and Yu Tsao. 2017. Learning transportation modes from smartphone sensors based on deep neural network. *IEEE Sensors Journal* 17, 18 (2017), 6111–6118.

[9] Shih-Hau Fang, Hao-Hsiang Liao, Yu-Xiang Fei, Kai-Hsiang Chen, Jen-Wei Huang, Yu-Ding Lu, and Yu Tsao. 2016. Transportation modes classification using sensors on smartphones. *Sensors* 16, 8 (2016), 1324.

[10] Hristijan Gjoreski, Jani Bizjak, Martin Gjoreski, and Matjaž Gams. 2016. Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer. In *Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, New York, NY, USA*, Vol. 10.

[11] Hristijan Gjoreski, Mathias Ciliberto, Francisco Javier Ordoñez Morales, Daniel Roggen, Sami Mekki, and Stefan Valentin. 2017. A versatile annotated dataset for multimodal locomotion analytics with mobile devices. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 61.

[12] Hristijan Gjoreski, Mathias Ciliberto, Lin Wang, Francisco Javier Ordóñez Morales, Sami Mekki, Stefan Valentin, and Daniel Roggen. 2018. The University of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access* (2018).

[13] Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016).

[14] Arash Jahangiri and Hesham Rakha. 2014. Developing a support vector machine (SVM) classifier for transportation mode identification by using mobile phone sensor data. In *Transportation Research Board 93rd Annual Meeting*.

[15] Arash Jahangiri and Hesham A Rakha. 2015. Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data. *IEEE Trans. Intelligent Transportation Systems* 16, 5 (2015), 2406–2417.

[16] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* 12, 2 (2011), 74–82.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[18] Abdulmajid Murad and Jae-Young Pyun. 2017. Deep recurrent neural networks for human activity recognition. *Sensors* 17, 11 (2017), 2556.

[19] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*. 5947–5956.

[20] Tsuyoshi OKITA and Sozo INOUE. 2017. Recognition of multiple overlapping activities using compositional CNN-LSTM model. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 165–168.

[21] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

[23] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)* 6, 2 (2010), 13.

[24] Chenguang Shen, Bo-Jhang Ho, and Mani Srivastava. 2018. Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing* 17, 7 (2018), 1609–1622.

[25] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level.. In *IJCAI*, Vol. 16. 2618–2624.

[26] Leon Stenneth, Ouri Wolfson, Philip S Yu, and Bo Xu. 2011. Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 54–63.

[27] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2018a. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* (2018).

[28] Lin Wang, Hristijan Gjoreski, Kazuya Murao, Tsuyoshi Okita, and Daniel Roggen. 2018b. Summary of the Sussex-Huawei Locomotion-Transportation Recognition Challenge. *Proceedings of the 6th International Workshop on Human Activity Sensing Corpus and Applications (HASCA2018)*. (2018).

[29] Zhibin Xiao, Yang Wang, Kun Fu, and Fan Wu. 2017. Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information* 6, 2 (2017), 57.

[30] Jianbo Yang, Minh Nhut Nguyen, Phyoe Phyoe San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition.. In *IJCAI*. 3995–4001.

[31] Gong Yanyun, Zhao Fang, Chen Shaomeng, and Luo Haiyong. 2017. A convolutional neural networks based transportation mode identification algorithm. In *Indoor Positioning and Indoor Navigation (IPIN), 2017 International Conference on*. IEEE, 1–7.

[32] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 197–205.

[33] Faming Zhang, Xinyan Zhu, Wei Guo, Xinyue Ye, Tao Hu, and Liang Huang. 2016. Analyzing urban human mobility patterns through a thematic model at a finer scale. *ISPRS International Journal of Geo-Information* 5, 6 (2016), 78.

[34] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 312–321.