Transportation Research Interdisciplinary Perspectives 26 (2024) 101181

Contents lists available at ScienceDirect

Transportation Research Interdisciplinary Perspectives

journal homepage: www.elsevier.com/locate/trip

Feature pyramid biLSTM: Using smartphone sensors for transportation mode detection

Qinrui Tang [a,b], Hao Cheng [c,*]

[a] Zhejiang Wision Digital Technology Information Industry Co., Ltd, Zhejiang, China
[b] German Aerospace Center (DLR), Institute of Transportation Systems, Berlin, Germany
[c] ITC Faculty Geo-Information Science and Earth Observation, University of Twente, Enschede, The Netherlands

ARTICLE INFO

Keywords:
Deep learning
Transportation mode detection
Activity recognition
Mobile sensors

ABSTRACT

The wide utilization of smartphones has provided extensive availability to Inertial Measurement Units, providing a wide range of sensory data that can be advantageous for transportation mode detection. This study proposes a novel end-to-end approach to effectively explore a reduced amount of sensory data collected from a smartphone, aiming to achieve accurate mode detection in common daily traveling activities. Our approach, called Feature Pyramid biLSTM (FPbiLSTM), is characterized by its ability to reduce the number of sensors required and processing demands, resulting in a more efficient modeling process without sacrificing the quality of the outcomes than the other current models. FPbiLSTM extends an existing CNN biLSTM model with the Feature Pyramid Network, leveraging the advantages of both shallow layer richness and deeper layer feature resilience for capturing temporal moving patterns in various transportation modes. It exhibits an excellent performance by employing the data collected from only three out of seven sensors, *i.e.,* accelerometers, gyroscopes, and magnetometers, in the 2018 Sussex-Huawei Locomotion (SHL) challenge dataset, attaining a noteworthy accuracy of 95% and an F_1-score of 94% in detecting eight different transportation modes.

1. Introduction

In the last decade, smartphones have gained widespread prevalence in daily existence, hence facilitating an unparalleled degree of accessibility to Inertial Measurement Units (IMUs) that are included within these devices. The utilization of IMUs, which comprise components such as Microelectromechanical Systems (MEMS) accelerometers, gyroscopes, and magnetometers, has the potential to provide a wide range of sensory data that can be advantageous for various applications. These applications include motion tracking, indoor positioning, and transportation mode detection.

This study focuses on the issue of detecting different transportation modes using smartphone sensory data. It can be viewed as a classification problem encompassing several categories such as walking, biking, and driving a car. Obtaining accurate information about the user's chosen mode of transportation is crucial for enhancing decision-making procedures and formulating effective strategies for urban transportation planning (Xiao et al., 2012; Liang et al., 2019). Additionally, comprehending the transportation mode preferences of passengers is crucial for providing tailored advertising strategies and optimizing the efficiency of transportation surveys. With its high costs and time requirements, the traditional interview-based approach can potentially be replaced by a more efficient and automated system for collecting and categorizing data. Smartphones with various IMUs are commonly carried by users while traveling, which are the perfect devices for transportation mode detection. Moreover, the detection of transportation modes can assist in approximating a user's overall location once the appropriate mode of transportation has been determined.

The problem of mode detection has been tackled via both traditional machine learning and deep learning methods. In the context of traditional machine learning, there is a frequent requirement for feature extraction and domain expertise. The inclusion of these conditions may result in an increase in workloads and could potentially restrict the adaptation of the technique to similar issues with modest differences in sensor configurations, thereby presenting a drawback in their applications. In contrast, deep learning has facilitated a multitude of researchers in attaining elevated levels of accuracy or F_1-scores throughout the evaluation of their models. Nevertheless, these models commonly employ a wide range of sensors including accelerometers, gyroscopes, magnetometers, linear accelerometers, gravity sensors, orientation sensors, and ambient pressure sensors. The presence of a larger quantity of sensors can suddenly result in an increased demand for greater capacity for data storage, additional resources for data

[*] Corresponding author.
E-mail address: h.cheng-2@utwente.nl (H. Cheng).

https://doi.org/10.1016/j.trip.2024.101181
Received 27 October 2023; Received in revised form 15 April 2024; Accepted 30 July 2024
Available online 6 August 2024

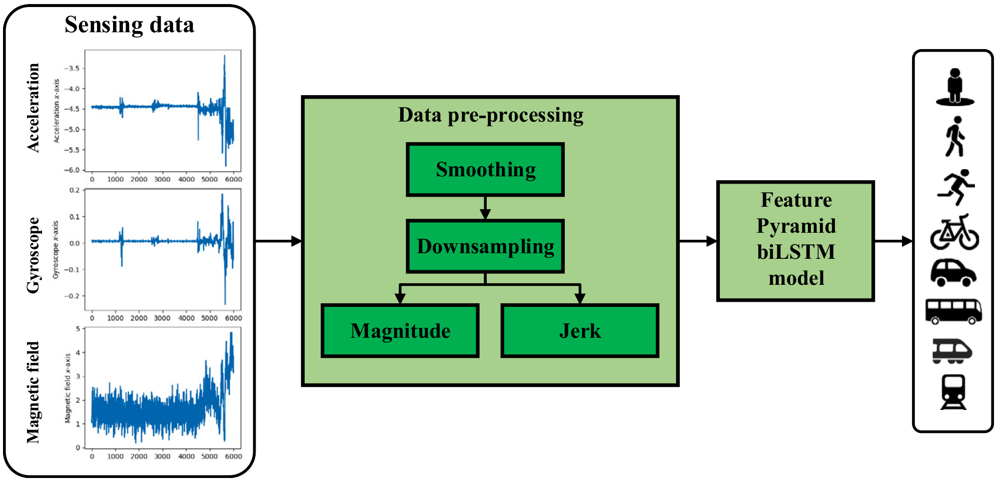

Fig. 1. Pipeline of transportation mode detection problem.

preprocessing, and perhaps heightened requirements for model training resources.

To comply with computational limits, utilizing a reduced number of input signals for transportation mode detection is more desirable for an economical deployment. The early works, *e.g.,* Ito et al. (2018) applied a CNN-based model that takes only the spectrogram from both accelerometer and magnetometer measurements, and Widhalm et al. (2018) employed a multilayer perception-based model with an additional magnetometer measurement. However, their performances are noticeably inferior to the other models that leverage the measurements from more sensors. Moreover, Tang et al. (2022) attempted to propose a more advanced network framework to compensate for the limited measurements. The approach involves the integration of Convolutional Neural Network (CNN) and bidirectional Long Short-Term Memory (biLSTM) models, and utilizes accelerometer and magnetometer measurements for the detection. It demonstrates on-par levels of accuracy and F_1-scores as the other methods that utilize a greater number of input signals. In this paper, we follow the constraint of using a reduced number of sensory measurements and seek a more effective end-to-end approach for transportation mode detection. Namely, this paper introduces a novel approach called Feature Pyramid biLSTM (FPbiLSTM), which is inspired by the Feature Pyramid Network (FPN) proposed by Lin et al. (2017) (see Fig. 1). The FPN leverages the abundant information present in lower levels and the robust features extracted from higher layers, resulting in enhanced model performance. Transportation modes can differ significantly in travel speeds, such as walking and driving a car, while they can also present very similar motion patterns, such as riding in a subway or on a train. Therefore, this pyramid architecture is exploited to learn temporal patterns from measurements at various time granularities, enabling the distinction of different transportation modes from both heterogeneous and homogeneous motion patterns. This assertion is substantiated in the experimental phase of our study by improving the F_1-score from 90.4% to 94.2% when compared to that of Tang et al. (2022) under the same setting. Moreover, our model is further enhanced by incorporating a feature selection technique that includes additional gyroscope measurements. Despite the increased complexity of the network structure and the addition of an extra sensor compared to Tang et al. (2022), FPbiLSTM remains a very efficient and lightweight end-to-end model.

The *main contributions* of this study are as follows.

- This paper proposes a novel Feature Pyramid biLSTM model that encompasses reduced sensors and computational resources compared to earlier research efforts for transportation mode detection.
- The proposed model is end-to-end as it takes as input the raw data and requires no extra feature extraction, post-processing, or ensembling techniques to boost the detection performance.
- It exhibits an excellent performance by employing the data collected from only accelerometers, gyroscopes, and magnetometers, in the 2018 Sussex-Huawei Locomotion (SHL) challenge dataset, achieving an accuracy of 95.1% and an F_1-score of 94.2% in detecting eight different transportation modes.

In the remainder of this paper, Section 2 reviews the related works closely aligned with our proposed model, followed by a detailed description of the method in Section 3. The experiments and discussions are presented in Section 4. Finally, Section 5 concludes the paper with further insights into future work.

2. Related work

2.1. Sensors used for transportation mode detection

Given the ubiquity of smartphones, many researchers are exploring the potential of using smartphone sensory data to determine traffic modes. This data can be broadly split into motion-based and location-based (Yu et al., 2014) sensory data. Motion-based data arises from devices like accelerometers, gyroscopes, magnetometers, linear accelerometers, gravity sensors, orientation (quaternions), and ambient pressure, while location-based data typically stems from Global Positioning System (GPS) or Global Navigation Satellite System (GNSS).

In motion-based applications, it is commonly observed that utilizing a greater number of sensors tends to yield improved outcomes in terms of classification performance. However, the types of sensors used in cutting-edge approaches vary. For example, Gjoreski et al. (2020), Janko et al. (2018, 2019), Choi and Lee (2019), Zhu et al. (2020) and Kalabakov et al. (2020) conducted studies utilizing data collected from seven distinct sensors, specifically, the accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation (quaternions), and ambient pressure. These studies yielded highly favorable classification outcomes, with Gjoreski et al. (2020) achieving an impressive F_1-score of 94.9%. To reduce the storage and computational costs, some works prioritize the advancement of streamlined models that incorporate a reduced number of sensors with some sacrifice of classification performance. The study by Qin et al. (2019) employed four sensors in total: the linear accelerometer, gyroscope, magnetometer, and pressure sensor. Fang et al. (2017) further reduced the number of sensors to three, specifically, accelerometer, magnetometer, and gyroscope readings. Furthermore, Ito et al. (2018) employed a combination of only accelerometer and gyroscope sensors in their study, and similarly, Tang et al. (2022) utilized two sensors, namely accelerometer and magnetometer. In the single sensor case, Liang et al. (2019) exclusively utilized data obtained from the accelerometer. Generally, the three most commonly utilized sensors are the accelerometer, magnetometer, and gyroscope. In this paper, the introduced approach focuses on these three commonly used sensors and attempts to achieve the same level of performance as that of using more sensors.

Models that utilize location-based data have also demonstrated successful results. The sensors for acquiring location-based data include GPS position, reception, WiFi, and cellular technology. The studies conducted by Saha et al. (2021) and Balabka and Shkliarenko (2021) have produced favorable results when utilizing these sensory data. GNSS is another option to be used (Munoz Diaz et al., 2023). Nevertheless, it is important to note that the overall efficacy of these sensors for transportation mode detection is typically lower compared to the data obtained using motion-based methodologies. It should be noted that this paper only focuses on utilizing motion-based data for transportation mode detection, which is much less privacy-concerning than location-based data that may reveal the origin and destination travel patterns of smartphone owners.

2.2. *Traditional machine learning methods*

Traditional machine learning methods have exhibited significant efficacy in tackling the problem of transportation mode detection, frequently resulting in high accuracy. For example, efficient algorithms, *e.g.,* XGBoost (Janko et al., 2018; Kalabakov et al., 2020; Zhu et al., 2020) and Random Forest (Antar et al., 2018; Janko et al., 2019; Saha et al., 2021; Zhu et al., 2021), have demonstrated their effectiveness in analyzing data obtained from accelerometers, gyroscopes, magnetometers, linear accelerometers, gravity, orientation, and ambient pressure sensors. Support Vector Machines (SVMs) are frequently utilized in this particular domain (Wu et al., 2018) and are frequently regarded as a standard against which deep learning methods are compared. Despite the performance superiority, these traditional machine learning methods often require extracting features in both the time and frequency domains, and they normally require a large amount of sensory data to extract those features.

Moreover, rather than utilizing raw sensory data, traditional machine learning techniques are often applied for feature transformation. The Fast Fourier Transformation is frequently used for gleaning frequency domain information from sensory data (Janko et al., 2018, 2019; Kalabakov et al., 2020), especially the time domain features. For example, Janko et al. (2018) highlighted the role of expert knowledge in selecting frequency domain features. Specifically, they chose to examine three primary magnitudes: Energy, Entropy, and Binned distribution, along with Skewness and Kurtosis. Widhalm et al. (2018) also incorporated autocorrelation as a key feature in their research. The process of feature transformation requires not only expert knowledge but also additional efforts from domain experts.

In addition, the sequential signal measurements are summarized in statistics before feeding into a detection model (Janko et al., 2018, 2019; Kalabakov et al., 2020; Zhu et al., 2021; Ren, 2021). These statistical features include minimum, maximum, mean, variance, standard deviation, mean absolute deviation, autocorrelation, count of samples above or below the mean, and the average variance between consecutive data points. Whereas, these high-level statistics inevitably lose the detailed signal changes in consecutive measurements with a high frequency, *e.g.,* 100 Hz, which can be very helpful to capture motion patterns in different transportation modes.

2.3. *Deep learning methods*

In recent years, deep learning methods have garnered significant attention with notable improvements in accuracy for transportation mode detection. The commonly used network architectures are, *e.g.,* Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) model, and Gated Recurrent Unit (GRU). For example, Fang et al. (2017) and Liang et al. (2019) employed CNNs with time-series inputs for transportation mode detection. Thanks to the gating mechanisms designed for propagating information along the time axis, Mishra et al. (2020) combined LSTM with a CNN, achieving impressive mode detection performance on two distinct datasets (Mishra et al., 2020). Similarly, GRU is also proven to be very efficient in mode detection problem (Zhu et al., 2020). Like LSTM and GRU, 1D convolutional layers have been demonstrated in handling time-series of mode detection, as demonstrated by 1D DenseNet proposed in Zhu et al. (2019).

Another advantage of deep learning models is that feature extraction is not always mandatory, compared to traditional machine learning methods. For example, although purportedly executing feature extraction, Choi and Lee (2019) employs convolutional layers on raw data and feeds them into an EmbraceNet. Deep learning models commonly apply a data loader module to prepare raw data directly for input. This module includes functions to compute the magnitude of variables like acceleration, gyroscope, and magnetic field (Zhu et al., 2019), jerk of the sensors (Tang et al., 2022), and adapting the mobile phone's coordinate system to a global reference frame, among others (Gjoreski et al., 2020). In this way, signal expert knowledge is not necessarily required to preprocess the raw data, which makes it considerably distinct from the feature extraction concept discussed earlier. However, these models mentioned above have not fully explored a strategy to reduce the amount of sensory data and build a more efficient and effective model for transportation mode detection. Therefore, this paper introduces a new data loader to prepare the input from raw sensory data and trains a more lightweight end-to-end deep learning model for transportation mode detection.

3. Mode detection using locomotion data

3.1. *Problem formulation*

The locomotion data collected from motion-based sensors is leveraged for transportation mode detection. Mathematically, the detection task is defined as given a frame of sensor measurements, denoted as X_i, in a certain reading rate, *e.g.,* 100 Hz, the detection function $f(X)$ is to recover the transportation mode Y. Each frame of a certain length contains a set of sensor measurements, such as an accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation, and ambient pressure. Namely, $X_i = \{x^t_{i,1}, x^t_{i,2}, \dots, x^t_{i,j}, \dots\}^T_{t=1}$, where j stands for one of the aforementioned sensor types and t is the time step of a total length of T time steps. The objective function is to minimize the detection errors $\sum_{i=1}^{N} L(\hat{Y}_i, Y_i)$ between the prediction mode $\hat{Y}_i = f(X_i)$ and ground truth mode Y_i across all the N frame samples.

3.2. *Dataset and data loader*

Concretely, the University of Sussex-Huawei Locomotion (SHL) dataset (Gjoreski et al., 2018) is used for transportation mode detection. It comprises eight primary modes, *i.e., Still*, *Walk*, *Run*, *Bike*, *Car*, *Bus*, *Train*, and *Subway*. This dataset was collected by three users with their smartphones at four distinct positions, including the hand, hips, torso, and bag, in a predefined weekly outline of activities. This paper follows the guidance of using the data complied with privacy and ethics requirements. It consists of the recorded measurements of various sensors, including the accelerometer, gyroscope, magnetometer, linear accelerometer, gravity, orientation (expressed as quaternions), and ambient pressure. This study focuses on utilizing accelerometer, magnetometer, and gyroscope measurements for transportation mode detection, aiming to maintain a low level of computing complexity. This SHL dataset contains 753 h of locomotion data, which has been appropriately labeled. The duration distributions of various modes are as follows: Car (88 h), Bus (107 h), Train (115 h), Subway (89 h), Walk (127 h), Run (21 h), Bike (79 h), and Still (127 h). The provided data is divided into distinct frames of one-minute duration, with the frames being randomly arranged to minimize temporal interdependence, *i.e.,* every frame comprises 6000 samples, representing a duration of 60 s, collected at a sampling rate of 100 Hz.

A subset of the SHL dataset known as *SHL challenge 2018* was utilized for our detection task. Specifically, 271 and 95 h were designated for training and testing, respectively. The training set has 16,310 frames, resulting in a total training data size of $16\,310 \times 6000$ for each sensor at the sample sampling rate. Similarly, the test set has a size of 5698×6000 for each sensor. It should be noted that we utilize one-hot encoding for the frame-level labeling during the training process. However, it is possible for a time window, *e.g.,* 60 s, to encompass multiple transportation modes due to the data segmentation process. Hence, the majority labeling policy is implemented, as described in Ref. Gjoreski et al. (2020), to resolve the controversy arising from the labels. Moreover, the impact of altering the window length on the model's performance was examined following the details provided in Section 4.3.

Before introducing the proposed detection model, we introduce the data loader to facilitate end-to-end training and detection. Specifically, the data loader provides the functions to automatically process the data for the model, including smoothing, downsampling, magnitude calculation, and jerk calculation.

Smoothing. The received results from an IMU are acknowledged to be susceptible to bias and noise (Titterton and Weston, 2004). Smoothing is a commonly employed method to mitigate random and undesired fluctuations within a dataset. The algorithm employed in this study is a central moving average technique, specifically a Savitzky–Golay filter (Savitzky and Golay, 1964). This filter calculates the average of an odd number of neighboring data points surrounding a given data item. The value of m, representing the number of nearest neighbors, is predetermined and discussed in Liang et al. (2019). The averaged smoothing value is calculated in (1).

$$\bar{d}_t = \begin{cases} \frac{\sum_{i=1}^{2t-1} d_i}{2t-1} & t \in \left[1, \lfloor \frac{m}{2} \rfloor\right], \\ \frac{\sum_{i=t-(m/2)}^{t+(m/2)} d_i}{m} & t \in \left(\lfloor \frac{m}{2} \rfloor, T - \lfloor \frac{m}{2} \rfloor\right), \\ \frac{\sum_{i=2t-T}^{T} d_i}{2(T-t)+1} & t \in \left[T - \lfloor \frac{m}{2} \rfloor, T\right], \end{cases} \tag{1}$$

where t represents the tth sample of a frame, and T is the total number of samples in a frame, and d_t represents the x, y, and z axis of accelerometer, gyroscope or magnetometer measurements, and $\bar{d}_t$ is the smoothed data.

Downsampling. Downsampling the dataset reduces its size, making it more manageable and decreasing the computing cost of the learning process. Mishra et al. (2020) applied downsampling to decrease the sample frequency by calculating the average of the data. In this paper, a similar strategy for the downsampling is applied. The pth data point after downsampling, $\hat{d}_p$, is obtained with (2).

$$\hat{d}_p = \frac{\sum_{t}^{t+S-1} \bar{d}_i}{S}, \tag{2}$$

where $t = nS + 1, n = \{0, 1, 2, \ldots, \lfloor F/S \rfloor\}$, and S is the downsampling size and F represents the original sampling frequency. For example, in the SHL dataset, if sampling frequency F is 100 Hz and $S = 2$, every two data points are averaged so the frequency after downsampling becomes 50 Hz. It should be noted that to determine the optimal settings, we employ various downsampling frequencies and conduct a comparative analysis. Further information regarding this experiment can be found in Section 4.

Magnitude calculation. Magnitude is used to mitigate the influence of orientation change effects. The direction of axes (*i.e.,* x, y, and z) in sensor measurements is determined by the phone coordinate system, whose values alter drastically if the smartphone's orientation is not fixed. Consequently, applying magnitude seeks to eliminate the influence of orientation changes on the detection of the transport mode, as stated in Liang et al. (2019), Fang et al. (2017) and Iskanderov and Guvensan (2020). The magnitude of each sensing data at data point p, M_p is calculated as shown in (3).

$$M_p = \sqrt{\hat{d}_{x,p}^2 + \hat{d}_{y,p}^2 + \hat{d}_{z,p}^2}, \tag{3}$$

where $\hat{d}_{x,p}$, $\hat{d}_{y,p}$ and $\hat{d}_{z,p}$ represent the data point from the x, y, and z axes of one sensor from the accelerometer or gyroscope after downsampling, respectively.

Jerk calculation. Jerk originally represents the rate at which acceleration changes (Dabiri and Heaslip, 2018). During abrupt movements, acceleration is not uniform, and its estimation can be aided by analyzing the jerk. This concept has been frequently used in GPS-based mode detection (Iskanderov and Guvensan, 2020; Dabiri and Heaslip, 2018), primarily due to its implications in safety scenarios like crucial driving actions and maintaining passenger stability in public transit (Dabiri and Heaslip, 2018). We aim to discern the advantages and understand the potential impact of estimating the jerk on transportation mode detection, as alluded to in Antar et al. (2018). The same principle is used for gyroscope and magnetometer readings, helping calculate rotation rates between two distinct data points around a given axis. Specifically, $\vec{J}_p$ signifies the jerk of specific data points $\vec{\hat{d}}_{p+1}$ and $\vec{\hat{d}}_p$ for sensors falling in accelerometer, gyroscope or magnetometer. The equation to obtain jerk is shown in (4).

$$\vec{J}_p = \frac{\vec{\hat{d}}_{p+1} - \vec{\hat{d}}_p}{\Delta t}, \tag{4}$$

where p denotes the data point index and Δt is the time difference between successive data point $p + 1$ and p.

3.3. Feature pyramid biLSTM model

As illustrated in Fig. 2, our proposed model is developed based on the CNN biLSTM model (Tang et al., 2022) and FPN (Lin et al., 2017). The CNN biLSTM model evolves from Liang et al. (2019), which drew inspiration from AlexNet (Krizhevsky et al., 2012) and utilizes only linear acceleration as an input for transport mode detection. On the contrary, the proposed model incorporates the outputs of various convolutional layers into the biLSTM and fully connected layers to facilitate prediction. This approach enables the utilization of abundant information from the shallow layer and the extracted features from the deeper layer, to further enhance the model's capabilities. We refer to our model as the Feature Pyramid biLSTM model (FPbiLSTM).

FPbiLSTM takes as input the measurements from the accelerometer, magnetometer, and gyroscope using the data loader defined in Section 3.2. Namely, they are jerk measurements from the x-, y-, and z-axis of acceleration, overall acceleration magnitude, jerk measurements from the x-, y-, and z-axis of the magnetic field, smoothed and downsampled readings from the x-, y-, and z-axis of the gyroscope, and the gyroscope's overall magnitude. The original *SHL challenge 2018* dataset consists of frames in a duration of 60 s with a sampling frequency of 100 Hz. A downsampling frequency of 20 Hz is used in the proposed model. In the following, we explain each step of the model in detail.

As depicted in Fig. 3, initially, the FPbiLSTM model processes each input through five separate channels to represent each input in a higher dimension individually. Each channel's architecture and hyperparameters share the same features. For every channel, the filter count in the convolutional layers, commonly known as a filter bank, gradually increases, while the kernel size diminishes in the following layers. Each layer's filter banks have a stride size of 1. Due to the

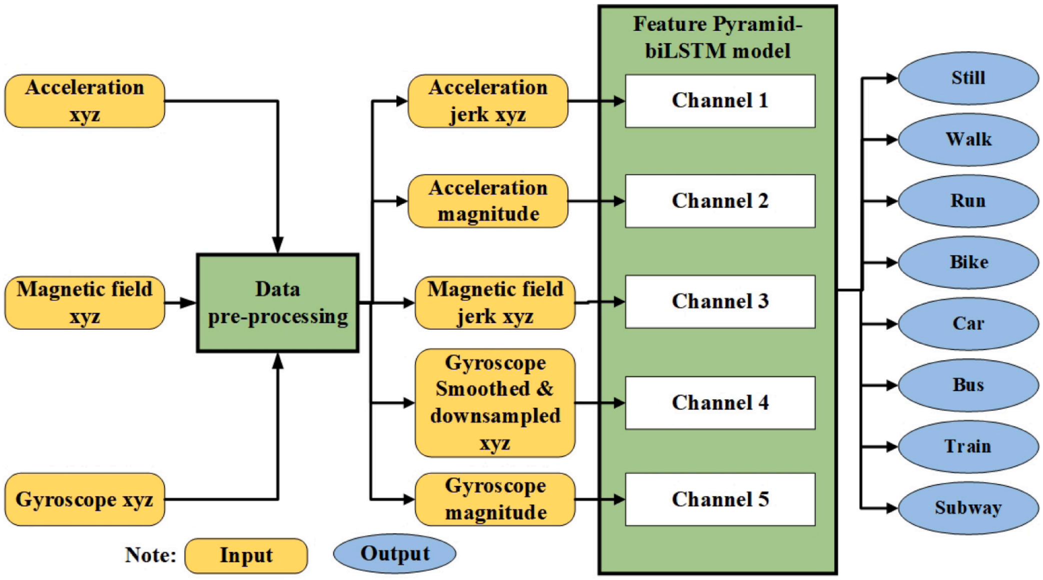

Fig. 2. System architecture of feature pyramid biLSTM.

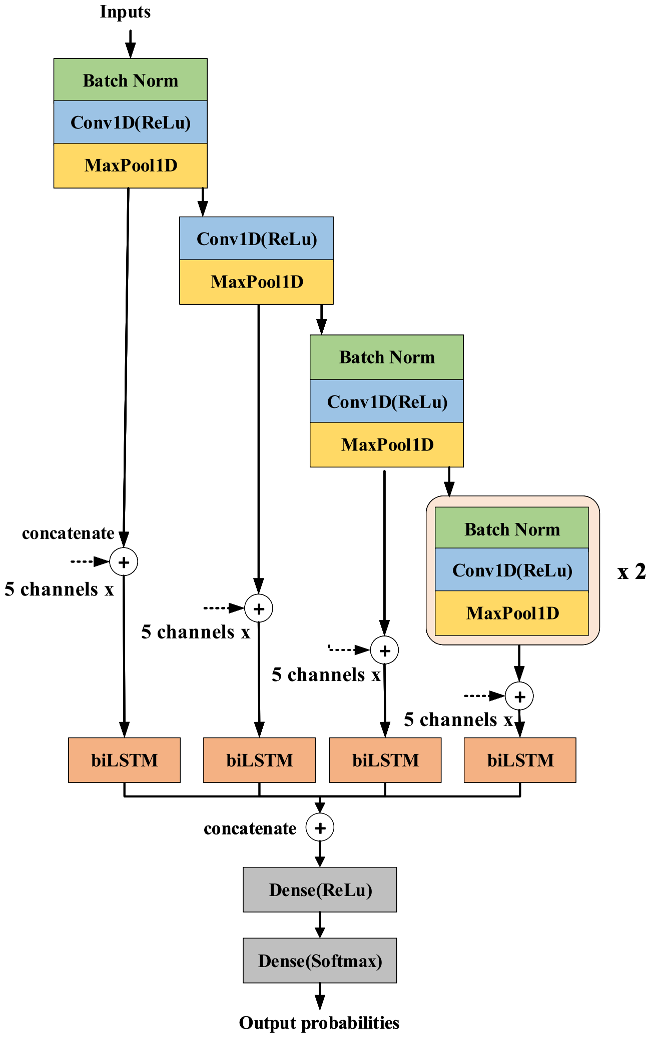

Fig. 3. Architecture of feature pyramid biLSTM.

high frequency of sensor measurements, a relatively large kernel size is used to process the input. Specifically, the inaugural convolutional layer consists of 32 filters with a kernel size of 15 each. The subsequent two layers feature 64 filters and a reduced kernel size of 10. The fourth and fifth convolutional layers incorporate 128 filters, each with a kernel size of 5. After every convolutional layer, a max-pooling layer is added, boasting a pooling size of 4 and a stride size of 2. To tackle the internal covariate shift and maintain stability during the model's training, batch normalization (Ito et al., 2018) is applied before the first, third, fourth, and fifth convolutional layers. However, empirically, we found that introducing batch normalization before the second layer does not notably improve the network's efficacy. Hence, it is not applied to the second layer. At the end of the last convolutional layer, the channel-wise feature maps are concatenated along the channel axis to fuse them.

Consequently, we employ a biLSTM to learn the temporal information from the feature maps outputted by the above convolution layers. To facilitate the learning of temporal information at different levels, we use skip connections to feed the feature maps to the biLSTM. Concretely, the outputs of the first, second, third, and fifth max-pooling layers are fed into four biLSTM layers with a unit size of 128. This biLSTM model is composed of two LSTMs. They process the input in both the forward and backward directions, aiming to capture additional contextual information and achieve superior performance compared to a unidirectional LSTM. A more detailed ablation study on the skip connections is provided in Section 4.5.

In the end, the learned information is fed to two stacked dense layers for computing the classification scores. The first layer has 128 units and the following layer has 8 units, matching the overall count of transportation modes. The outputs after the second dense layer followed by the Softmax activation represent the probability scores across different transportation modes. Finally, the anticipated transportation mode is indexed with the highest score.

The selection of a suitable activation function is pivotal in the design of a neural network's architecture. The Rectified Linear Unit (ReLU) activation function was chosen for the hidden layers due to its effectiveness in mitigating the vanishing gradients issue (Xu et al., 2015). For multi-class classification tasks, the Softmax function is used in the output layer to produce a set of values denoting the probability of each category.

Loss function. The proposed model utilizes the Mean Squared Error (MSE) as the loss function, measuring the mean of squared differences.

Table 1
Hyperparameters for the feature pyramid biLSTM model.

Hyperparameters	Value
L2 regularization	0.001
Learning rate	0.0001
Minimum learning rate	0.00001
The 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	50

To reduce the loss with every epoch, the Adaptive Moment Estimation (Adam) optimizer is applied (Kingma and Ba, 2015). Lastly, the $L2$ regularization is introduced to the first dense layer to counteract overfitting on training data.

4. Experiment

4.1. Experiment settings and evaluation metrics

The initial training dataset is partitioned into two subsets to facilitate the training process: a sub-training set and a sub-validation set. This partitioning is achieved by employing the Stratified Shuffle Split technique (Brewer, 1999), with a ratio of 90:10, thus the label distribution of the sub-training set and the sub-validation set is the same. The test set utilized in this study is identical to the original SHL challenge dataset. To mitigate the issue of overfitting, the early stopping method with patience 5 is employed throughout the training phase, wherein the loss and accuracy on the sub-validation set are continuously monitored. The corresponding hyperparameter values of the model are presented in Table 1.

The model is implemented in the Keras framework and trained on eight GTX 1080 Ti GPUs in less than one hour. While the inference was computed on a single GTX 1080 Ti GPU. We reported more detailed computational performance in Tables 5 and 6. The model starts with randomized weights at the training time, leading to potential variations in the following evaluation metrics across different training sessions. We run the training process ten times and report the optimal results.

To assess and compare the performance of our proposed network with existing approaches, we employ accuracy, and the macro-averaged F_1-score (Dalianis, 2018) as indicated in (5).

$$F_1 = \frac{1}{C}\sum_{k=1}^{C}\frac{2\cdot \mathrm{recall_k}\cdot \mathrm{precision_k}}{\mathrm{recall_k}+\mathrm{precision_k}}, \tag{5}$$

where C is the number of transportation modes, $\mathrm{recall_k}$ and $\mathrm{precision_k}$ are the recall and precision of class k, respectively.

The proposed model is designed as a sequence-to-one model, wherein the predicted class remains consistent inside a frame regardless of the number of samples present. Hence, it is imperative to acknowledge that the accuracy and F_1-score represent the performance metrics based on per frame. In line with the *SHL challenge 2018*, where the performance per sample was reported, we also provide an evaluation of the *accuracy* and *F_1-score per sample* for model comparison. The analysis of the impact of the majority labeling policy is conducted in Section 4.6.

4.2. Results

By utilizing a time window of 60 s and a downsampling frequency of 20 Hz, the model achieves an accuracy of 95.1% and a F_1-score of 94.2%. Moreover, Fig. 4 depicts the learning curve. The initial epoch exhibits fluctuating validation accuracy and loss, which subsequently stabilize. The training process concludes at epoch 35 as a result of the use of early stopping. Based on the absence of an increasing trend in the validation loss curve, it is inferred that overfitting was mitigated throughout the training process.

Table 2
Confusion matrix with frequency 20 Hz in time window 60 s.

		Predicted labels							
		Still	Walk	Run	Bike	Car	Bus	Train	Subway
Ground truth	Still	923	6	0	1	5	15	9	2
	Walk	9	719	2	0	0	0	1	0
	Run	0	1	336	0	0	0	0	0
	Bike	3	0	0	508	0	0	0	0
	Car	10	0	0	0	1249	12	3	2
	Bus	40	4	0	0	18	826	4	9
	Train	41	2	0	0	13	4	541	46
	Subway	7	0	0	0	0	0	19	308
	Recall	96.0	98.4	99.7	99.4	97.9	91.7	83.6	92.2
	Precision	89.4	98.2	99.4	99.8	97.2	96.4	93.8	83.9
	F_1-score	92.6	98.3	99.6	99.6	97.5	94.0	88.4	87.9

Table 2 presents the per-class confusion matrix, precision, recall, and F_1-score at a downsampling frequency of 20 Hz and a time window of 60 s. The F_1-scores for the categories of Walk, Run, Bike, and Car in the confusion matrix demonstrate a high level of effectiveness in classification (over 97%). In contrast, the F_1-scores for the Train and Subway modes exhibit considerably lower values. Similar observations of frequent misclassification between Train and Subway modes can be found in previous studies analyzing the SHL dataset (Gjoreski et al., 2018). We conjecture that the motion dynamics of trains and subways are very similar, especially in the city environment when the trains and subways travel at similar speeds. Extra information, such as location-based data, may be needed to distinguish between them. We leave this aspect as our future work.

4.3. Evaluation on time window and downsampling frequency

The proposed model was evaluated by altering the time window lengths and downsampling rates. The time frame duration of 60 s was initially studied, as specified in the SHL dataset. Following this, we evaluated shorter durations for the window lengths, specifically 30 s, 20 s, 10 s, and 5 s. This choice was made because 60 s is divisible evenly by these durations. The frequencies used for downsampling evaluation included 100 Hz (representing no downsampling), 50 Hz, 25 Hz, 20 Hz, 10 Hz, 5 Hz, and 1 Hz.

The prediction accuracy on the test set, with the specified window lengths and downsampling frequencies, is depicted in Fig. 5. Our findings show a general trend of enhanced model performance as the downsampling frequency augments. However, an exception arises when employing a time window of 60 s. Specifically, with this time window, the accuracy and F_1-scores for the 50 Hz and 100 Hz downsampling frequencies are inferior to those at 20 Hz and 25 Hz where the model obtains highest accuracy 95.1% and F_1-score 94.7%. One plausible explanation is that the extended time series may impede the model's feature extraction capability. Furthermore, exceedingly low downsampling frequencies might overly dilute the information within the data, leading to a degradation in model performance. It is noteworthy to observe that there is a substantial improvement in accuracy when downsampling frequencies transition from 1 Hz to 5 Hz, regardless of the length of the window. In scenarios with shorter time windows, decreased data duration, and inherent information, diminished performance is expected. However, even within a 5-s time window at 50 Hz and 100 Hz, the model's accuracy nearly approaches an impressive 90%. This underscores the model's efficacy in feature extraction and recognition.

4.4. Feature contribution and selection

In Section 3.2, we delineated our data preprocessing approach. After these preprocessing steps, we distilled three distinct features from each sensor: the accelerometer, gyroscope, and magnetometer. These

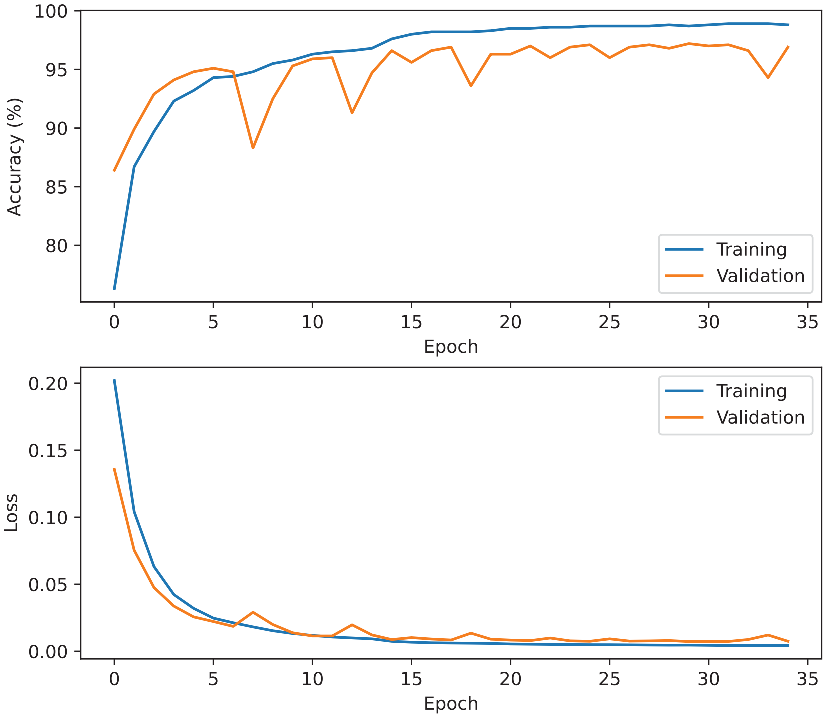

Fig. 4. Learning curve.

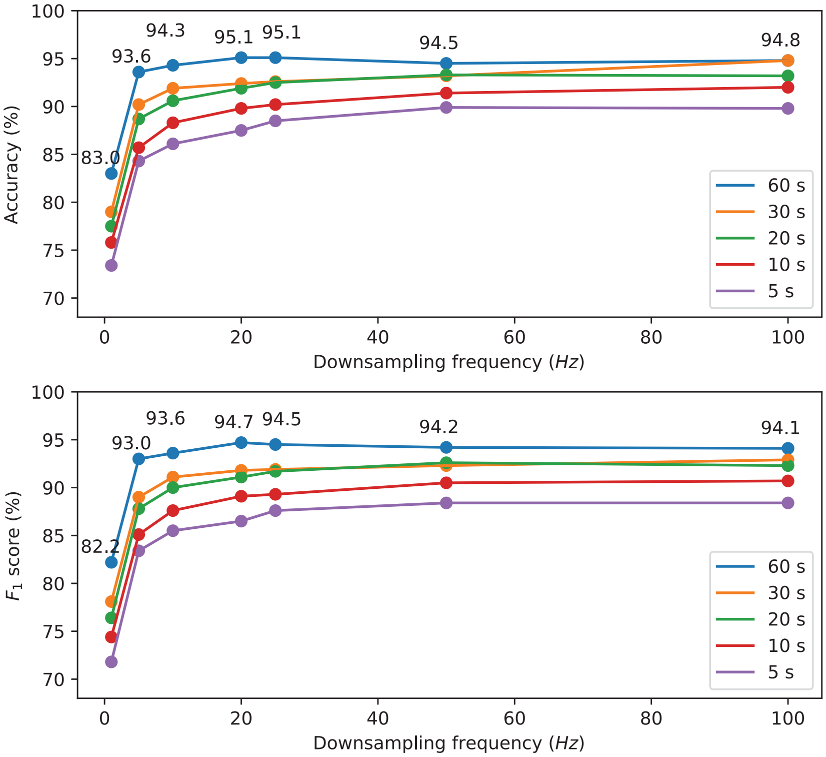

Fig. 5. Evaluation on different time windows and downsampling frequencies.

features include the smoothed and downsampled data from the x-, y-, and z-axis, the magnitude, and the jerk. This section delves into an examination of the significance of these derived features.

Table 3 presents the performance metrics for each feature when employed as a solitary input, implying that the model operates with just a single channel. The reported outcomes pertain to scenarios wherein the time window is fixed at 60 s, the downsampling frequency is established at 20 Hz, and a consistent random seed is deployed during execution to mitigate any fluctuations stemming from system randomness. As evidenced in the table, when considering the acceleration data, the magnitude, when used as a solo input, achieves an accuracy of 89.7% and an F_1-score of 88.9%. Both in terms of accuracy and F_1-score, the

Table 3
Feature contribution from single sensor.

Sensor	Feature			Accuracy	F_1-score
	$x\,y\,z$	M_s	J_s		
Accelerometer	√			76.7	74.0
		√		**89.7**	**88.9**
			√	85.3	82.9
Gyroscope	√			**82.2**	**80.8**
		√		81.6	79.0
			√	77.3	76.0
Magnetometer	√			62.9	61.4
		√		61.6	61.4
			√	**72.4**	**74.5**

Table 4
Feature contribution from multiple sensors.

No.	Sensor			Feature			Accuracy	F_1-score
	A	G	M	xyz	M	J		
1	√				A	A	90.3	88.6
2		√		G	G		86.5	84.8
3	√	√		G	A, G	A	91.9	90.3
4	√		√		A	A, M	93.9	93.2
5		√	√	G	G	M	88.3	88.4
6	√	√	√	G	A, G	A, M	95.0	94.4

A: Accelerometer; G: Gyroscope, M: Magnetometer.

jerk feature outperforms the x-, y-, and z-axis data by an approximate margin of 10%. when considering the gyroscope data, the smoothed & downsampled x-, y-, and z-axis and magnitude are markedly superior to the jerk feature. However, concerning the magnetic field data, jerk exhibits a notably enhanced performance compared to other features. Therefore, according to our analyses, we assert that these mentioned features substantially contribute to the model's performance.

In light of the notable performance exhibited by the individual features, it becomes imperative to investigate if amalgamating them might further enhance the outcomes. Table 4 elucidates the results stemming from the integration of various features both within a single sensor and across different sensors. Regarding the acceleration data, when magnitude and jerk are incorporated concurrently as dual channels, the model yields an accuracy of 90.3% and an F_1-score of 88.6%, surpassing the performance of any singular feature. Similarly, for the gyroscope data, a synthesis of the smoothed & downsampled xyz with magnitude culminates in an accuracy of 86.5%, which is notably superior to the 82.2% achieved by the smoothed & downsampled xyz in isolation and the 81.6% procured by the magnitude independently. These findings corroborate the proposition that strategic combinations can amplify the model's efficacy.

Subsequently, we orchestrated a pairing of salient features from the accelerometer, gyroscope, and magnetometer, and then integrated these pairs with the prominent features from the trio of sensors. The Spearman's correlation test is carried out to further validate the combination of magnitudes derived from the accelerometer and gyroscope, and the combination of jerks derived from the accelerometer and magnetometer. The magnitudes derived from the accelerometer and gyroscope show a non-significant weak correlation (r = 0.13, p-value = 0.71). Also, the jerks derived from the accelerometer and magnetometer in the $x-$, $y-$, and z-axis are r = 0.39 (p-value = 0.24), r = −0.09 (p-value = 0.79), and r = −0.15 (p-value = 0.67), respectively, showing no significant strong correlation. These indicate that those features are independent, and discern that specific combinations further enhance the model's performance. Finally, with the combination of three sensors, the accuracy of the model reached 95.0%. Although this marginally trails the apex accuracy of 95.1% delineated in Section 4.2, given the inherent randomness of the system, such a discrepancy is deemed within acceptable bounds.

4.5. Ablation study

Two ablation studies are conducted to ascertain the individual contributions of several factors to the model's performance. The ablation study is employed to remove convolutional layers to get an insight into their contribution as the depth of the model increases. Additionally, it investigates the connections between the biLSTM and convolutional layers to assess the impact of the feature pyramid.

Table 5 displays the results of ablating convolutional layers. Four tests were conducted. Initially, the model preserved only the initial convolutional layer, gradually augmenting the convolutional layers until reaching five, as outlined in our proposed model. In this ablation experiment, our focus is solely on the contribution of the convolutional layer. Therefore, the connection relationship that forms the feature pyramid remains unchanged throughout the experiment. Table 4 demonstrates a positive correlation between the number of convolutional layers and the model's performance. Notably, the third convolutional layer has the most significant impact, resulting in a significant improvement in accuracy from 78.6% to 94.0%. Subsequently, the task of enhancing accuracy encountered heightened challenges. Also, upon the inclusion of the fifth convolutional layer, a notable enhancement in accuracy was observed, with an increase of 0.9%. This advancement can be considered substantial in the context of the study. Likewise, the number of parameters, duration of training, and floating point operations per second (FLOPS) all exhibit an upward trend as the model's depth increases, with the most pronounced variations observed specifically at the third convolutional layer. It can be seen that with the addition of more convolutional layers, the complexity of the model increases. While the model remains with a fast inference speed, *i.e.,* the complete model (PFbiLSTM) takes only 2.3 ms to detect the transportation mode for each frame.

The connection between the biLSTM layer and the convolutional layer is depicted in Table 6 to explore the effect of the feature pyramid. Except for the fourth convolutional layer and biLSTM, all the other convolutional layers are connected in our proposed model. To investigate the function of these connections, the ablation experiment interrupted some of the connections or added a fourth convolutional layer to the biLSTM. In cases No. 1–3, the initial, second, and third convolutional layers are sequentially disconnected. When the connection between the three convolutional layers is disconnected and only the connection between the fifth convolutional layer is retained, the accuracy is as high as 94.8%, which is significantly higher than the accuracy of Tang et al. (2022) (91.2%), indicating the improvement brought by feature selection. After connecting all convolutional layers to the biLSTM layer, the accuracy falls to 94.5%. As we propose, the optimal combination consists of concatenating the first, second, third, and fifth convolutional layers (an accuracy of 95.1%). However, the number of parameters, training time, and inference time increase considerably as the number of connections grows. This ablation study provides a hint for selecting model effectiveness and efficiency.

4.6. Influence of labeling policy

The majority labeling strategy has been implemented in various studies, including Gjoreski et al. (2020) and Richoz et al. (2020), to handle transportation mode transitions in frames. Transition refers to multiple modes presented inside a frame, indicating the change of transportation modes from one data sample to the next one. The labeling policy that assigns the majority label to the entire frame, considering it as a single transportation mode, is likely to have a detrimental impact on the model's performance.

Table 7 presents the difference in F_1-scores between the per frame and sample approaches. As the time window lowers, there is a drop in the fraction of transition frames among all frames in both the training set and test set. The time window is reduced and the transportation mode is more likely to appear in only one frame. As the ratio drops,

Table 5
Ablation study on application of convolutional layers.

No.	Conv layers					Accuracy	# Params	Train	Test	FLOPS
	1st	2nd	3th	4th	5th	(%)	(M)	(min)	(ms)	(G)
1	√					73.9	0.3	11.6	1.1	0.7
2	√	√				78.6	0.9	10.0	1.6	6.8
3	√	√	√			94.0	1.6	21.1	1.9	13.0
4	√	√	√	√		94.1	2.7	21.8	2.1	16.1
Ours	√	√	√	√	√	95.1	3.1	24.2	2.3	19.2

Table 6
Ablation study on feature pyramid.

No.	Conn. biLSTM & Conv					Accuracy	# Params	Trai	Test	FLOPS
	1st	2nd	3th	4th	5th	(%)	(M)	(min)	(ms)	(G)
1		√	√		√	94.6	2.7	11.0	1.58	19.2
2			√		√	93.9	2.2	8.8	1.05	19.2
3					√	94.8	1.8	8.9	0.87	19.2
4	√	√	√	√	√	94.5	3.9	26.3	2.46	19.2
Tang et al. (2022)					√	91.2	2.1	10.1	1.6	57.7
Ours	√	√	√		√	95.1	3.1	24.2	2.3	19.2

Table 7
F_1-score difference between per frame and per sample.

Window	Training	Test set	F_1-score difference (%)						
size (s)	set ratio	ratio	1 Hz	5 Hz	10 Hz	20 Hz	25 Hz	50 Hz	100 Hz
60	3.99	3.97	0.29	0.53	0.53	0.51	0.52	0.63	0.50
30	1.99	1.98	0.11	0.07	0.12	0.13	0.11	0.07	0.13
20	1.33	1.33	0.05	0.09	0.12	0.14	0.16	0.14	0.13
10	0.66	0.66	0.03	0.05	0.07	0.06	0.07	0.05	0.05
5	0.33	0.33	0.01	0.02	0.02	0.02	0.02	0.00	0.02

there is a corresponding decrease in the difference between the F_1-score calculated per frame and per sample. When the time window duration is set to 60 s, within the downsampling frequency range of 10 Hz to 100 Hz, the F_1-score per frame exhibits an average increase of over 0.5% compared to the F_1-score per sample. When the duration of the time interval is set to 30 s, the magnitude of this discrepancy is noticeably diminished. In most cases, where the duration is shorter than 30 s, the disparity is typically around 0.1%, which can be practically considered inconsequential.

4.7. Comparison with previous work

In this section, the *SHL challenge 2018* dataset was utilized by all previous studies participating in the comparison. In other words, the comparison did not encompass utilizing additional data sources such as video and audio (Richoz et al., 2020). To maintain consistency in our analysis, we employed the F_1-score per sample as a benchmark when comparing our findings to those of prior studies.

Table 8 displays the comparison results with previous studies utilizing traditional machine learning and deep learning techniques. Gjoreski et al. (2020) (ML+DL) demonstrates enhanced performance by applying preprocessing techniques, incorporating seven distinct sensor types, implementing complicated feature extraction methods, and deep multimodal spectro-temporal fusion. Namely, ML+DL integrated a meta-model that combines deep learning and machine learning-based models. This was followed by a post-processing step involving the application of Hidden Markov Model (HMM) smoothing. By applying so, ML+DL achieved the highest F_1-score 94.9%, but with the model size of 500 MB. Nevertheless, our model FPbiLSTM exhibits a reduced weight and requires low resources for preprocessing, enabling it to categorize input data without necessitating subsequent post-processing. Upon utilizing the data from the accelerometer, gyroscope, and magnetometer, FPbiLSTM exhibits a compact size of merely 36 MB, while achieving a commendable F_1-score of 94.2%. The size of FPbiLSTM is greater than that of CNN+BiLSTM (Tang et al., 2022) measured in 24 MB, while it has exhibited a notable enhancement in the F_1-score, demonstrating a 4% increase.

Table 8 also contrasts FPbiLSTM with conventional machine learning approaches. XGBoost (Janko et al., 2018) demonstrates a superior F_1-score (92.4%), albeit with increased complexity resulting from the utilization of seven distinct sensor types, resulting in a model size of 43 MB. Taking into account various elements such as the volume of sensing data, the utilization of a subset of the SHL dataset (Gjoreski et al., 2018), and the avoidance of intensive preprocessing for feature extraction, FPbiLSTM is more lightweight and effectively captures essential characteristics of diverse transportation modes.

4.8. Discussion

This paper presents a comprehensive empirical study on the contribution and selection of features for transportation mode detection using motion-based sensory data. The accelerometer measures the rate of change of velocity experienced by the mobile device during travel. This measurement varies significantly across different transportation modes, such as walking steadily or boarding a train or subway that is about to depart from a station. The gyroscope records the angular velocity of the transportation mode. In the case of traveling by car, train, or subway, the angular velocity tends to be less dynamic compared to activities like running or walking in crowded environments. The magnetometer detects alterations in the Earth's magnetic field. These alterations are more pronounced during significant changes in location over a short period, such as those experienced while traveling by car, train, or subway, as opposed to minor location adjustments, like remaining still or walking slowly. As discussed in Section 4.4, it is apparent that the accelerometer, gyroscope, and magnetometer provide valuable motion features for mode detection, even when used individually. The performance is further bolstered by augmenting the motion information captured from these sensors. However, the efficacy of augmentation hinges on how the motion features are utilized. In this context, the magnitude and jerk features prove to be more robust compared to the x-, y-, and z-axis readings. This is illustrated in Table 4, where augmenting the magnitude and jerk features from the accelerometer and magnetometer leads to performance gains. Conversely, augmenting the axis readings from different sensors may result in a performance decline. This discrepancy arises because the axis feature heavily relies on the orientation of the mobile devices, while during travel, the mobile devices move dynamically with the travelers. These findings align with previous studies by Liang et al. (2019), Fang et al. (2017) and Iskanderov and Guvensan (2020).

Table 8
Comparison with state-of-the-art models.

Method	Sensing data							Input	Post process	Model	F_1-score
	A	G	M	L	g	O	Ap			size (MB)	
Random forest (Antar et al., 2018)	√	√	√	√	√	√	√	F	MV	1122	87.5
MLP (Widhalm et al., 2018)	√	√	√				√	F	HMM	0.04	87.5
XGBoost (Janko et al., 2018)	√	√	√	√	√	√	√	F	–	43	92.4
DNN (Akbari et al., 2018)	√	√	√	√	√	√	√	F	MV	84	86.3
CNN (Ito et al., 2018)	√	√						S	–	3	88.8
CNN+BiLSTM (Tang et al., 2022)	√		√					R	–	24	90.4
ML+DL (Gjoreski et al., 2020)	√	√	√	√	√	√	√	S+F	HMM	500	94.9
FPbiLSTM (ours)	√	√	√					R	–	36	94.2

A: Accelerometer; G: Gyroscope, M: Magnetometer; L: Linear; g: Gravity.
O: Orientation; Ap: Ambient pressure.
F: features; S: spectrogram R: raw data.
MV: Majority Voting; HMM: Hidden Markov model.

An empirical observation reveals a trade-off between computational cost and the number of convolutional layers in the PFbiLSTM model, as shown in Table 6. With an increase in the convolutional layers, the overall performance of PFbiLSTM improves, albeit at the expense of longer inference times as the model complexity escalates. Thus, it is viable to adjust the model's size and inference speed based on available computational resources and desired accuracy levels. This is especially important for edge devices with limited computation and power capacity.

Limitation. The PFbiLSTM model exhibits constrained performance when distinguishing between subway and train modes. This limitation arises due to their strikingly similar motion patterns, which pose a challenge for accurate classification. To address this issue in future research endeavors, we plan to delve deeper into incorporating supplementary information. Specifically, we aim to integrate location-based sensory data obtained from GNSS and GPS. Leveraging such additional data streams holds promise for enhancing mode detection capabilities by providing contextual information that complements the motion-based features. By incorporating location data alongside motion patterns, we anticipate improving the model's accuracy and robustness in discriminating between subway and train modes, thus enriching the overall performance of the transportation mode detection system. Moreover, we will explore more advanced deep learning models, such as Transformer (Vaswani et al., 2017), to increase the robustness and generalization of transportation mode detection by *e.g.,* reducing the frame duration or testing on other datasets. We also realize that the performance of the mode detection problem is influenced by the smartphone positions and user behavior which the practical application must concern.

5. Conclusion

The ubiquity of smartphones equipped with IMUs offers a unique opportunity to harness a vast range of sensor data for numerous applications, most prominently, transportation mode detection. We aim to propose the Feature Pyramid biLSTM model, drawing its roots from both the FPN and the CNN biLSTM model, innovatively incorporating sensor data from accelerometers, gyroscopes, and magnetometers. Despite its structural complexity, this integration presented a compelling case of efficiency and reduced computational demand. Benchmarking against prior research offered a testament to the model's prowess. While some techniques previously registered higher performance metrics, they often came laden with increased computational burdens, intensive preprocessing, or inflated model sizes. In contrast, our model epitomized efficiency and lightweight design, demanding minimal preprocessing while yielding commendable performance metrics. Notably, the superior performance achieved by utilizing only three sensors with minor computational resources underscores the model's potential for real-world applications, especially in environments constrained by resource availability or processing power.

The empirical study using the *SHL challenge 2018* unraveled the nuanced relationships between downsampling frequencies, time window lengths, and their impact on model performance. Namely, certain time windows and frequencies augmented the model's capabilities, while exceedingly low downsampling frequencies and prolonged time series occasionally impeded effective feature extraction. In real applications, a short time window and a low downsampling frequency are desirable to facilitate prompt and economical transport mode detection. In this paper, the optimal setting is one minute at 20 Hz. Through extensive feature analysis, the study underscored the significance of the three most commonly utilized sensors, *i.e.,* the accelerometer, magnetometer, and gyroscope, for transportation mode detection, which aligns with the literature review. As revealed in the feature contribution and selection analysis, a careful combination of these features using their magnitude and jerk attributes can further improve the detection performance and mitigate the risk of overfitting. Moreover, the ablation study indicates that the number of convolutional layers included in the FPbiLSTM model impacts not only the performance but also the computational overhead. A trade-off can be made by reducing the convolutional layers to meet the constraints of computational capacity for edge devices, such as smartphones. Last but not least, location-based information can be further explored to overcome the difficulties in distinguishing similar transportation modes, such as taking a train or subway. Nevertheless, the location-based information should be anonymized to avoid potential privacy risks.

CRediT authorship contribution statement

Qinrui Tang: Conceptualization, Funding acquisition, Methodology, Writing – original draft, Writing – review & editing. **Hao Cheng:** Methodology, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This research is supported by the German DLR-funded project *VMo4Orte*. Hao Cheng is funded by the Marie Skłodowska-Curie Actions (MSCA) European Postdoctoral Fellowship with the 101062870 - VeVuSafety project.

References

Akbari, A., Wu, J., Grimsley, R., Jafari, R., 2018. 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. pp. 1596–1605.

Antar, A.D., Ahmed, M., Ishrak, M.S., Ahad, M.A.R., 2018. 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. pp. 1497–1502.

Balabka, D., Shkliarenko, D., 2021. Human activity recognition with AutoML using smartphone radio data. In: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 346–352.

Brewer, K.R.W., 1999. Design-based or prediction-based inference? Stratified random vs stratified balanced sampling. Internat. Statist. Rev. 67, 35–47.

Choi, J.-H., Lee, J.-S., 2019. EmbraceNet for activity: A deep multimodal fusion architecture for activity recognition. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. pp. 693–698.

Dabiri, S., Heaslip, K., 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. Transp. Res. C 86, 360–371.

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

Fang, S.-H., Fei, Y.-X., Xu, Z., Tsao, Y., 2017. Learning transportation modes from smartphone sensors based on deep neural network. IEEE Sens. J. 17 (18), 6111–6118.

Gjoreski, H., Ciliberto, M., Wang, L., Ordonez Morales, F.J., Mekki, S., Valentin, S., Roggen, D., 2018. The university of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. IEEE Access 6, 42592–42604.

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

Iskanderov, J., Guvensan, M.A., 2020. Breaking the limits of transportation mode detection: Applying deep learning approach with knowledge-based features. IEEE Sens. J. 20 (21), 12871–12884.

Ito, C., Cao, X., Shuzo, M., Maeda, E., 2018. 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. pp. 1503–1510.

Janko, V., Gjoreski, M., De Masi, C.M., Reščič, N., Luštrek, M., Gams, M., 2019. Cross-location transfer learning for the sussex-huawei locomotion recognition challenge. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. pp. 730–735.

Janko, V., Reşçiç, N., Mlakar, M., Drobnič, V., Gams, M., Slapničar, G., Gjoreski, M., Bizjak, J., Marinko, M., Luštrek, M., 2018. 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. pp. 1511–1520.

Kalabakov, S., Stankoski, S., Reščič, N., Kiprijanovska, I., Andova, A., Picard, C., Janko, V., Gjoreski, M., Luštrek, M., 2020. Tackling the SHL challenge 2020 with person-specific classifiers and semi-supervised learning. In: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers. pp. 323–328.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations. ICLR.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25.

Liang, X., Zhang, Y., Wang, G., Xu, S., 2019. A deep learning model for transportation mode detection based on smartphone sensing data. IEEE Trans. Intell. Transp. Syst. 21 (12), 5223–5235.

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Mishra, R., Gupta, A., Gupta, H.P., Dutta, T., 2020. A sensors based deep learning model for unseen locomotion mode identification using multiple semantic matrices. IEEE Trans. Mob. Comput..

Munoz Diaz, E., Rubio Hernan, J.M., Jurado Romero, F., Karite, A., Vervisch-Picois, A., Samama, N., 2023. Advanced smartphone-based identification of transport modes: Resilience under GNSS-based attacks. Future Transp. 3 (2), 568–583.

Qin, Y., Luo, H., Zhao, F., Wang, C., Wang, J., Zhang, Y., 2019. Toward transportation mode recognition using deep convolutional and long short-term memory recurrent neural networks. IEEE Access 7, 142353–142367.

Ren, Y., 2021. Multiple tree model integration for transportation mode recognition. In: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 385–389.

Richoz, S., Wang, L., Birch, P., Roggen, D., 2020. Transportation mode recognition fusing wearable motion, sound, and vision sensors. IEEE Sens. J. 20 (16), 9314–9328.

Saha, P., Alam, M.M., Tapotee, M.I., Baray, S.B., Ahad, M.A.R., 2021. An empirical approach for human locomotion and transportation recognition from radio data. In: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 390–395.

Savitzky, A., Golay, M.J., 1964. Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. 36 (8), 1627–1639.

Tang, Q., Jahan, K., Roth, M., 2022. Deep CNN-BiLSTM model for transportation mode detection using smartphone accelerometer and magnetometer. In: 2022 IEEE Intelligent Vehicles Symposium. IV, pp. 772–778.

Titterton, D., Weston, J., 2004. Strapdown Inertial Navigation Technology - 2nd Edition. The institution of Engineering and Technology, London, UK and The American Institute of Aeronautics, Virginia, USA.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. (NeurIPS) 30.

Widhalm, P., Leodolter, M., Brändle, N., 2018. 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. pp. 1479–1487.

Wu, J., Akbari, A., Grimsley, R., Jafari, R., 2018. 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. pp. 1571–1578.

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

Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. CoRR abs/1505.00853, arXiv:1505.00853.

Yu, M.-C., Yu, T., Wang, S.-C., Lin, C.-J., Chang, E.Y., 2014. Big data small footprint: The design of a low-power classifier for detecting transportation modes. Proc. VLDB Endow. 7 (13), 1429–1440.

Zhu, Y., Luo, H., Chen, R., Zhao, F., Su, L., 2020. DenseNetX and GRU for the sussex-huawei locomotion-transportation recognition challenge. In: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers. pp. 373–377.

Zhu, Y., Luo, H., Guo, S., Zhao, F., 2021. Data mining for transportation mode recognition from radio-data. In: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp. 423–427.

Zhu, Y., Zhao, F., Chen, R., 2019. Applying 1D sensor DenseNet to Sussex-Huawei locomotion-transportation recognition challenge. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. pp. 873–877.