Deep CNN-BiLSTM Model for Transportation Mode Detection Using Smartphone Accelerometer and Magnetometer

Qinrui Tang[1,*], Kanwal Jahan[2], Michael Roth[2]

***Abstract*— Transportation mode detection from smartphone data is investigated as a relevant problem in the multi-modal transportation systems context. Neural networks are chosen as a timely and viable solution. The goal of this paper is to solve such a problem with a combination model of Convolutional Neural Network (CNN) and Bidirectional-Long short-term memory (BiLSTM) only processing accelerometer and magnetometer data. The performance in terms of accuracy and F_1 score on the Sussex-Huawei Locomotion-Transportation (SHL) challenge 2018 dataset is comparable to methods that require the processing of a wider range of sensors. The uniqueness of our work is the light architecture requiring less computational resources for training and consequently a shorter inference time.**

***Index Terms*— mode detection, self-supervised learning, deep learning, SHL dataset, BiLSTM, batch normalization**

I. INTRODUCTION

Smartphones have become ubiquitous over the last decade. In a before unseen dimension, users carry Inertial Measurement Units (IMUs) with them in every phone. The utilized MEMS accelerometers, gyroscopes, and magnetometers can provide sensor data [1] for a multitude of applications like motion monitoring, indoor positioning, and transportation mode detection.

Here addressed is the problem of transportation mode detection from smartphone sensor data, which can be viewed as a classification problem with walking, bicycling, going by car, etc. as classes (see Fig. 1). The information about the transportation mode of a user is crucial for improved decision-making and urban transport planning [2], [3]. The knowledge of a traveler's transportation mode is imperative for targeted advertisement and to automate the transportation surveys [4]. The traditional interviewing is expensive and time-consuming which can be replaced with an efficient and automated way of data collection and classification. Transportation mode detection can also support the coarse position estimation of a user once the correct mean of transportation is known [2], [3].

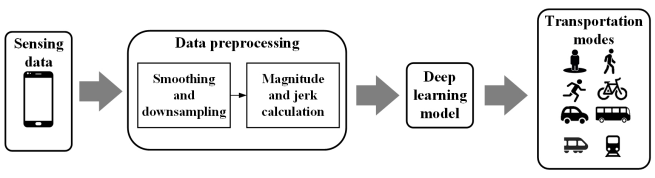

Fig. 1. Scenario of transportation mode detection problem.

[1]Institute of Transportation Systems, German Aerospace Center (DLR), Berlin, Germany. Qinrui.Tang@dlr.de

[2]Institute of Transportation Systems, German Aerospace Center (DLR), Brunswick, Germany.

Mode detection problem was solved with both traditional machine learning methods and deep learning methods. With traditional machine learning methods, feature extraction and expert knowledge are usually required. The former requests extra workloads and the required expert knowledge in the design process and the limited applicability to similar problems with e.g. slightly different sensor set-ups appear as disadvantages. By applying deep learning, many researchers can obtain high accuracy or F_1 after testing their models, but use a wide range of sensors (accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation, and ambient pressure). An increased number of sensors potentially requires more pre-processing, data storage, and also network training time.

Therefore, in this paper we aim to propose an approach with fewer input signals. Specifically, a model combining CNN and BiLSTM on only accelerometer and magnetometer measurements is suggested. Comparable accuracy and F_1 scores to approaches with more input signals are demonstrated on the Sussex-Huawei Locomotion-Transportation (SHL) dataset[1]. The main contributions of this paper are:

- using fewer sensors and computational resources compared to much previous work,
- obtaining comparable results without using feature extraction and post-process such that the workload of modeling is relatively light.

The outline of the paper is as follows. In Section II we present related work. The methods of data pre-processing and the experiment results can be seen in Section III and Section V, respectively. The details of the proposed model are described in Section IV.

II. RELATED WORK

A. Traditional machine learning methods

Traditional machine learning methods have been applied to solve the problem with good performance (e.g. high accuracy). Sauerlaender-Biebl et al. [5] use GPS and accelerometer data as inputs to predict transportation modes with Fuzzy rules and achieve relatively high accuracy in practice. If the data collected from accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation, and ambient pressure are used, XGBoost [6] and Random Forest [7] are proved as effective methods. Support Vector machines is also a classical method in this problem (e.g. [8]) and frequently regarded as a benchmark compared with deep learning methods. Usually traditional machine learning

[1]http://www.shl-dataset.org/activity-recognition-challenge/

methods require to extract features in time an frequency domain [9], which means expert knowledge and extra workload are critical.

B. Deep learning methods

Meanwhile, Deep learning methods attract attention. Deep neural networks (DNNs) with fully-connected layers for mode detection have been documented in [10], with considerable accuracy improvements over traditional machine learning methods.

With advances in deep learning, more complicated models have been constructed, but the models mainly focus on Convolutional Neural Networks (CNNs) with time-series inputs [3], [11]. A Long Short-Term Memory (LSTM) combined with CNN with good performance on two different data sets is documented in [12].

The employed sensors in the contemporary approaches differ.

A combination of accelerometer, gyroscope, and magnetometer is used in [10]. Further linear accelerations and orientations, derived from the IMU measurements, are employed in [12]. With additional gravity and ambient pressure, a total of 7 input signals are processed in [11]. Processing techniques including Hidden Markov Models (HMM) [11], [13] and Majority Voting (MV) [7], [14] are applied to improve the accuracy or F_1 scores at the price of increased computations.

III. Dataset and Data Preprocessing

A. Dataset and Sensors

The publicly available dataset, known as the University of Sussex-Huawei Locomotion (SHL) dataset [15] contains a total of 8 annotated main modes i.e. Still, Walk, Run, Bike, Car, Bus, Train, and Subway. In this paper, for the training and evaluation, we have used a subset of this dataset which was used for the SHL challenge in 2018. The data provided for this challenge comprises 271 hours of training data and 95 hours of test data. The given data is segmented into non-overlapping frames of length 1-minute, and frames are randomized to reduce the temporal dependency among them. Each frame consists of 6000 samples, i.e. 60 seconds of data recorded with a frequency of $100\,\mathrm{Hz}$. The training set consists of 16310 labelled frames in total. As each frames consists of 6000 samples, the total size of training data used for each sensor is 16310×6000 indicating the sample-wise transportation activity. The dataset contains the values of accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation (quaternions), and ambient pressure. In this paper, we use only accelerometer and magnetometer measurements for transportation mode detection, while keeping the computational complexity low.

An accelerometer records the earth's gravity and the acceleration of the device along three axes (x-, y-, and z-axis) [1]. Fig. 2 shows the acceleration values for different transportation modes measured along x-axis. The trend of mode "Still" is relatively stable but acceleration values are non-zero due to the influence of gravity. When the smartphone user moves, the value as they are combined effect of earth's gravity becomes more fluctuating and noisy.

A magnetometer is used to measure the strength and direction of the magnetic field along x, y, and z axes of the used device [1]. The magnetometer measurements are influenced by both the earth's magnetic field and the presence of magnetic materials in the surroundings, such as other electrical devices. In an indoor environment, the measurements are badly affected by such devices. Similarly, the magnetometer measurements are distorted by the electromagnetic field when smartphone users are in a car, bus, subway, or train. These modes are easy to distinguish from other transportation modes as shown in Fig. 3, the magnetometer values along the x-axis for each transportation mode. The trends of car, subway, and train differ, especially with values on the negative axis as compared to the modes like still, walk, run, and bike. A similar pattern follows along the y-axis and z-axis of the measured data.

B. Data segmentation and labelling policy

In the used dataset, a time window of duration 60 seconds is used to assign the labels to raw sensor values. The same time window is used during the testing and evaluation phase. We have investigated the effects of varying the window length on the performance of the model if any, as described in Sec. V. One-hot encoding is used for the labels while training our model. But due to the data segmentation process, a time window can have more than one mode of transportation. To address the conflict among the labels, the majority labelling policy is used as explained in [11]. When there is more than one label in a given time window, the label with the maximum time duration is assigned to the whole window. For example, in a time window of 30 seconds, if the transportation mode "Still" occupies more than 15 seconds and the other modes occupy fewer than 15 seconds, the label of this time window is assigned as "Still".

C. Smoothing and downsampling

Raw values received from an IMU are known to suffer from bias and noise [1]. Smoothing is one of the techniques to remove the random and unwanted variations from the data. We use a central moving average algorithm which is a special Savitzky-Golay filter [16] and averages an odd number of the nearest neighbors of a data item. The number of the nearest neighbors, $m = 5$, is predefined as explained in [3].

By downsampling the dataset we bring it to a more manageable size and make the learning phase computationally less expensive as well. We, as described in [12], reduce the sampling frequency by taking the mean of the data. For example, if the required frequency after downsampling is $50\,\mathrm{Hz}$, from the original $100\,\mathrm{Hz}$, every two data items are merged into one data item. We apply and compare different downsampling frequencies, details in Sec. V, to explore the best settings.

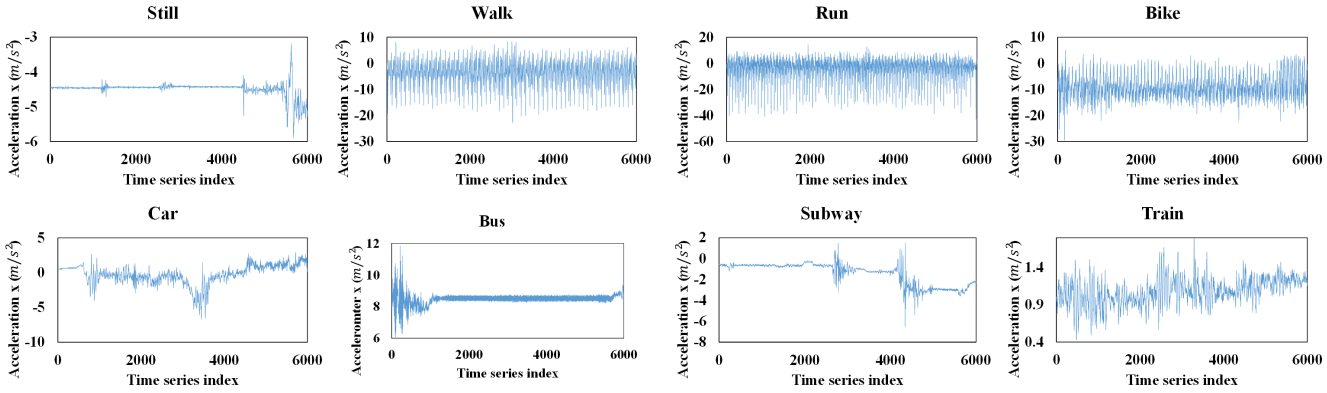

Fig. 2. Acceleration recorded along the x-axis of the measurement device for the transportation modes.

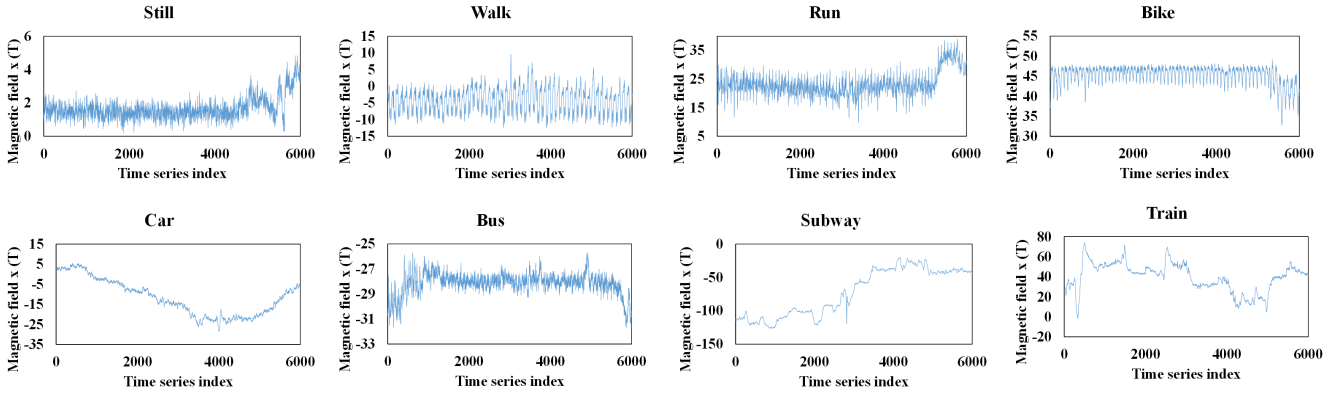

Fig. 3. Magnetic field recorded along the x-axis of the measurement device for the transportation modes.

D. Magnitude and jerk calculation

The x-, y- and z- values of an accelerometer and magnetometer measurements indicate the quantity and direction of acceleration and magnetic field respectively [1]. However, the direction of axes is based on the phone coordinate system. The values change dramatically if the orientation of the smartphone is not fixed. Therefore, applying magnitude aims to eliminate the influence of orientation changes on the transportation mode detection as mentioned by [3], [10] and [17] as well. The magnitude of each sensing data, M_s here $s \in \{Acc, Mag\}$, is calculated as shown in Eq. (1).

$$M_s(t) = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{1}$$

where d is the data point form x,y and z axis of either a magnetometer or accelerometer after applying smoothing and down sampling at a given time t.

Jerk is the rate of change of acceleration [18]. In case of sudden movements, the acceleration is non-uniform and can be estimated with the help of the jerk. It is often utilized in GPS-based mode detection [18], [19] as a significant factor in safety issues such as critical driver maneuvers and passengers' balance in public transport [18]. We aim to benefit and gain more insights by estimating the jerk and its effects on the detection of transportation modes if any, as mentioned by [7]. The same logic is applied to magnetometer measurements, to calculate the rotation rates between two-time points around an axis. $\vec{j}_s(t)$ in Eq. (2) donates the jerk of a data point $\vec{d}$ for sensor $s \in \{Acc, Mag\}$.

$$\vec{j}_s(t) = \frac{\vec{d}(t+1) - \vec{d}(t)}{\Delta t} \tag{2}$$

Here $t = 1, 2, 3, ..., n-1$ is the timestamp of data points and Δt is the time difference between two data points.

IV. CNN-BiLSTM Model

The architecture of our proposed model is shown in Fig 5. The network is inspired by Liang et al. [3] but compared to their only one source of information i.e. 1D magnitude of acceleration, our proposed model is multi-dimensional for 6 parameters. We have used the following parameters: 3D accelerometer measurements, the 1D magnitude of accelerometer measurements, the jerks for 3D accelerometer measurements, 3D magnetometer measurements, the 1D magnitude of magnetometer measurements, and the change rate for 3D magnetometer measurements. In this paper, we refer to each parameter as a channel for our network. Our proposed network has an input layer, 5 1D convolutional hidden layers handling the local dynamics of time series, a BiLSTM layer, and 2 fully connected layers. Thus the network has two major parts i.e. CNN layers and BiLSTM layer. At first, each of the 6 parameters is processed through 5 convolutional

layers separately and then concatenated together to be fed to the BiLSTM layer. The overall system architecture, where conretizing the inputs, model and outputs, is displayed in Fig. 4.

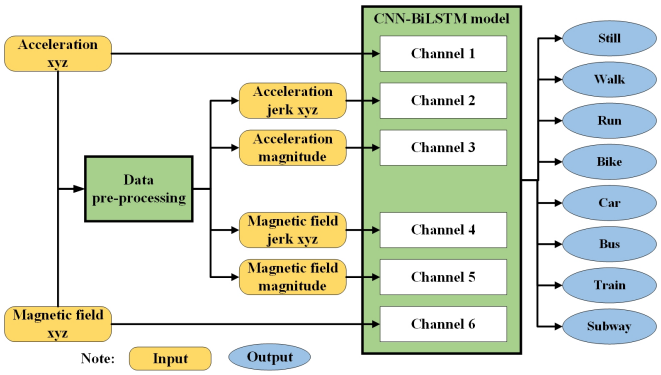

Fig. 4. System architecture.

For the first part, we gradually increase the number of filters in the convolutional layers (termed as a filter bank) and reduce the kernel size in each successive layer. The filter banks move with a stride size of 1 in each layer. The convolutional layer has 32 filters with a kernel size of 15. The second and the third convolutional layer have 64 filters and a reduced kernel size of 10. Then the fourth and the fifth convolutional layer have 128 filters with kernel size 5. After each convolutional layer, a max-pooling layer is added with pooling size 4 and stride size 2. To solve the internal covariate shift problem and make the model training phase stable, batch normalization [20] is applied before the first, the third, the fourth, and the fifth convolutional layer. Batch normalization before the second layer does not improve the performance of the network.

After processing by five convolutional layers, six channels are concatenated and fed into a BiLSTM layer with the unit size 128. A BiLSTM consists of two LSTMs: each taking the input in forward and backward directions, respectively, which is believed to extract more context information of inputs and outperforms a unidirectional LSTM.

The proposed configuration for the depth of the network, and the number of neurons in the input and subsequent layers represent the used training data well. The data finally is processed by two fully-connected layers. The first layer has a unit size of 128 and the second one has a unit size of 8 which is the same as the number of transportation modes. The output after this fully-connected layer is the probability of transportation modes. The mode with the highest probability is labeled as the predicted transportation mode. The selection of an appropriate activation function is a critical part of the design of a neural network. We have used Rectified Linear Unit (ReLU) for the hidden layers owing to its capability to overcome the vanishing gradient problem [21]. As we have a multi-class classification, for the output layer we use softmax to output the series of values, i.e. the probability of each class which sums to 1.0. The loss function of the proposed model is Mean Squared Error (MSE) to measure the average

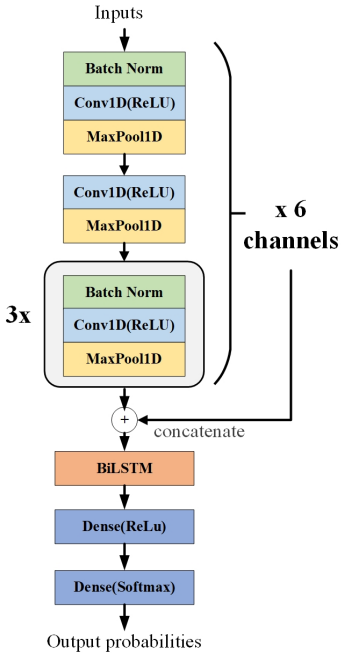

Fig. 5. Proposed CNN-BiLSTM model.

of the squares of the errors. To reduce the loss for each epoch, an Adaptive Moment Estimation (Adam) [22] is used as the optimizer. Unlike classical stochastic gradient descent which can only maintain a single learning rate, Adam can adapt the learning rate to achieve better performance. Lastly, to handle the issue of overfitting on training data, dropout and regularization are two widely accepted techniques. We however found that for our case $L2$ regularization applied in the first fully-connected layer performs better as explained in Sec. V.

V. Experiment

In our experiments, we evaluate the performance of the proposed network with varying the downsampling frequencies and the time window length on the SHL dataset [15]. We also compare the results with previous work and other machine learning-based approaches.

The model is implemented with Keras. For training the network, the originally provided train dataset is split into a sub-training set and a sub-validation set with a ratio of 9:1 by using Stratified Shuffle Split [23]. In which the label distribution of the sub-training set and the sub-validation set are the same. The test set is the same as the original SHL challenge dataset. To address the problem of overfitting, we use the early stopping strategy during the training phase while monitoring the loss and accuracy on the sub-validation set. The hyperparameters of our model, details mentioned in Sec. IV, and their selected values are listed in Tab. I.

To evaluate and compare the performance of our proposed network with state-of-the-art methods, we use precision, recall and the macro-averaged F_1 score [24] as shown in

TABLE I

PARAMETERS FOR THE CNN-BILSTM MODEL

Parameter	Value
L2 regularization	0.001
Learning rate	0.0001
Minimum learning rate	0.00001
Factor of reduced learning rate	0.2
First order moment weight in Adam	0.9
Second order moment weight in Adam	0.999
Batch size	100

Eq. (3).

$$F_1 = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \cdot recall_i \cdot precision_i}{recall_i + precision_i} \tag{3}$$

Where C is the number of transportation modes.

During the training phase, the model is initialized with random weights so the value of evaluating metrics can change in different runs of the training. To better evaluate the model, the values of metrics are estimated using the mean of 10 different training sessions. Meanwhile, we also test whether these 10 metrics follow a Normal distribution using a Shapiro-Wilk test method [25]. By doing so, the outliers in the training results can be detected. If an outlier is found, we train the model again and replace the outlier with the new evaluation result until no outliers are found.

A. Evaluation on time window and downsampling frequency

Our proposed model is tested with different lengths of the time window and downsampling frequencies. At first, we test the time window length of 60 s as originally provided in the SHL dataset [15]. Then we reduce and evaluate the window lengths of 30 s, 20 s, 10 s and 5 s, as 60 s is integer multiple of these window lengths. The used downsampling frequencies include 100 Hz (i.e. no downsampling), 50 Hz, 25 Hz, 20 Hz, 10 Hz, 5 Hz and 1 Hz.

First, it is worth to noticing the computing efficiency of this model. After we count the running time on the test set, it takes 1.57 milli second per frame with 1 GPU on NVIDIA GPU- GeForce GTX 1080 Ti.

Fig. 6 shows the prediction accuracy on the test set with the mentioned window lengths and downsampling frequencies. As the length of time window increases from 5 s to 60 s, the average accuracy of transportation mode detection increases. The highest accuracy of 91.21% is observed with a window length of 60 s and frequency of 100 Hz, i.e. when no downsampling is performed. For the remaining time windows, downsampling frequency 50 Hz shows relatively higher accuracy. It is noticed that when downsampling frequencies range from 20 Hz to 50 Hz of time window length 60 s, the accuracy values are quite similar. Then a One-Way ANOVA [26] is applied to see whether they have significant differences in statistical view and we found they do not differ from each other in the confidential interval of 95%. It is also interesting to see when downsampling frequencies jump from 1 Hz to 5 Hz, irrespective of window length, the increase in accuracy is significant.

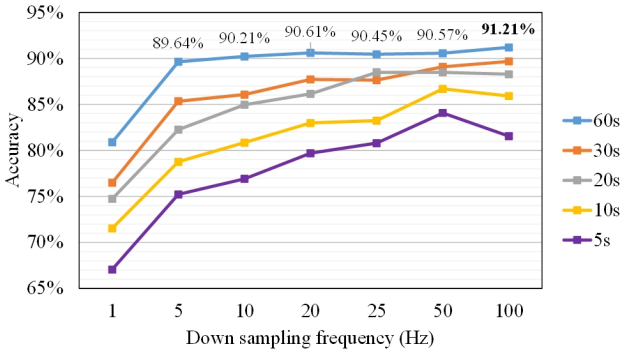

Fig. 6. Accuracy with different time windows and frequencies.

The results of F_1 scores are shown in Fig. 7. The time window 60 s gets the best F_1 scores compared to other window lengths. The downsampling frequency 20 Hz in time window 60 s gets the highest average F_1 score of 90.40%. Other higher sampling rates 25 Hz, 50 Hz and even 100 Hz have no significant difference in the F_1 score. So, the best settings to run the proposed model is with downsampling frequency 20 Hz in time window 60 s as it consumes less training resources compared to higher frequencies. The proposed network shows promising results with the smallest time window and a moderate downsampling frequency too. As depicted in Fig. 7, with a time window of 5 s and downsampling frequency of 50 Hz, the F_1 score is more than 80% which means the proposed model can extract enough features in a short time window length and predict transportation mode with good F_1 scores.

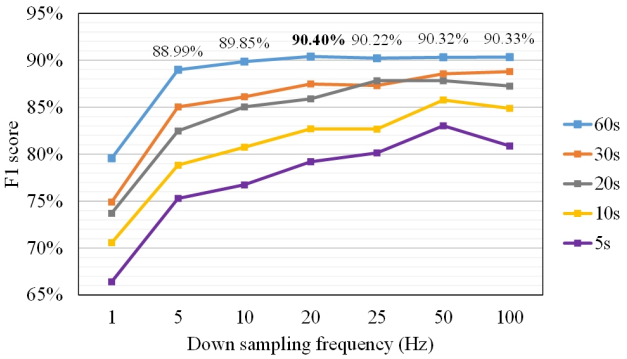

Fig. 7. F_1 score with different time windows and frequencies.

Tab. II displays the class-wise confusion matrix, precision, recall and F_1 score with downsampling frequency 20 Hz for the time window 60 s. In the confusion matrix, it can be seen that the F_1 scores of Walk, Run, Bike, and Car are more than 90%, indicating a good classification. However, the F_1 scores of Train and Subway are relatively low. This phenomenon is also observed in other research papers testing SHL dataset [15], as train and subway are very similar and prone to miss classification against each other.

TABLE II

EVALUATION WITH FREQUENCY 20 Hz IN TIME WINDOW 60 s

		Predicted labels							
		Still	Walk	Run	Bike	Car	Bus	Train	Subway
Ground Truth	Still	929	8	0	4	1	10	9	0
	Walk	9	722	0	0	0	0	0	0
	Run	0	1	336	0	0	0	0	0
	Bike	4	0	0	507	0	0	0	0
	Car	73	0	1	2	1113	84	2	1
	Bus	70	4	1	0	10	805	2	9
	Train	96	1	0	0	4	39	457	50
	Subway	11	0	0	0	0	1	31	291
	Recall(%)	96.7	98.8	99.7	99.2	87.2	89.4	70.6	87.1
	Precision(%)	78.0	98.1	99.4	98.8	98.7	85.7	91.2	82.9
	F_1 score(%)	86.3	98.4	99.6	99.0	92.6	87.5	79.6	85.0

B. Comparison with previous work

Tab. III presents the comparison results with previous works using deep learning methods. Gjoreski et al. [11] show better performance while preprocessing, 7 types of sensors, using complex feature extraction, and deep multi-modal spectro-temporal fusion. For the classification, they introduced a meta-model that combines the deep learning and machine learning-based model followed by a post-processing step of Hidden Markov Model (HMM) smoothing. By doing so, their model size is 500 MB. However, our model is lighter with minimal resources needed to preprocess, and classify the input data without the need to post-process it. Our proposed CNN-BiLSTM model, whose model size is only 24 MB, has a light architecture requiring less computational resources for training and inference. Additionally, We use data from only two types of sensors.

Tab. IV compares our approach with traditional machine learning methods. The approach used by [6] shows a higher F_1 score but at the expense of greater complexity due to using 7 types of sensors (model size 43 MB). Considering the factors like, the number of sensing data, using the subset of SHL dataset [15], and not using extensive pre-processing for feature extraction our model is lightweight and worthy of recognition in capturing key features of different transportation modes.

VI. CONCLUSION

In this paper, we focus on solving the transportation mode detection problem using a CNN-BiLSTM model. The used dataset to train and evaluate the model is SHL dataset challenge 2018. We have evaluated different configurations for the time window lengths and downsampling frequencies of given data to achieve the optimal performance of the classifier. We can detect and classify the 8 modes of transportation with high accuracy of 90.61% and an overall F_1 score of 90.40%.

Compared to many previous studies using multiple sensors as input, our proposed model only uses accelerometer and magnetometer measurements. Meanwhile, we skip feature extraction and post-process to save workloads. The experiment results show that the proposed light architecture of the model can obtain comparable accuracy and F_1 score with even a lower frequency of 20 Hz making the deployment of our model effortless. In future, we want to further investigate the relatively lower F_1 score of the classes i.e. train and subway and the rationale behind false classification of them into the transportation mode still.

REFERENCES

[1] D. Titterton and J. Weston, *Strapdown inertial navigation technology - 2nd edition*. The institution of Engineering and Technology, London, UK and The American Institute of Aeronautics, Virginia, USA, 2004.

[2] Y. Xiao, D. Low, T. Bandara, P. Pathak, H. B. Lim, D. Goyal, J. Santos, C. Cottrill, F. Pereira, C. Zegras, and M. Ben-Akiva, "Transportation activity analysis using smartphones," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012, pp. 60–61.

[3] X. Liang, Y. Zhang, G. Wang, and S. Xu, "A deep learning model for transportation mode detection based on smartphone sensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5223–5235, 2019.

[4] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and gis information," *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011.

[5] A. Sauerländer-Biebl, E. Brockfeld, D. Suske, and E. Melde, "Evaluation of a transport mode detection using fuzzy rules," *Transportation research procedia*, vol. 25, pp. 591–602, 2017.

[6] V. Janko, N. Rešçiç, M. Mlakar, V. Drobnič, M. Gams, G. Slapničar, M. Gjoreski, J. Bizjak, M. Marinko, and M. Luštrek, "A new frontier for activity recognition: the sussex-huawei locomotion challenge," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1511–1520.

[7] A. D. Antar, M. Ahmed, M. S. Ishrak, and M. A. R. Ahad, "A comparative approach to classification of locomotion and transportation modes using smartphone sensor data," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1497–1502.

[8] J. Wu, A. Akbari, R. Grimsley, and R. Jafari, "A decision level fusion and signal analysis technique for activity segmentation and recognition on smart phones," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1571–1578.

[9] 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*, vol. 18, no. 7, pp. 1662–1686, 2016.

[10] S.-H. Fang, Y.-X. Fei, Z. Xu, and Y. Tsao, "Learning transportation modes from smartphone sensors based on deep neural network," *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, 2017.

[11] M. Gjoreski, V. Janko, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Luštrek, *et al.*, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Information Fusion*, vol. 62, pp. 47–62, 2020.

[12] R. Mishra, A. Gupta, H. P. Gupta, and T. Dutta, "A sensors based deep learning model for unseen locomotion mode identification using multiple semantic matrices," *IEEE Transactions on Mobile Computing*, 2020.

[13] P. Widhalm, M. Leodolter, and N. Brändle, "Top in the lab, flop in the field? evaluation of a sensor-based travel activity classifier with the shl dataset," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1479–1487.

[14] A. Akbari, J. Wu, R. Grimsley, and R. Jafari, "Hierarchical signal segmentation and classification for accurate activity recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1596–1605.

TABLE III

COMPARISON WITH PREVIOUS DEEP LEARNING MODELS

Research	Sensing data	Input	Classifier	Post process	F_1 score
Gjoreski et al. [11]	Accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation, ambient pressure.	Spectrogram + features	ML+DL	HMM[1]	94.9%
Proposed model	**Accelerometer, magnetometer.**	**Raw data**	**CNN+BiLSTM**	**/**	**90.4%**
Ito et al. [27]	Accelerometer, gyroscope.	Spectrogram	CNN	/	88.8%
Akbari et al. [14]	7 sensors as [11]	Features	DNN	MV[2]	86.3%

1 HMM: Hidden Markov model
2 MV: Majority Voting

TABLE IV

COMPARISON WITH PREVIOUS MACHINE LEARNING MODELS

Research	Sensing data	Input	Classifier	Post process	F_1 score
Janko et al. [6]	7 sensors as [11]	Features	XGBoost		92.4%
Proposed model	**Accelerometer, magnetometer,**	**Raw data**	**CNN+BiLSTM**	**/**	**90.4%**
Widhalm et al. [13]	Accelerometer, Gyroscope, magnetometer, ambient pressure,	Features	Multilayer perceptron	HMM	87.5%
Antar et al. [7]	7 sensor as [11]	Features	Random forest	MV	87.5%

[15] H. Gjoreski, M. Ciliberto, L. Wang, F. J. Ordonez Morales, S. Mekki, S. Valentin, and D. Roggen, "The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices," *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[16] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[17] J. Iskanderov and M. A. Guvensan, "Breaking the limits of transportation mode detection: Applying deep learning approach with knowledge-based features," *IEEE Sensors Journal*, vol. 20, no. 21, pp. 12 871–12 884, 2020.

[18] S. Dabiri and K. Heaslip, "Inferring transportation modes from gps trajectories using a convolutional neural network," *Transportation research part C: emerging technologies*, vol. 86, pp. 360–371, 2018.

[19] L. Li, J. Zhu, H. Zhang, H. Tan, B. Du, and B. Ran, "Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using gps data," *Transportation Research Part A: Policy and Practice*, vol. 136, pp. 282–292, 2020.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37. PMLR, 2015, pp. 448–456.

[21] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR*, 2015.

[23] K. R. W. Brewer, "Design-based or prediction-based inference? stratified random vs stratified balanced sampling," *International Statistical Review*, vol. 67, pp. 35–47, 1999.

[24] H. Dalianis, *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer International Publishing, 2018, ch. Evaluation Metrics and Evaluation, pp. 45–53.

[25] S. Shapiro and M. Wilk, *An analysis of variance test for normality*. Oxford University Press and Biometrika Trust, 1965, vol. 52, no. 3, pp. 591–611.

[26] R. M. Heiberger and E. Neuwirth, *R Through Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics*. Springer New York, 2009, ch. One-Way ANOVA, pp. 165–191.

[27] C. Ito, X. Cao, M. Shuzo, and E. Maeda, "Application of cnn for human activity recognition with fft spectrogram of acceleration and gyro sensors," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1503–1510.