

## Research Article

# Wi-CL: Low-Cost WiFi-Based Detection System for Nonmotorized Traffic Travel Mode Classification

Runnan Xu <sup>1</sup>, Zilin Huang <sup>2</sup>, Sikai Chen <sup>2</sup>, Jinlong Li <sup>3</sup>, Pan Wu <sup>3</sup>,  
and Yongjie Lin <sup>3</sup>

<sup>1</sup>School of Computer, Guangdong University of Science and Technology, Dongguan 523083, China

<sup>2</sup>Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

<sup>3</sup>School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510000, China

Correspondence should be addressed to Zilin Huang; [zilin.huang@wisc.edu](mailto:zilin.huang@wisc.edu)

Received 18 September 2022; Revised 26 February 2023; Accepted 24 May 2023; Published 14 June 2023

Academic Editor: Wen Liu

Copyright © 2023 Runnan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic travel mode identification and classification are crucial for the development of intelligent transportation systems (ITSs). At present, scholars have investigated the classification of motorized and nonmotorized traffic travel in various road environments; however, the classification of walking and bicycle modes in nonmotorized travel has been largely ignored. Therefore, in this paper, we investigate nonmotorized traffic travel and propose a new low-cost nonmotorized traffic travel mode classification system, known as the Wi-Fi classification (Wi-CL) system that uses Wi-Fi signal detectors and the refined characteristics of nonmotorized travel modes. The Wi-CL system includes four modules: data acquisition module, data processing module, feature extraction module, and mode classification module. In the data acquisition module, the proposed system detects the Wi-Fi signals of traffic participants in road environments. In addition, we propose a received signal strength indicator (RSSI) filtering algorithm for hybrid traffic networks that effectively addresses surrounding obstacles and environmental noise. In the feature extraction module, we extract relevant traffic features to construct a mode classification model. Finally, a recurrent neural network (RNN) framework based on the long short-term memory (LSTM) algorithm is successfully implemented in the mode classification module for traffic travel mode identification. To validate the effectiveness of the Wi-CL system, extensive experiments were conducted using field data collected by Wi-Fi detectors installed at the South China University of Technology (SCUT). The experimental results show that the proposed RSSI filtering algorithm achieves excellent signal filtering results in real road traffic environments. In addition, the constructed travel speed estimation algorithm outperforms other baseline models in four different scenarios (flat-peak walking, midday peak walking, flat-peak cycling, and midday peak cycling), achieving an overall classification accuracy of 97.92%. In summary, our Wi-CL system is a feasible approach for nonmotorized traffic travel mode classification.

## 1. Introduction

Intelligent transportation systems (ITSs) and city surveillance systems are important applications of the Internet of Things (IoT) [1, 2] that apply the sensing, control, and communication technologies of ground transportation systems through IoT devices to improve the safety and smoothness of urban road networks [3]. Traffic mode detection plays an important role in helping urban planners and transportation agencies determine the occupancy of road resources by different traffic modes at various times [4]. This information can be used to plan, design,

and operate the multimodal infrastructures required by transportation network users [5]. Additionally, extracting traffic mode information over short periods can help city planners monitor anomalies in traffic networks. Model-based information has also been utilized in other fields, e.g., private route recommendation [6], daily commuting surveys [7], and congestion prediction [8]. Although most existing studies investigated motorized transportation, several scholars have realized the importance of nonmotorized transportation modes in urban transportation systems and have attempted to optimize these nonmotorized transportation modes [9].

Movement data obtained from pedestrians and cyclists are critical in modeling travel behavior and habits, especially in urban surveillance systems [10]. However, collecting travel data from pedestrians and cyclists on streets, sidewalks, and public trails is a considerable challenge [11]. In addition, widely used vehicle sensors (such as induction loops, cameras, ultrasonic sensors, and radars) suffer from several problems, including high installation and maintenance costs and inefficiencies in pedestrian and cyclist detection and tracking, because pedestrians and cyclists usually have weak behavioral norms, in contrast to motor vehicles, which are restricted by lane rules [12]. Thus, automatic nonmotorized traffic data collection and mode detection techniques should be developed and applied in practical scenarios.

To address these problems, several studies have proposed special sensors for pedestrian counting, such as infrared sensors, ultrasonic sensors, and pressure footpads [13]. However, these sensors provide data for only specific points in a network. More importantly, point-based counting techniques fail to detect the same person at different points in the network when determining travel routes, destinations, and travel times [14]. Currently, as a low-cost alternative [15], location-aware technology has attracted considerable interest in tracking pedestrian and nonmotorized movements due to the popularity and development of smart mobile devices such as cell phones and tablets. Most early studies on location-aware technologies used the prevalent and well-established global positioning system (GPS) [16], global system for mobile communications (GSM) [17], and accelerometers as data sources [18]. Among them, GPS-based approaches require users to install and run a mobile application to actively transmit GPS records to the center, which is not convenient in the real world [19]. In some studies, GSM data has been suggested as an effective way to track cell phones based on their cellular signal strength. However, the location estimation is very coarse and only appropriate for O-D surveying [17]. Thus, our study focuses on the receiving signal strength indicator (RSSI) values captured by wireless communication signals (such as bluetooth and Wi-Fi). Specifically, in wireless channels, RSSI values have a mathematical relationship with the 2D distance [20], i.e., the plane distance between the smart mobile device and the detector. Also, it has been reported that the detection rate of bluetooth devices is usually between 5% and 12% since most applications involving Bluetooth technology are carried out in the vehicle network [21]. In contrast, Wi-Fi-enabled smart devices periodically attempt to connect to wireless LAN (WLAN) when sending detection request data. Thus, a simple low-cost monitoring unit that is independent of user participation is sufficient for passively collecting these data, and this device does it involve any hardware or software modifications [22].

Given the advantages of Wi-Fi detector data for movement mode detection and the fact that previous studies involving Wi-Fi detector data have focused on limited indoor movements, we explored the feasibility of using Wi-Fi detector data to identify nonmotorized traffic modes in urban road environments. In the literature [5, 10, 22], several traffic mode classification systems based on Wi-Fi detector data have been proposed. The general steps for determining

traffic modes using Wi-Fi detection data can be summarized as follows: first, offline Wi-Fi data are collected, and the data are cleaned to remove errors and redundant information; then, relevant features are extracted from the cleaned data and delivered to a classifier for training; and finally, the traffic modes are predicted based on the trained classifier. However, these systems face several problems: (1) the most difficult challenge is the lack of accuracy in distinguishing modes with similar speeds, accelerations, or routes [23]. When different traffic modes have similar speeds, accelerations, or movement routes, methods for separating these routes are not sufficiently accurate. In particular, existing methods cannot be used to directly distinguish walking from cycling, even if the approaches are valid for vehicle classification. (2) Most previous works did not consider noise in RSSI signals. In addition, the ground data as the main classification feature were not validated according to the expected movement speed. (3) Traditional machine learning classification models, such as logistic regression (LR), support vector machines (SVM), and multilayer perception (MLP), cannot adapt to complex urban traffic environments, resulting in reduced classification accuracy.

To address the above issues, we propose a real-time traffic monitoring system that automatically and accurately identifies walking and cycling travel modes in mixed traffic networks using commercial Wi-Fi detectors. Moreover, we tested the proposed system in a realistic traffic environment at the South China University of Technology (SCUT) campus. The main contributions of this paper can be summarized as follows:

- (1) We construct a novel, low-cost, portable Wi-Fi classification (Wi-CL) system that identifies fine-grained nonmotorized traffic modes by using only Wi-Fi detection data as the information source. Notably, the proposed system is nonintrusive and can be retrofitted using existing road infrastructures, such as posts, walls, or barriers.
- (2) We propose and validate an RSSI filtering algorithm for mixed traffic networks that suppresses the ambient noise caused by surrounding obstacles. The experimental results show that the average error in the walking traffic mode is 5.74 m, which is 27.78% less than that of the conventional constant velocity filtering (CVF) algorithm and 7.57% less than that of the Kalman filtering (KF) algorithm. Similarly, the average error in the cycling traffic mode is 5.53 m, which is 19.74% and 18.68% less than the errors in the CVF and KF algorithms, respectively.
- (3) We design a recurrent neural network (RNN) model based on a long short-term memory (LSTM) network to identify and classify different nonmotorized traffic modes. We extract features from the raw data instead of using the RSSI raw data itself. The evaluation results indicate that the recognition accuracy of the LSTM model is 25%, 18.78%, and 8.34% better than that of the conventional LR, SVM, and MLP algorithms, respectively.

The remainder of this paper is organized as follows. Related work is reviewed in Section 2. Section 3 presents the design idea and framework of the Wi-CL system and the inspiration for identifying nonmotorized traffic modes. Section 4 describes the methodology details. The experimental results and performance evaluation are presented in Section 5. Section 6 discusses the conclusions and outlook of this study.

## 2. Related Works

The development and application of nonmotorized traffic data collection and monitoring systems have been investigated in the field of public transportation. The federal highway administration (FHWA) has released the most recent version of its traffic monitoring guide, which includes a new section on monitoring and identifying nonmotorized traffic [11]. Currently, there are two main types of travel data collection methods based on location-aware technologies: user-centric methods and network-centric methods [5].

*2.1. User-Centric Methods.* Active user participation in the data collection process is required in user-centered approaches. Commonly used data sources include GPS data, inertial measurement unit (IMU) data, or a combination of the two. Zheng et al. [16] proposed a supervised learning method that can infer the travel modes of traffic participants (e.g., walking, cycling, driving, and taking the subway) from GPS data alone. However, in this study, the collected GPS data were not sufficient for classification, and the accuracy rate was only 72.8%. Dabiri and Heaslip [24] used convolutional neural networks (CNNs) to predict the travel modes of the original GPS trajectories. The integration of the best CNN configuration resulted in a maximum accuracy of 84.8%. Although neural networks can achieve better classification performance than traditional machine learning methods, using only GPS data reduces the accuracy of the system. Stenneth et al. [25] proposed a method for inferring user traffic travel modes based on geographic information system (GIS) data and knowledge of the underlying traffic network. The results showed that the detection accuracy of the method was 17% better than that of the GPS-only method. Reddy et al. [18] developed a traffic mode classification system using a GPS receiver and an IMU built into a cell phone. The system used a two-stage decision tree approach and a hidden Markov model (HMM) and achieved an accuracy of more than 90%. However, these studies usually involve high operational costs, as they require additional mobile applications. In addition, the excessive time and equipment energy costs for traffic participants increase the difficulty of implementing these methods at large scales to address real-world traffic problems [26].

*2.2. Network-Centric Methods.* Network-centric approaches attempt to collect data passively without network user intervention. The primary data sources for network-centric approaches include Wi-Fi, Bluetooth, and GSM data. Sohn et al. [17] used GSM signals collected by cell phones to

determine whether a person was standing, walking, or driving. A two-stage logistic regression analysis resulted in an average accuracy of 85% for walking and driving detection. GSM signals are suitable for O-D measurements but insufficient for detecting traffic modes [5]. Yang and Wu [27] used Bluetooth data to classify three travel modes: walking, cycling, and driving. However, in this study, 6.12% of driving modes were incorrectly identified as bicycling, and 10.53% of driving modes were identified as driving.

Due to the popularity of Wi-Fi facilities and the prevalence of IoT devices, mobile crowd sensing based on Wi-Fi detection data, such as activity recognition [28], crowd counting [29], and location estimation [30], has become increasingly popular. Abedi et al. [10] compared the efficiency of Wi-Fi and Bluetooth devices for human mobility data collection. Their study showed that Wi-Fi is a more efficient media access control (MAC) address dataset than Bluetooth devices for tracking spatio-temporal movements of pedestrians and cyclists. Lesani and Miranda-Moreno [22] developed a Wi-Fi/bluetooth-based sensor for identifying mixed traffic networks. Kalatian and Farooq [5] used Wi-Fi data collected by smartphones to identify and predict people's traffic travel modes. The results showed that the MLP model had the best prediction accuracy of 86.52%. Unfortunately, most previous studies on Wi-Fi data-based traffic mode detection did not consider RSSI noise. In terms of classification models, Vu et al. [31] proposed a new RNN-based method for identifying traffic modes. The results showed that deep learning methods have faster speeds and higher accuracies than traditional machine learning algorithms with the same learning parameters. However, instead of extracting the LSTM features, they directly input the raw data into the LSTM. Since LSTM models have been shown to have high accuracy in traffic mode detection studies with a large number of classes [32], this study uses LSTM gates for long sequences. Moreover, we extracted a new set of features from the original data instead of using the original data directly.

## 3. System Overview

*3.1. Inspiration for the Proposed System.* The design of the proposed system was inspired by the increasing use of smart electronic devices such as cell phones, laptops, and tablets. Every smart electronic device has a unique MAC address, which is usually expressed as 12 hexadecimal digits [22]. According to the IEEE 802.11 white paper [33], Wi-Fi-enabled smart electronic devices attempt to connect to nearby WLAN by periodically broadcasting probe requests, which are special frames that provide information to particular access points or all nearby access points, including the MAC address of the sender and recognized service set. Wi-Fi-enabled devices broadcast probe signals even when the device is not in use. In addition, each detection request frame from a Wi-Fi-enabled smart electronic device can be captured and stored by Wi-Fi detectors [20]. The vibration of the signals may be caused by the travel speed, travel time, and different traffic trajectories; thus, traffic participants with different travel modes generate distinct RSSI signals,

and Wi-Fi detectors can capture the dynamic characteristics of these signals as pedestrians and cyclists move if the signals are sensitive, as discussed in the literature [10, 22].

Figure 1 compares the RSSI signals acquired in various time domains for different traffic modes. Figure 1 shows that different traffic modes exhibit distinct time-domain characteristics. Specifically, the RSSI signal generated by the cycling mode has a larger first-order derivative than that generated by the walking mode because speeds change more frequently during cycling than during walking. In contrast, the RSSI signals generated by the walking mode have more connections because more time is required for users to walk through the coverage area and for signals to be sent and received repeatedly by the same detector. This figure demonstrates the feasibility of using RSSI signals generated by Wi-Fi detectors to classify different traffic modes. In brief, feature extraction and traffic mode classification techniques can divide traffic modes in mixed traffic networks into two categories, walking and cycling, according to the RSSI signals generated by the traffic participants.

**3.2. System Architecture.** The purpose of this study is to design an enhanced traffic travel mode identification system by exploring the sequence information acquired by Wi-Fi detectors. Figure 2 illustrates the architecture of the Wi-CL system. The figure shows that the Wi-CL system consists of four main modules: a data acquisition module, a data processing module, a feature extraction module, and a mode classification module. The data acquisition module captures the detection requests broadcast by Wi-Fi-enabled devices in the coverage area, recording information such as MAC addresses, RSSI sequences, and timestamps, and collecting the information into packets [34]. The packets are stored in internal databases or transferred to central databases via WLAN. To improve the results, multiple sensors can be placed at a site to increase the probability of capturing packets when scanning the channel. In this study, the pedestrian and cyclist tracking data are anonymous to prevent the potential leakage of personal information. Thus, each fixed MAC address is not associated with any personal information, such as names or phone numbers [35]. The data processing module has three key functions: removing redundant data, and erroneous data generated by motor vehicles; recovering missing data due to packet loss; and reducing signal noise caused by the environment. The feature extraction module extracts the parameters and relevant features of the model, such as the driving speed, number of connections, and first-order derivatives of the RSSI signals. The mode classification module has two key components: LSTM training and prediction. In the first part, the module trains the LSTM model based on the relevant features; in the second part, the signal features corresponding to the MAC addresses are classified into two different nonmotorized traffic modes, namely, walking and cycling, based on the trained LSTM model.

The system is divided into two phases: an offline phase and an online phase. In the offline phase, the system uses Wi-Fi detectors to collect a large amount of smart electronic

device data in the coverage area and performs data processing to integrate the raw data into a new set of features and train the LSTM model. In the online phase, the system converts the detected real-time data packets into feature vectors through the data processing and feature extraction modules and feeds the feature vectors into the previously trained LSTM model to calculate the traffic trips with the highest probabilities.

## 4. Methodology

This section details the operation of the four modules in the proposed system. The main symbols and meanings are shown in Table 1.

**4.1. Data Collection and MAC Address Grouping.** In the coverage area, the Wi-Fi detector passively collects data from all surrounding Wi-Fi-enabled smart electronic devices. The system will encrypt the user's personal and private data and package the MAC address, RSSI signal, and other relevant data [10]. Because each detector captures a large number of MAC addresses during each scan, the raw packets must be grouped according to MAC addresses [16], and the basic idea of this process is shown in Figure 3.

Assume that there are  $m$  detectors deployed along the nonmotorized road lanes. For the  $k_{th}$  target, the time interval between the first and last packets detected by sensor  $m$  can be expressed as equation (1), and if the target is detected only once,  $t_{k,1}^m = t_{k,2}^m$ .

$$\Delta t_k^m = t_{k,2}^m - t_{k,1}^m. \quad (1)$$

The matrix  $G_k^m$  represents the RSSI dataset collected by sensor  $m$  for moving target  $k$  and is sorted in ascending time order. The data samples include the MAC address, timestamp, and RSSI value of target  $k$ , and each row represents a data sample collected by sensor  $m$  at a certain time.

$$G_k^m = \begin{bmatrix} 00: 16: EA: AE: 3C: 40 & 7/10//2019 & 10: 20: 30 & -68 \\ 00: 16: EA: AE: 3C: 40 & 7/10//2019 & 10: 20: 31 & -70 \\ 00: 16: EA: AE: 3C: 40 & 7/10//2019 & 10: 20: 35 & -74 \end{bmatrix}. \quad (2)$$

**4.2. Data Processing.** Since Wi-Fi detectors were not originally designed for traffic sensing, the RSSI values of user devices usually contain more noise in outdoor environments; thus, this noise must be eliminated before the traffic modes can be identified [36]. The processing method has three key steps: filtering anomalous and redundant data, recovering missing data, and eliminating data noise.

**4.2.1. Filtering Abnormal and Redundant Data.** The data collected by Wi-Fi detectors inevitably includes some data from motorized vehicles, even in environments where walking and bicycling are the primary modes of transportation, such as campuses and residential areas. Therefore, the first data processing step for this system is removing

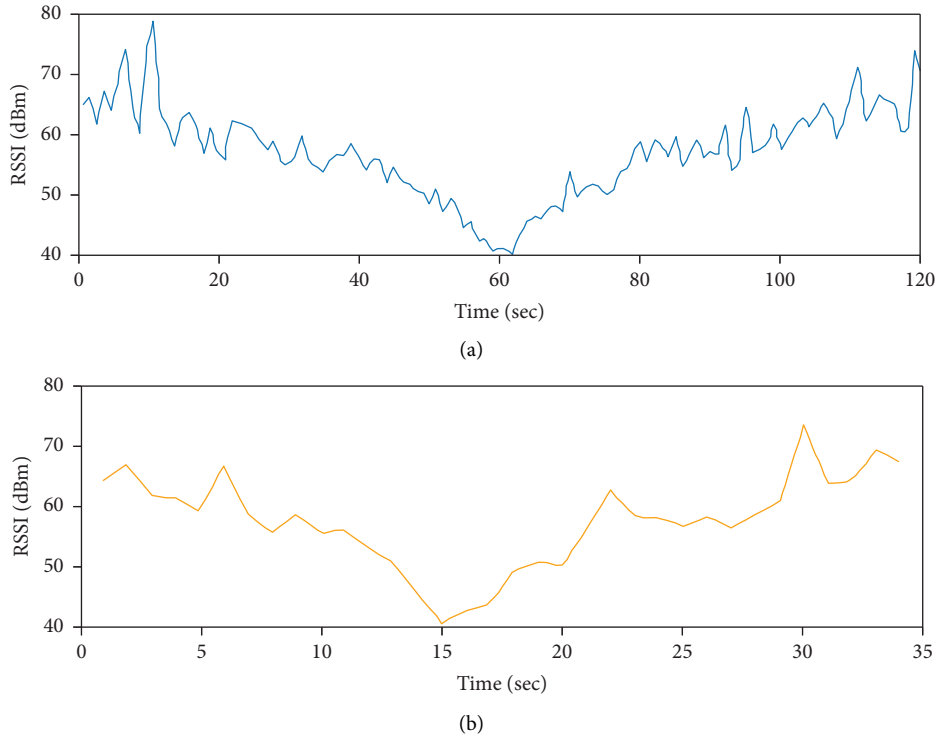


FIGURE 1: Time-domain comparison of RSSI signals for different transportation modes. (a) Walking. (b) Biking.

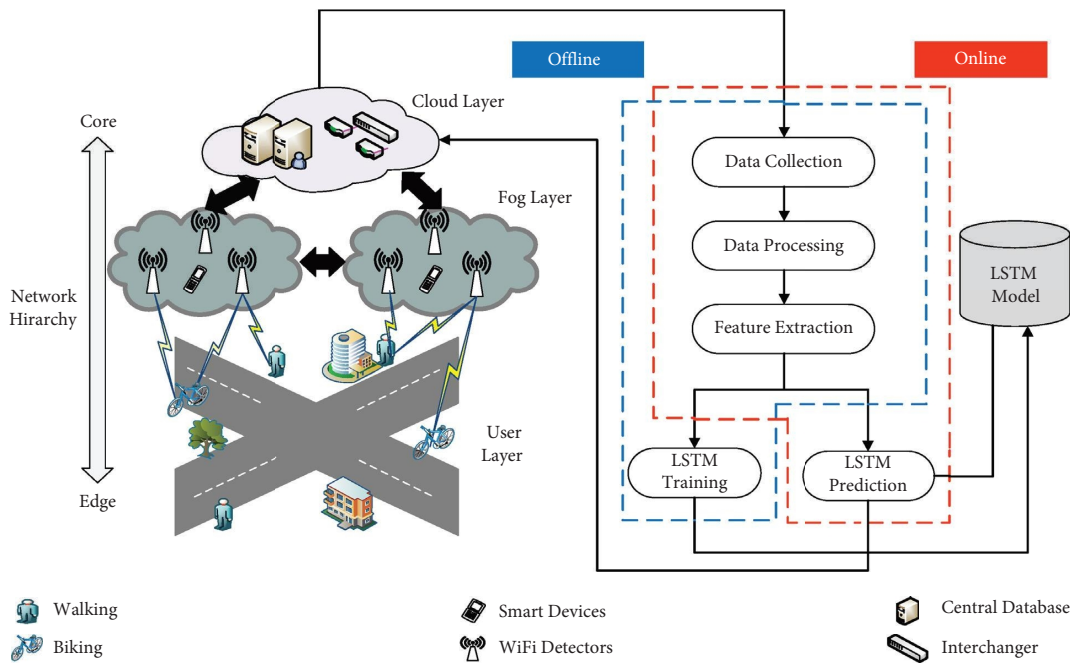


FIGURE 2: The architecture of the Wi-CL system.

potentially inaccurate data generated by motorized vehicles, which can lead to significant errors in the identification of nonmotorized traffic modes.

Similar to the accelerometer-based vehicle detection algorithm [37], RSSI signal-based vehicle detection is based on the fact that vehicles travel faster than nonmotorized

vehicles. Thus, we apply an average speed threshold algorithm to identify RSSI signals generated by motor vehicles [38]. The average operating speed of device  $k$  in the monitoring area of detector  $m$  can be calculated using equation (3), where  $\Delta l_k^m$  denotes the Euclidean distance between target  $k$  and Wi-Fi detector  $m$ .

TABLE 1: Symbols and meanings.

Symbol	Meaning
$t_{k,1}^m$	Timestamp of the first packet of target $k$
$t_{k,2}^m$	Timestamp of the last packet of target $k$
$\Delta t_k^m$	Time interval between the first and last packets
$l_{k,1}^m$	Linear distance between sensor $m$ and target $k$ at the first packet timestamp
$l_{k,2}^m$	Linear distance between sensor $m$ and target $k$ at the last packet timestamp
$\Delta l_k^m$	Distance moved by the target $k$
$n_k^m$	Number of times that sensor $m$ detects target $k$
$S_{k,i}^{ff}$	Value of the $i_{th}$ RSSI signal sequence after classification
$d_{k,t}^{eff}$	Linear distance between target $k$ and detector $m$ at moment $t$

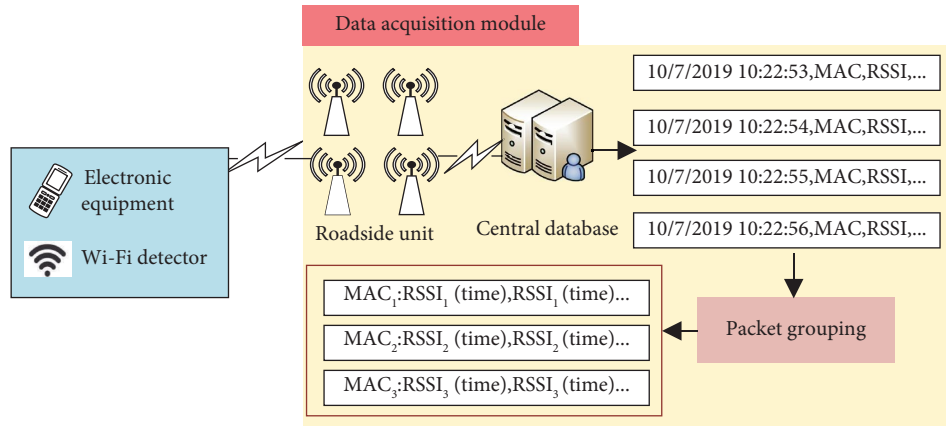


FIGURE 3: Data acquisition with the Wi-CL system.

$$v_k^m = \frac{\Delta l_k^m}{\Delta t_k^m}. \quad (3)$$

Wi-Fi detectors cannot estimate target locations based on single RSSI values collected by one detector. In other words, sensors at different locations may obtain the same RSSI value due to noise interference. To reduce interference caused by inaccurate data, this study assumes that the target is on different sides of the detector when the number of connections is greater than a given threshold [39]. In particular, when target  $k$  moves from the first detected position to the last detected position, the moving distance can be calculated as:

$$\Delta l_k^m = \begin{cases} l_{k,2}^m + l_{k,1}^m, & n_k^m \geq \varepsilon^m, \\ 0, & n_k^m < \varepsilon^m, \end{cases} \quad (4)$$

where  $\varepsilon^m$  is the threshold value for the number of connections, which is related to the antenna characteristics of the detector, antenna type, and detection radius. When  $n_k^m \geq \varepsilon^m$ , the first and last packets are located on different sides of detector  $m$ ; conversely, when  $n_k^m < \varepsilon^m$ , the first and last packets are located on the same side of the detector. If we set  $\Delta l_k^m = 0$ , the mode classification fails because the number of connections is too small. Thus, in this study, the value of  $\varepsilon^m$  was set to 5.

In 2010, Parkin and Rotheram [40] experimentally measured and analyzed volunteers of different ages, genders, bicycle types, and cycling experience in the

Leeds, UK, in different dimensions and for different purposes. The experimental results showed that over the gradient 3% to +3% the eighty-fifth percentile speed varies from 18 kph to 25 kph, and this suggests that 25 kph is a reasonable design speed to adopt for cycle traffic. Since the actual road gradient in this experiment is within the conditions of the above experiment and does not have a sharply changing road alignment, the experimental result of 25 kph, i.e., 7 meters per second, provided in the literature [40] will be used as the threshold to differentiate the speed of bicycle traffic in this experiment.

**4.2.2. Recovering Missing Data.** Within the communication range of Wi-Fi-enabled smart electronic device detectors, detectors may fail to receive detection requests broadcast by Wi-Fi-enabled smart devices due to shielding and environmental factors. If a packet is lost, the RSSI value captured by the detector is displayed as *NULL*. In the literature [41], Dong and Dargie demonstrated that the moving average (MA) method is an applicable filtering method for signal fluctuations. The MA approach uses a set of existing serial data to predict the next phase or phases of data. However, the original MA algorithm changes the entire RSSI signal sequence. Therefore, in this paper, the MA algorithm is improved by interpolating only the missing data [42]. For a certain MAC address  $k$ , the modified MA algorithm can be calculated as follows:

$$S_{k,i}^{\text{ff}} = \begin{cases} \frac{S_{k,i-w}^{\text{ff}} + \dots + S_{k,i-1}^{\text{ff}}}{w}, & S_{k,i}^{\text{ff}} \in \text{NULL}, \\ S_{k,i}^{\text{ff}}, & S_{k,i}^{\text{ff}} \notin \text{NULL}, \end{cases} \quad (5)$$

where  $S_{k,i}^{\text{ff}}$  is the  $i_{\text{th}}$  value of the RSSI signal sequence after excluding inaccurate data and  $w$  is the given window size, which has a considerable impact on algorithm performance. The value of  $w$  was set to 4 in this study.

**4.2.3. Eliminate Data Noise.** In this paper, we show that the distance-based method does not need a lot of pre-training for parameters. RSSI ranging has the advantages of low cost and low time system requirements and is independent of transmission delay, antenna delay, and other factors. The strength of the wireless signal can be used to determine the distance between the transmitting node and the receiving node without requiring additional hardware. Therefore, an a priori database is not required for support. However, the Wi-Fi detector is sensitive to the surrounding environment, and the real-time data collected contains a lot of noise, which needs to be removed before the data is used for traffic pattern recognition. In order to overcome the problems of RSSI signal instability and inaccurate range estimation in ranging methods, scholars have proposed noise reduction preprocessing using Kalman filter [43], Bayesian filter [44], and particle filter [45] according to the characteristics of fluctuating real-time RSSI signal. The main idea is based on iteration. Therefore, if the initial detected RSSI sequence contains large errors, the accuracy of the algorithm will be greatly affected. In our previous work [20], we proposed a constant velocity Kalman filter (CVKF) algorithm for noise reduction. The CVKF algorithm effectively solves the large error problem in the previous observations of the RSSI sequence by embedding a constant speed filter.

**4.3. Feature Extraction.** After the above data processing steps, the filtered RSSI signals are fed into a classifier to distinguish different traffic travel modes. Regardless of the classification technique, dichotomy-free classification is possible only if the signals of different traffic modes do not substantially overlap in feature space [46]. Speed-related variables are the main classification features in the relevant literature [22]. However, the use of speed variables alone does not guarantee satisfactory results. For example, in congested areas, different traffic modes move at similar speeds [5]. Therefore, feature selection is crucial in classification systems.

Although feature selection is usually not necessary in deep neural networks (DNNs), we find that using raw or filtered RSSI signals as input does not provide high prediction accuracy. Various methods have been applied to select the most relevant features for improving classification performance, such as analysis of variance (ANOVA) tests [47] and relative mutual information (RMI) [37]. However,

the use of statistical tests or mutual information to select the top-ranked features is insufficient because the filtered RSSI signals may still be noisy. In the literature [5], Kalatian et al. proposed the ReliefF algorithm for feature selection and assigned different weights according to the importance of the variables, where the basic idea is to estimate the quality of the variables based on their weights to distinguish highly similar detection results.

However, the algorithm takes more time to train and analyze the results since it uses 15 features as input to the classifier. In this study, we identify and select key variables in each category based on the literature results [5]. The variables in this paper include the movement velocity  $v_k^{\text{eff}}$ , number of connections  $n_k^{\text{ff}}$ , and first-order derivatives of the RSSI time series  $S_k^{\text{ff}} = \{S_{k,1}^{\text{ff}}, S_{k,2}^{\text{ff}}, \dots, S_{k,t}^{\text{ff}}\}$ .

The number of connections and first-order derivatives of the RSSI signals can be easily calculated from the processed data; however, the operating speed of device  $k$  is difficult to calculate. In the literature [22], Lesani and Miranda-Moreno used the average travel speed (ATS) [48] as the operating speed variable. However, we found that the speed values estimated by this method are inaccurate because the detection targets often encounter unexpected events, such as extreme weather and congestion, in real traffic environments. Another movement speed estimation method is calculating the ratio of the real-time physical distance to time and converting the filtered real-time RSSI data into the real-time physical distance, which is known as the real-time travel speed (RTS) [49]. For MAC address  $k$ , the real-time travel speed estimation method can be expressed as follows:

$$v_{k,t}^{\text{rtf}} = \frac{d_{k,t+1}^{\text{eff}} - d_{k,t}^{\text{eff}}}{\tau}, \quad (6)$$

where  $v_{k,t}^{\text{rtf}}$  denotes the real-time movement speed of device  $k$  at moment  $t$  in the coverage area,  $d_{k,t}^{\text{eff}}$  is the physical distance between the target and detector at moment  $t$  for device  $k$ , and  $\tau$  represents the time interval between two consecutive detections. The physical distance is based on the filtered RSSI data. Unfortunately, the filtered RSSI signal may still be noisy, so the accuracy of the driving speed calculated by this method needs to be improved. To address this problem, in this paper, we embed the moving average filter into the real-time travel speed and propose a new travel speed estimation method called the real-time filtered travel speed (RFTS).

The key idea of the RFTS algorithm is to convert the RSSI signal into a physical distance between the moving target and the detector. Typically, the most commonly used propagation model to describe the relationship between the RSSI and physical distance is the logarithmic distance path loss model, as shown in equation (7), where  $\lambda$  represents the environment-specific loss parameter and  $B$  denotes the calibrated RSSI value when the distance between the detection target and detector is set to 1 m. For a given  $S_{k,j}^{\text{ff}}$ , the distance estimator can be converted to equation (8), where the values of  $B$  and  $\lambda$  are determined through extensive experiments [20].

$$S_{k,j}^{\text{eff}} = \lambda \times 10 \log_{10} d_{k,j}^{\text{eff}} + B, \quad (7)$$

$$d_{k,j}^{\text{eff}} = 10^{\left( \frac{B - S_{k,j}^{\text{eff}}}{10 \cdot \lambda} \right)}. \quad (8)$$

**4.4. Mode Classification.** Traditional machine learning methods rely heavily on manually extracted features, resulting in issues with feature extraction in machine learning-based image recognition, speech recognition, and natural language processing approaches [50]. Fully connected neural network-based methods also encounter various problems, such as too many parameters and an inability to utilize time series information in the data [51]. As more effective RNN structures have been proposed, the ability of RNNs to mine time-series information and semantic information has been fully utilized, and breakthroughs have been achieved in speech recognition, language modeling, machine translation, and time-series analysis [52].

An RNN is a typical DNN, and the most substantial difference between RNNs and traditional neural networks is that each previous output is sent to the next hidden layer during training in an RNN. Recurrent neural networks portray the relationship between the current output of a sequence and the previous information. Structurally, an RNN remembers the previous information and uses it to influence the output of the following nodes. Thus, the output depends on the current input information and memory units [53].

An RNN has an additional weight, namely, the hidden state of the hidden layer unit, and can process variable-length sequences with a recursive hidden state whose activation depends on the previous state. Therefore, RNNs are suitable for the mutual interpretation of repetitive sequence data.

$$h_t = \begin{cases} 0, & t = 0, \\ \varnothing(h_{t-1}, x_t), & \text{otherwise.} \end{cases} \quad (9)$$

We assume that the input sequence of the classification model is  $x = \{x_1, x_2, \dots, x_t\}$ . At moment  $t$ , the RNN updates its hidden state  $h_t$  according to equation (9).  $\varnothing$  is an activation function, such as a logistic sigmoid with affine transformations. Traditionally, recursive implicit states can be updated by  $h_t = g(Wx_t + Uh_{t-1})$ , where  $g$  is a smooth bounded function and  $W$  and  $U$  are weights.

However, RNNs utilize gradient-based optimization algorithms, increasing the difficulty of training long sequences [54]. In other words, the rate of change of the weights decreases sharply over time, which tends to result in undertraining long sequences [55]. In contrast, LSTM models have memory that can be read, written, and deleted, and these functions allow LSTM models to select the data that should be remembered [15]. In this study, the proposed RNN model includes an LSTM module and an output layer for classification. The structure of the LSTM module is shown in Figure 4.

Here,  $c_t$  is the signal that follows the up line,  $x_t$  is the input vector, and  $h_t$  is the hidden state (value of the recurrent weight). The input and previous hidden state enter the forget gate first. The output of the forget gate  $f_t$  can be calculated as follows:

$$f_t = \delta(W_{\text{uf}}[h_{t-1}, x_t] + b_{\text{uf}}), \quad (10)$$

where  $W_{\text{uf}}$  is the weight between the cell state and forget gate and  $b_{\text{uf}}$  is the additive bias of the forget gate.

The second step determines which input to choose. This step has two substeps, as shown in equations (11) and (12).

$$i_t = \delta(W_{\text{ui}}[h_{t-1}, x_t] + b_{\text{ui}}), \quad (11)$$

$$\hat{c}_t = \tan h(W_{\text{ug}}[h_{t-1}, x_t] + b_{\text{ug}}), \quad (12)$$

where  $W_{\text{ui}}$  and  $W_{\text{ug}}$  are the weights between the cell state and the input and external output gates, respectively. Moreover,  $b_{\text{ui}}$  and  $b_{\text{ug}}$  are the additive biases of the input gate and external output gate, respectively.

Next, the LSTM model updates  $c_t$  according to the outputs of these two gates with equation (13).

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t. \quad (13)$$

These changes are applied to  $h_t$ , and the hidden state is updated as shown in equations (14) and (15).

$$o_t = \delta(W_{\text{uo}}[h_t, x_t] + b_{\text{uo}}), \quad (14)$$

$$h_t = o_t * \tan h(c_t), \quad (15)$$

where  $W_{\text{uo}}$  is the weight between the cell state and the output gate and  $b_{\text{uo}}$  is the additive bias of the output gate. In addition, the sigmoid function  $\delta(\cdot)$  and hyperbolic activation function  $\tan h(\cdot)$  are used as activation functions.

Finally, to identify the walking or biking mode, we input the feature  $h_t$ , which is extracted in the last LSTM cell into a single perceptron layer. The output  $h_\theta$  of the model is calculated as follows:

$$h_\theta = \delta(W_\theta h_t + b_\theta), \quad (16)$$

where  $W_\theta$  is a weight matrix that transfers the values in the fully connected (FC) layer to the output layer and  $b_\theta$  is a bias factor. In equation (16), the sigmoid function  $\delta(\cdot)$  is used to transform the logit of a single neuron in the final stage to calculate the probability of classifying walking or biking.

The packet captured by each detector is split into windows after processing. If there are  $p$  frames in a window, the inputs are passed through the LSTM  $p$  times. As previously explained, in this study, each frame in the window has three features. Following feature extraction, the feature values are normalized in the range (0, 1). Finally, all features within a window are input into the LSTM model.

## 5. System Evaluation

**5.1. Hardware Platform.** The hardware used for data acquisition in this experiment is shown in Figure 5. In this figure, 1 is the Wi-Fi detector charger, which converts from



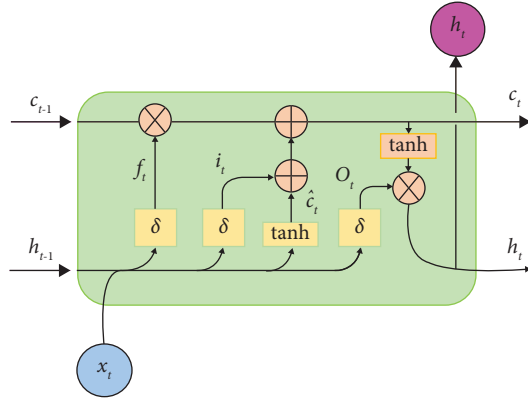


FIGURE 4: Basic LSTM cell structure.

220 V AC to 12 V DC; 2 is the DS-007 detector, which was produced by Chengdu DataSky Company, China, and has been proven to be suitable for use in outdoor environments; 3 is the NewsMY-W12 battery source; 4 is the cable connecting the battery source and Wi-Fi detector; 5 is a tape measure used to measure the straight-line distance between the target and the Wi-Fi detector; 6 is a LAN cable that allows the collected data to be transmitted to a personal computer in real-time; and 7 is a laptop.

For short-term field experiments, the Wi-Fi detector can be powered by a mobile power supply, while for long-term field experiments, an external AC power cable should be connected to the detector. Before the formal experiment, the DS-007 detector was pretested to evaluate its performance, such as its scan interval, signal fading rate, directional inhomogeneity, detection rate, and packet loss rate.

During the preexperiment, we tested different numbers and different types of small samples of travelers by calculating the values of  $B$  and  $\lambda$  by collecting the RSSI values between the target and the wireless detector at different distances. We moved the smart device from 1 m to 15 m in steps of 1 m and acquired 25 RSSI values at each fixed point. Then, the outliers at each position were discarded by computing the mean and variance of the measurements. In detail, we removed data samples that were greater than one standard deviation from the mean. Subsequently, we performed a logarithmic interpolation of the RSSI data according to equation (8). The resulting fitting curve is shown in Figure 6. The calibrated values of  $B$  and  $\lambda$  were set to  $B = 49.51$  and  $\lambda = 1.2$ . For the DS-007 detector, a calibrated PM was used in this study. Finally, the RSSI value was converted to a distance with equation (17).

$$d_{k,j}^{\text{eff}} = 10^{\left(49.51 - S_{k,j}^{\text{eff}}/12\right)}. \quad (17)$$

In addition, according to the pre-experiment test results, the effective detection area of the DS-007 Wi-Fi detector can be approximated as a sphere with a radius of 30 m, which is slightly smaller than the 50 m radius of the Wi-Fi detector used by Lesani and Miranda-Moreno [22]. The effective detection range of the detector depends on the chosen antenna. Antennas with low gains have lower detection rates but more accurate velocity estimations. Therefore, in this



FIGURE 5: Hardware equipment for data acquisition.

experiment, when the linear distance between the intelligent terminal device and the DS-007 Wi-Fi detector was less than 30 m, all connection details were collected by the detector.

**5.2. Data Acquisition.** To validate the proposed Wi-CL system, four DS-007 Wi-Fi detectors were deployed in a specific area (namely, a circle consisting of four streets) at the SCUT, Wushan Campus, to collect Wi-Fi trajectories as participants walked or cycled. It is important to note that unlike traditional intrusive traffic sensors such as toroidal induction coils, piezoelectric sensors, and magneto-resistive sensors, the deployment of the Wi-Fi detectors proposed in this paper utilizes existing traffic support facilities such as traffic signal frames or intersection light poles and does not require additional deployment of road structures. Therefore, the deployment of the experimental detectors does not affect the road traffic environment and normal traffic operations. It should be pointed out that the equipment proposed in this paper is easily affected by the multipath effect when conducting data acquisition outdoors. The multipath effect means that the electromagnetic wave propagates through different paths and the component fields reach the receiving end at different times according to their respective phases, causing interference and distortion or error of the original signal. The multipath effect will lead to signal fading and phase shift. Therefore, before sending the collected data into the pattern classification system, the data processing and filtering module is required to filter the RSSI signal.

This campus has a large-scale pedestrian and cyclist network with reduced vehicular traffic. Therefore, this is a suitable place to test the proposed system. In this study, we recruited four volunteers from the Intelligent Transportation Laboratory of the SCUT for data collection. In the experiment, in order to control the rationality of the experiment, the participants were all school students, aged about 20–28 years old, with a male to female ratio of about 2 : 1. The gait speed was normal human speed. We conducted 20 replicated experiments at each testing location to reduce random errors. The participants were encouraged to carry a Wi-Fi-enabled smartphone and move on the road by walking and cycling. A total of 160 trips, including 80 walking and 80 biking trips, were collected; 70% of the trips were included in the training set to calibrate the developed model, and the remaining 30% were used to validate the performance of the classifier. Furthermore, in the Wi-Fi-based approach, our pre-experiments demonstrated that

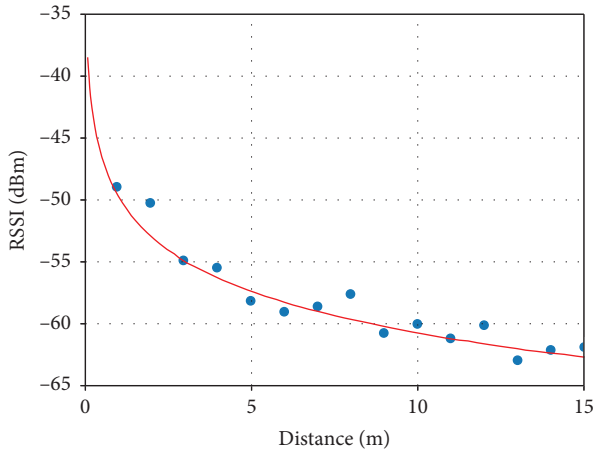


FIGURE 6: The distribution of the RSSI value and distance in the pre-experiment.

individual differences were not significant for the classification of travel mode (motor vehicle/bicycle/walking). Therefore, this paper does not focus on individual differences.

The trace data in this experiment is labeled data; for example, the MAC Address is unique, so there will be no confusion during detection. There are also filtering modules, LSTM modules, etc. for auxiliary identification, and they all have good signal noise reduction and feature extraction performance.

The experimental measurement data used in the analysis were collected during two separate periods: (1) 10:00 to 11:00 on July 10, 2019, namely, the flat peak, and (2) 12:00 to 13:00 on July 10, 2019, namely, the noon peak. Two data collection periods were included to compare the impact of crowding on the classification performance of the system. The locations of the DS-007 Wi-Fi detectors and the participants' trajectories are shown in Figure 7. Mutual influence between the detectors was ignored only when the distance between two detectors was considerably greater than the coverage area of the detectors. On-site, the locations of the detectors were carefully determined to ensure that overlap did not occur between the sensing ranges of different detectors. The shortest distance between the detectors was approximately 200 m, which is greater than the detection radius of 30 m.

**5.3. Performance Analysis of the Proposed System.** In this subsection, we evaluate the performance of the proposed system based on the collected dataset. The evaluation has three key objectives: (1) to assess the noise reduction performance of the system; (2) to evaluate the speed estimation performance of the system; and (3) to compare the proposed classification framework with traditional classification algorithms.

**5.3.1. Noise Reduction Performance Analysis.** The high volatility of RSSI signals causes system errors; thus, the tolerance to RSSI signal fluctuations is an important performance metric. In the literature [20], we demonstrated

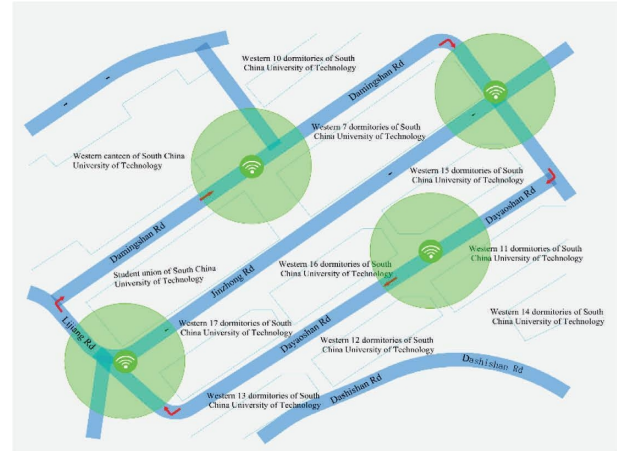


FIGURE 7: Experiment area and Wi-Fi detector locations.

that the CVKF filter can effectively suppress the localization divergence caused by RSSI signal fluctuations, regardless of whether pedestrians are stationary or moving at low speeds. However, we do not know whether the CVKF filter is equally applicable to bicycles, which travel at faster speeds than walking pedestrians. Therefore, in this study, we first analyze the noise reduction performance of the system. In the experiment, the RSSI signal was received by the smart terminal device in three scenarios: (1) the tester remained stationary for 80 seconds at a distance of 10 m from the detector; (2) the tester started at a distance of 30 m from the detector, approached the detector, and walked 30 m away from the detector; and (3) the tester started at the detector, moved 30 m away from the detector, rode a bicycle to the detector, and rode their bicycle 30 m away from the detector. While walking or riding, the tester maintained a constant speed as much as possible. The ground truth was generated at several reference points with measured positions. A stopwatch was used to record the time taken by the tester to pass these reference points and was interpolated to obtain the true position of the ground between these reference points [56]. In addition, we assume that the pedestrian moves at a constant speed between the two reference points.

In fact, since the detection cycle of a single Wi-Fi detector is short. Therefore, it can be assumed that the activity during the detection cycle is a single activity, i.e., the user is either riding a bike or walking. For a case corner task such as someone in the middle of a bike ride, outliers can be handled by the action of the filtering module. Moreover, using the distinction between stationary and active states is relatively simple for the RSSI-based approach, as shown in Figure 8.

Figure 8 compares the raw RSSI data and the filtered RSSI data collected in the three scenarios. The RSSI filtering algorithms use CVF, KF, and CVKF filters. As shown in Figure 8, in schemes (1) to (3), the raw RSSI data collected by the detector are relatively noisy. Thus, if these data are used directly for mode classification, large errors may occur. For example, as shown in Figure 8(a), the raw RSSI data fluctuate quickly between 0 dBm and 10 dBm even when the data are collected at the same fixed location. The raw RSSI data are processed using a filtering algorithm to obtain smoothed

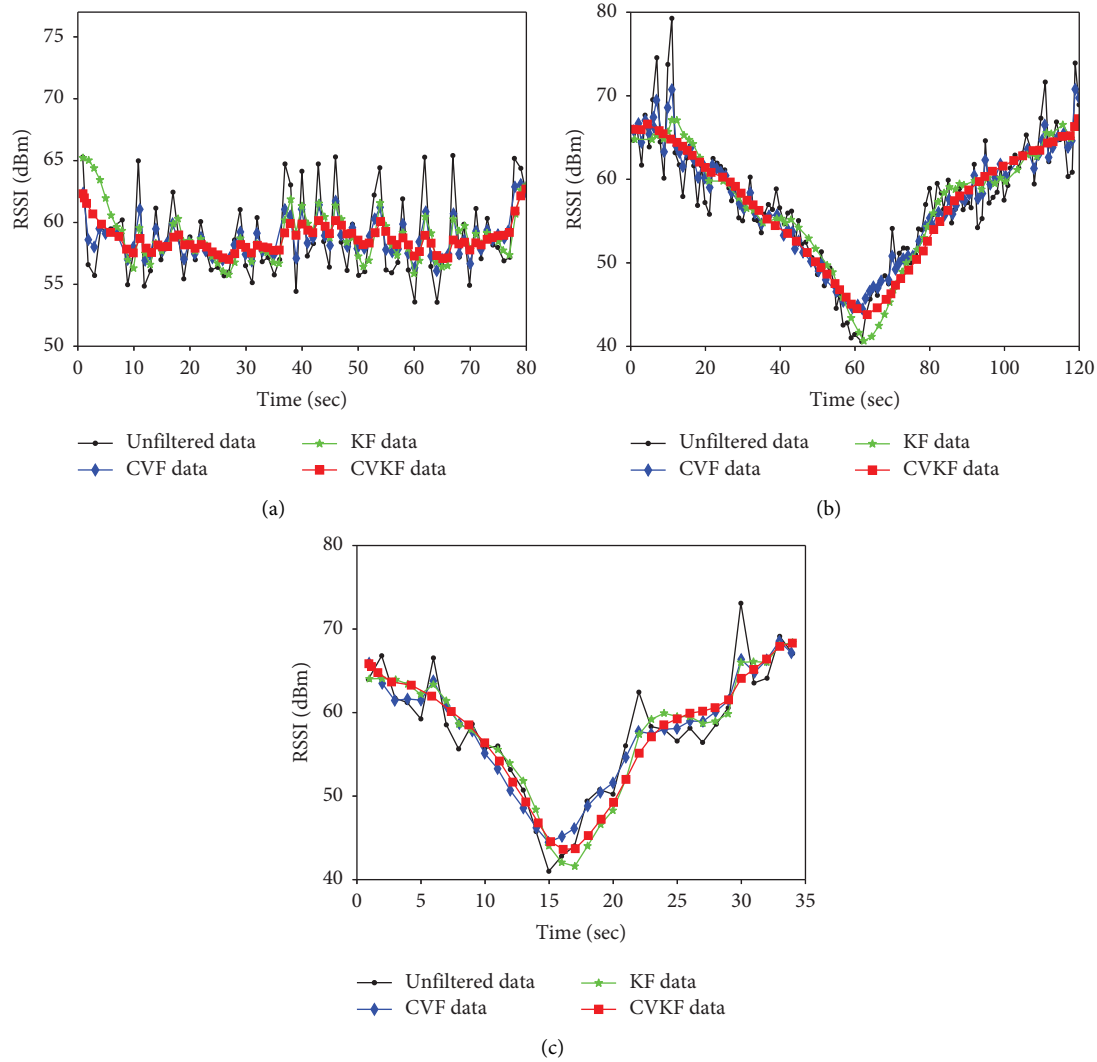


FIGURE 8: Comparison of the filtering performance in the three scenarios. (a) Stationary. (b) Walking. (c) Biking.

data. In Schemes (1) to (3), the CVF, KF, and CVKF algorithms enhance data with less errors more than unfiltered data. However, significant error peaks in the first few RSSI data sequences have a considerable impact on the optimization results of the KF algorithm. In Figure 8(a), the first few raw RSSI values have large errors at 0–10 s, leading to large errors in the KF algorithm (the filtering performance improves only after approximately 10 s). Unfortunately, in a real traffic environment, the possibility of peak errors in the first few data sequences cannot be eliminated. Moreover, the CVF algorithm may overfit data with peaks, as shown in Figure 8(b) between 50 and 60 s and Figure 8(c) between 12 and 17 s. This result likely occurs because the prediction principle of the CVF algorithm is based on a fixed speed, which is not sensitive enough to the actual situation of the RSSI peaks. It is worth noting that the CVKF algorithm proposed in this paper addresses the above two problems.

In addition, this study evaluates the effectiveness of the proposed filter by converting the RSSI into a distance value. Two evaluation metrics are considered: the mean error and

the root mean square error. The estimated distance errors of the CVF, KF, and CVKF algorithms for the three cases are shown in Table 2.

The distance estimates obtained from the original unfiltered RSSI data, including stationary, walking, and biking data, are subject to large errors. Regardless of whether the CVF, KF, or CVKF algorithm is used, the mean error and root mean square error (RMSE) are larger in the sports environment than in the stationary environment. This may be related to the fact that people move faster in walking and cycling environments, resulting in larger signal fluctuations. On the one hand, for the walking scenario, the average error of the CVKF algorithm is 5.74 m, which is 27.78% (7.94 m) and 7.57% (6.21 m) less than the average errors of the CVF and KF algorithms, respectively. On the other hand, for the cycling scenario, the average error of the CVKF algorithm is 5.53 m, which is approximately 19.74% (6.89 m) and 18.68% (6.80 m) less than the average errors of the CVF and KF algorithms, respectively. Therefore, the filtering performance of the CVKF algorithm is better than that of the CVF

TABLE 2: Comparison of the estimated distance error in the three scenarios (unit: m).

Scenario		Method			
		Unfiltered	CVF	KF	CVKF
Stationary	Average error	5.89	3.62	4.07	3.46
	RMSE	3.91	1.51	2.68	1.21
Walking	Average error	15.01	7.94	6.21	5.74
	RMSE	45.04	7.68	3.43	2.90
Cycling	Average error	11.31	6.89	6.80	5.53
	RMSE	18.02	4.63	4.35	4.45

and KF algorithms in these three cases. Moreover, the average error of the KF algorithm in the cycling environment (6.80 m) is larger than that in the walking environment (6.21 m), indicating that the KF algorithm cannot adapt to changes in the cycling environment. In contrast, the CVKF algorithm maintains a better filtering performance, even in the faster cycling environment.

*5.3.2. Speed Estimation Performance Analysis.* Travel speed is the main feature of existing traffic travel mode classification models based on data collected by smart electronic devices. However, most studies do not validate the estimated travel speeds. In addition to analyzing the noise reduction performance of the system, in this study, we verify the accuracy of the estimated travel speed extracted from the collected MAC addresses and RSSI signals. To verify the accuracy of the estimated travel speeds, the locations of the testers during movement were generated using the specialized open-source software Sensorlog, which is a mobile data collection and annotation application that is effective for collecting mobile location data, as explained in [57]. This application can be downloaded from the Google Play Marketplace. In this study, we used the velocity data of the testers collected by this application as ground-truth data. In this experiment, we compared the ATS, RTS, and RFTS data collected in four scenarios. The four scenarios are defined as follows: (a) walking data collected during the flat peak; (b) walking data collected during the noon peak; (c) cycling data collected during the flat peak; and (d) cycling data collected during the noon peak. Figure 9 shows the vehicle speeds estimated by the three methods in the four scenarios, and Figure 10 shows the cumulative distribution functions (CDFs) of the speed estimation errors for the ATS, RTS, and RFTS algorithms.

As shown in Figure 9, the ground truth speeds for walking and bicycling are relatively stable during the flat peaks (scenarios (a) and (c)). In contrast, during the midday peak hours, the ground truth speeds for walking and cycling decrease sharply. For example, the cycling speed decreases substantially from 1.5 to 2.2 m/s to 0.3–0.8 m/s within 20–30 s in scenario (d). This decrease may be due to the increase in pedestrian network traffic on Damien Hill Road during the noon peak period due to students leaving school. As a result, bicyclists had to reduce their speed when they reached that road section. The ATS algorithm uses the average moving speed of the moving target in the coverage area, which is influenced considerably by the first and last

RSSI signal values. If large errors occur, the accuracy of the ATS algorithm decreases rapidly. In addition, this algorithm does not accurately reflect changes in the velocity of the moving target during the monitoring period. For example, in scenarios (b) and (d), the ATS algorithm maintains the original value even after the speed of the moving target changes. In addition, when the first few values in the RSSI sequence have large errors, the speed estimation result of the RFTS algorithm is closer to the ground truth speed than that of the RTS algorithm. For example, in scenario (d), the RSSI values are closer between 0 and 5 s, and the RTS algorithm estimates a velocity value of 0 m/s. However, the RFTS algorithm calculates the average of the estimated velocities within a window, and the final smoothed input is 0.5 m/s.

Table 3 shows the vehicle speed estimation errors for the three methods in the four scenarios. According to Figure 10 and Table 3, the average error of the RFTS algorithm in the four scenarios (1.9 m/s, 0.19 m/s, 0.57 m/s, and 0.42 m/s) is smaller than those of the ATS (0.24 m/s, 0.24 m/s, 0.82 m/s, and 0.60 m/s) and RTS algorithms (0.22 m/s, 0.23 m/s, 0.80 m/s, and 0.58 m/s). Moreover, the 90% CDF errors of the RFTS algorithm in the four cases are 0.38 m/s, 0.39 m/s, 1.25 m/s, and 1.0 m/s, which are smaller than those of the ATS and RTS algorithms. Inevitably, the estimated speed errors obtained by the three algorithms are larger in the cycling environment than in the walking environment. However, it is worth noting that the proposed RFTS algorithm has the smallest error among the three algorithms. In other words, compared with the other algorithms discussed in this paper, the RFTS algorithm is more stable in the four cases, and the estimated travel speed is closer to the ground truth speed.

*5.3.3. Classification Accuracy Performance Analysis.* We compare the proposed classification framework with LR [58], SVM [59], and MLP [5] three machine learning algorithms that are widely used in classification models. The parameters in the LR, SVM, and MLP models are well tuned to achieve good accuracy. The SVM classifier uses a linear kernel function with a soft edge constant of 1. The MLP classifier uses the following parameters: number of epochs: 200; optimization method: Adam; number of hidden layers: 2; input and hidden layer activation function: ReLU; all hidden layer activation: 4; output layer activation function: sigmoid; and batch size: 20. In addition, the classification framework designed in this paper consists of an LSTM layer and a fully connected layer. The sigmoid activation function

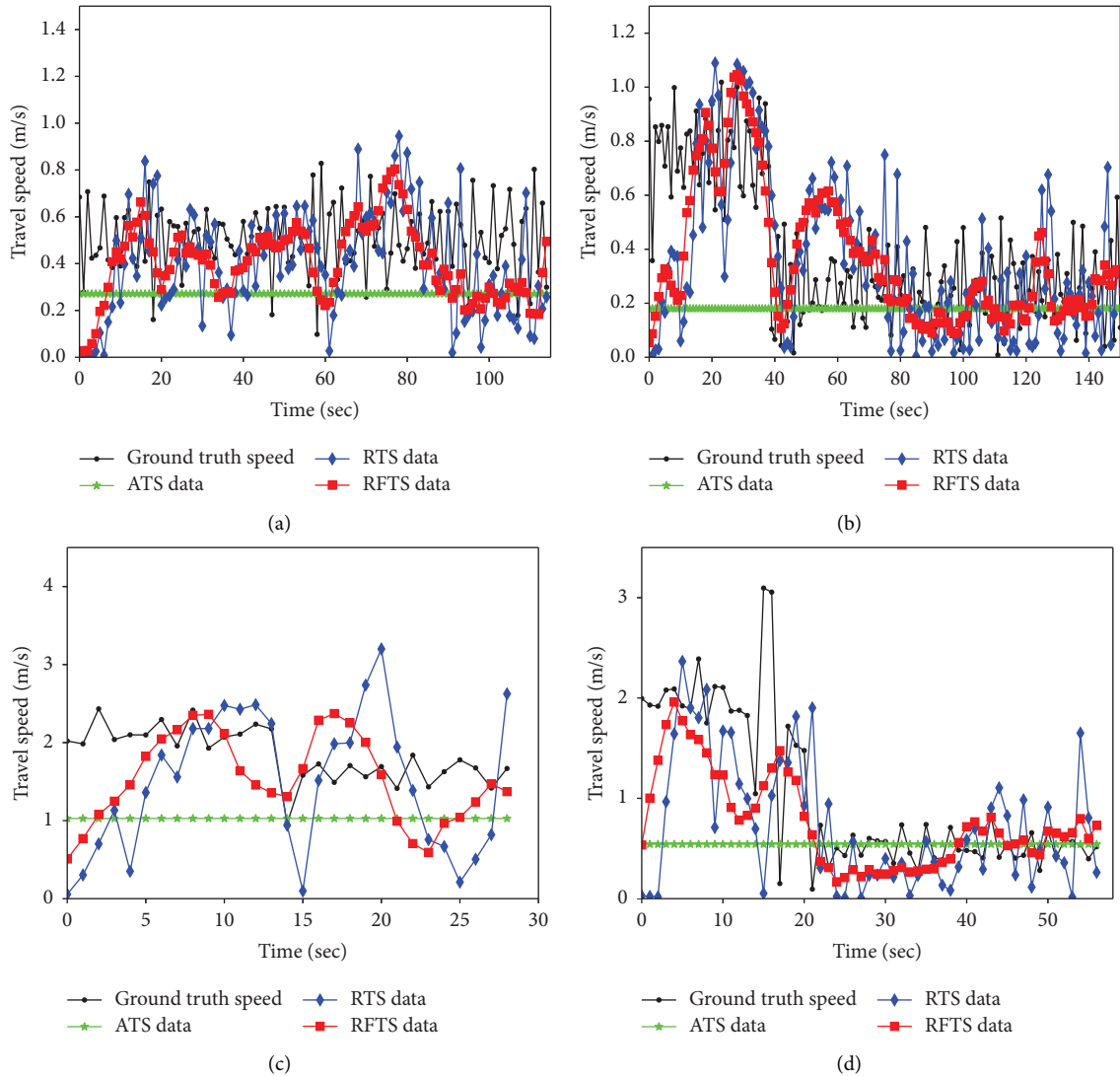


FIGURE 9: Distribution of estimated travel speed determined by three methods in four scenarios.

was used in the output layer, and the cross-entropy loss function and Adam optimizer were applied. The ReLU activation function was used between the outputs of the LSTM layer and the fully connected layer. In addition, the output size of the LSTM layer was 128, and the size of the fully connected layer was 1. When the length of the data in the batch was inconsistent, we padded the data with 0s in front. For a fair comparison, we perform the same data processing and feature selection methods for the RSSI signals. For each classifier, we perform 10 cross-validations on the collected dataset [46]. For a detailed analysis of the results, the results of each algorithm are shown in Table 4.

The experiments were conducted on a Linux system on a Lenovo G40 computer with an Intel(R) Core (TM) i5-4258U CPU @2.40 GHz, Python version 2.7.15, and TensorFlow version 1.12.0 CPU model. The classification metrics considered in our analysis are accuracy, precision, and recall. We determined the values of these metrics for each travel mode class and reported the average of each class value for each classifier. The accuracy is defined as the

number of blocks that were correctly classified as belonging (true positive) or not belonging (true negative) to a class divided by the total number of inferences (overall). The precision is obtained by dividing the number of correctly classified blocks by the total number of inferences made for that class (true positives + false positives). The recall is calculated by dividing the number of correctly classified blocks by the total number of blocks that belong to that class (true positives + false negatives) [7].

In Table 4, in the analysis of the algorithm prediction results, the header columns are the actual labels, and the header rows are the predicted labels. These data show the error ratios for different error attributes. To further analyze the LSTM model designed in this paper, the accuracy and training loss are shown in Figure 11. We can summarize some interesting findings as follows:

- (i) The classification process starts with the training of the LR model. Although the calibration process is simple, the 72.92% accuracy of the LR model is not

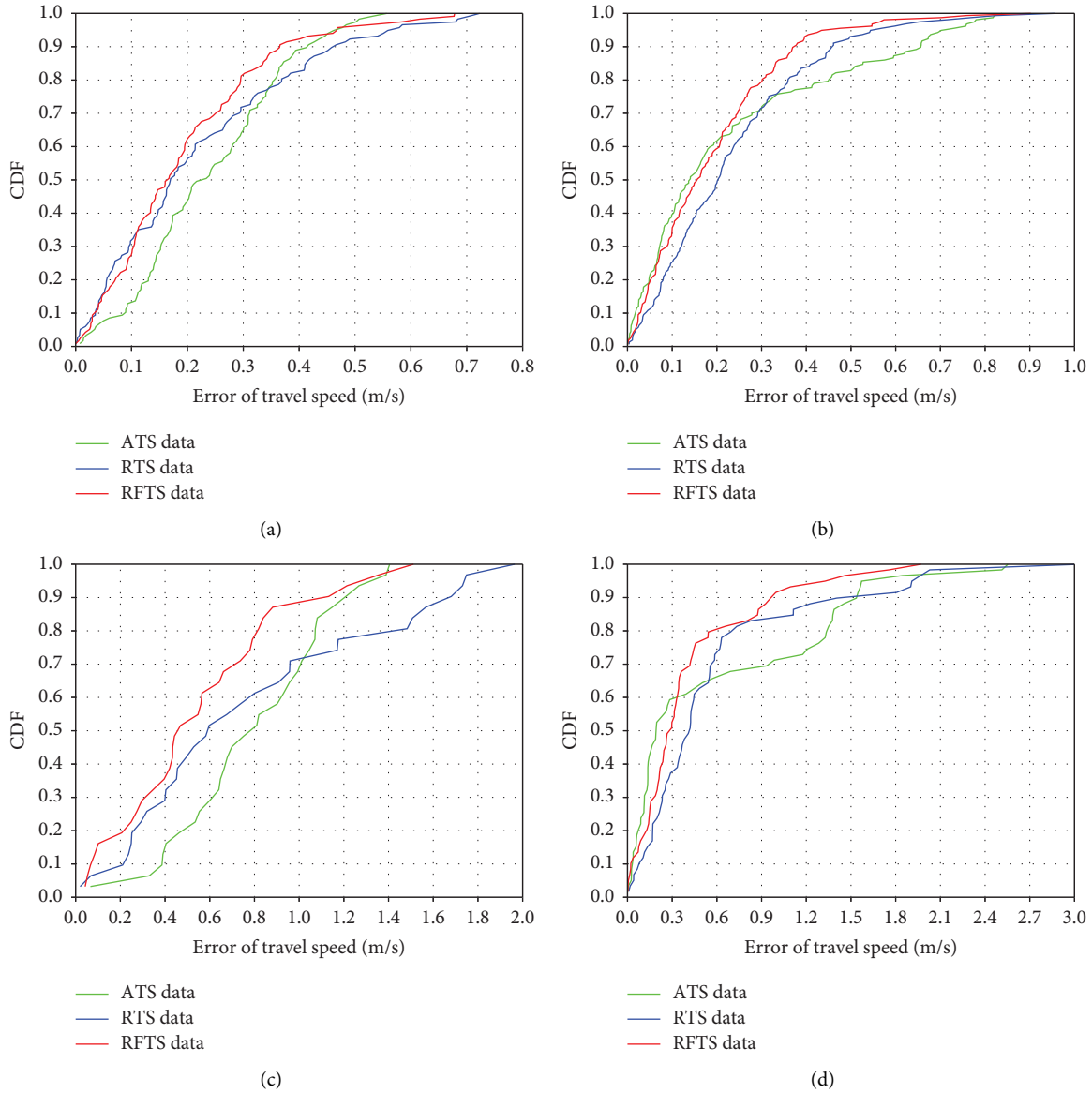


FIGURE 10: CDFs of the estimated travel speed determined by three methods in four scenarios.

TABLE 3: Comparison of the travel speed estimation errors of the three methods (unit: m/s).

Scenario		Method		
		ATS	RTS	RFTS
Scenario (a)	Average error	0.24	0.22	1.94
	RMSE	0.13	0.18	0.15
Scenario (b)	Average error	0.24	0.23	0.19
	RMSE	0.23	0.18	0.16
Scenario (c)	Average error	0.82	0.80	0.57
	RMSE	0.33	0.58	0.40
Scenario (d)	Average error	0.60	0.58	0.42
	RMSE	0.69	0.62	0.44

satisfactory and is the lowest among the four models. Compared with the LR model, the accuracy of the SVM model is improved, reaching a value of 79.14%, which is still below 80%. The MLP model

for predicting the moving modes achieved better recall and accuracy scores than the first two models; however, the results were still unsatisfactory. To reduce the error in the MLP model and improve the

TABLE 4: Analysis of the algorithm prediction results.

Model	Actual modes	Predicted modes		Recall	Accuracy (%)
		Walking	Biking		
LR	Walking	21	3	87.50%	72.92
	Biking	10	14	58.33%	
	Precision	65.63%	82.35%	—	
SVM	Walking	24	0	100.00%	79.14
	Biking	10	14	58.33%	
	Precision	70.59%	100%	—	
MLP	Walking	24	0	100.00%	89.58
	Biking	5	19	79.17%	
	Precision	82.76%	100%	—	
LSTM	Walking	24	0	100.00%	97.92
	Biking	1	23	95.83%	
	Precision	96.00%	100%	—	

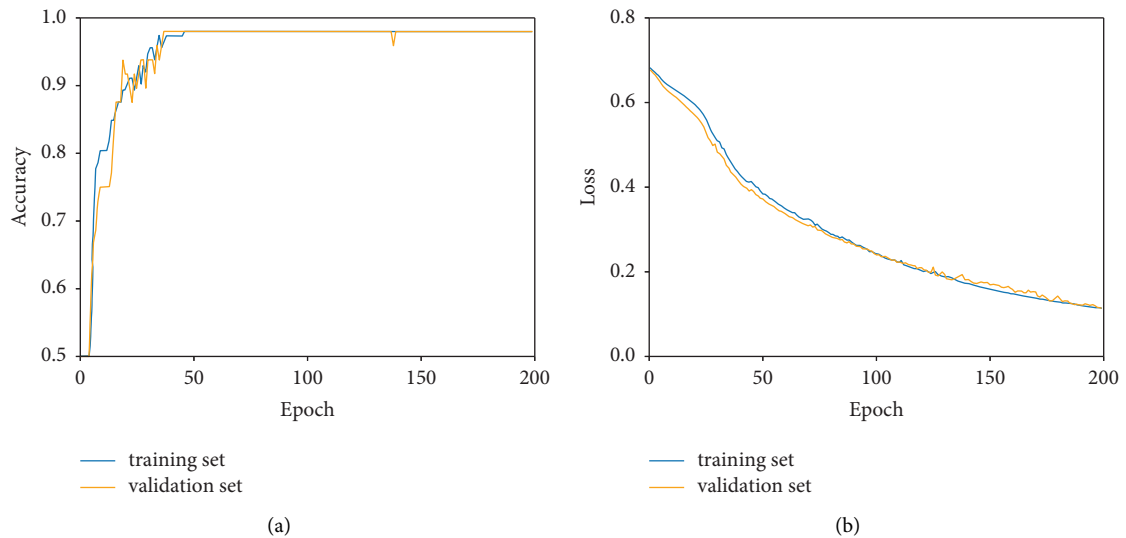


FIGURE 11: Performance of training and validation data versus the number of epochs. (a) Accuracy. (b) Loss.

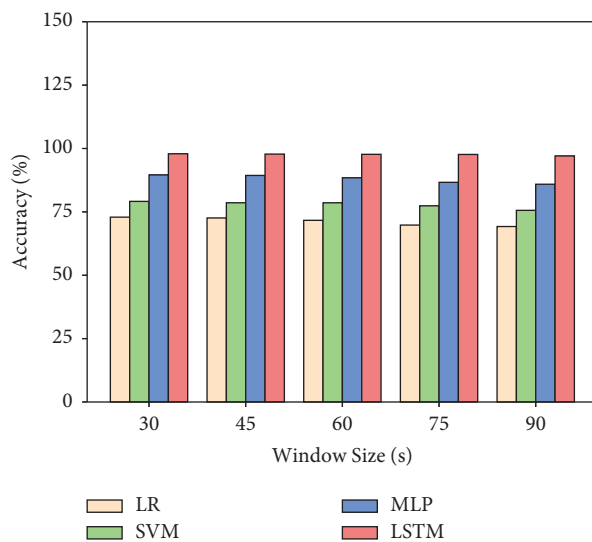


FIGURE 12: Accuracy of LR, SVM, MLP, and LSTM model models in different time windows.

classification prediction accuracy, the LSTM algorithm was experimentally implemented. The overall prediction accuracy of the LSTM model was 97.92%, and the check-all rate and accuracy of the labeled observations were greater than 90%, which were higher than the respective values of the first three models. The results show that for the classification of walking traffic modes, the LSTM model exhibits the best classification performance among the four models.

- (ii) Cycling is the transportation mode with the lowest recall rate. The recall rates of the LR, SVM, MLP, and LSTM models were 59.33%, 58.33%, 79.17%, and 95.83%, respectively. A large number of observed cycling trips were classified as walking trips. This error may reflect the fact that in crowded spaces, cycling and walking share many speed-related features, which increases the difficulty of distinguishing the two as different modes. In this case, the LSTM model exhibits the most stable classification performance among the four models. Out of 24 bicycle observations, 23 observations were correctly predicted, and only one observation was incorrectly predicted as walking, yielding a recall rate of 95.83%. This recall result is more accurate than the results of the LR, SVM, and MLP models.
- (iii) Figure 11 shows that the accuracy of the LSTM model reaches 95% in the first 50 epochs during training, indicating that the LSTM model can effectively classify nonmotorized traffic modes. In addition, the loss of the LSTM model decreases significantly in the first 100 epochs. Between the 150<sup>th</sup> epoch and the 200<sup>th</sup> epoch, the loss does not change substantially, as shown in Figure 11(b). The results show that the model converges to the optimal solution by the 200<sup>th</sup> training epoch.
- (iv) As shown in Figure 12, the accuracy of the four models of LR, SVM, MLP, and LSTM improves sequentially in different time windows. In general, the accuracy of the four types of models slightly decreases as the time window gets larger, but LSTM maintains high performance and high accuracy. The accuracy stays above 95% in all five-time windows tested, which is the best among the four types of models proposed.

## 6. Conclusion

This study considers nonmotorized travel mode classification and proposes a nonmotorized travel mode classification system using only a single Wi-Fi detector as a data source based on existing research. The proposed system achieves fine-grained identification of different traffic travel modes with a low deployment cost, good real-time performance, and satisfactory recognition accuracy. In contrast to other related studies, this study does not combine Wi-Fi detection data with other data sources to explore the travel patterns of traffic participants; thus, our approach is more cost-effective

and easier to implement in practice. More desirable results were achieved in terms of processing data anomalies and effectively reducing signal noise. The proposed RFTS algorithm has the smallest speed estimation error among four comparison algorithms, and the results are closer to the real movement speeds of traffic participants. Moreover, the proposed algorithm achieves good results in terms of travel mode classification accuracy, which is our greatest concern, and has the best results among the four algorithms in terms of both classification accuracy and recall recognition rate.

Following this research, we can also collect more valuable trajectory-type data for data mining, such as origin-destination backpropagation, urban traffic state estimation, traffic trip characterization, and traffic safety assessment. Moreover, we can better understand the basic relationship among traffic flow velocity, traffic flow density, and traffic volume. The above research results can be used as an alternative model in future traffic information monitoring systems in smart cities.

In future work, this study can be improved in several ways, and future research will address the following three issues: (1) validating the effectiveness and reliability of the Wi-CL system for various road geometries and traffic demand modes and conducting more extensive field experiments with the system; (2) improving the filtering algorithms and classification methods; and (3) designing a broader range of urban road network applications that cover a wide range of traffic modes, such as small cars, pedestrians, bicycles, subways, surface buses, and light rails.

## Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported in part by the Guangdong Provincial Universities Key Field Special Project (Grant no. 2021ZDZX1077); Research Ability Enhancement Project of Key Construction Disciplines of Guangdong Province (Grant no. 2021ZDJS116); and Jiangxi Province Graduate Student Innovation Special Fund Project (Grant no. YC2021-S587); Zilin Huang and Sikai Chen did not receive any funding for this work.

## References

- [1] L. Laghari and A. Asif, "A review and state of art of Internet of Things (IoT)," *Archives of Computational Methods in Engineering*, vol. 8, pp. 1–19, 2021.
- [2] X. Tian, Y. Cheng, D. M. Shila, and A. Wolisz, "Guest editorial special issue on enabling a smart city: Internet of Things meets AI," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7469–7472, 2019.



- [3] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: applications, challenges, and opportunities with China Perspective," *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 349–359, 2014.
- [4] M. Handte, S. Foell, S. Wagner, G. Kortuem, and P. J. Marron, "An internet-of-things enabled connected navigation system for urban bus riders," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 735–744, 2016.
- [5] A. Kalatian and B. Farooq, "Mobility mode detection using WiFi signals," in *Proceedings of the 2019 IEEE International Smart Cities Conference (ISC2 2019)*, pp. 1–7, Kansas City, MO, USA, September 2019.
- [6] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1202–1212, 2017.
- [7] E. F. de S. Soares, K. Revoredo, F. Baiao, C. A. de M. S. Quintella, and C. A. V. Campos, "A combined solution for real-time travel mode detection and trip purpose prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4655–4664, 2019.
- [8] R. Jia, P. Jiang, L. Liu, L. Cui, and Y. Shi, "Data driven congestion prediction of urban transportation," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 581–591, 2018.
- [9] L. Xu, Y. Pian, Y. Lin, and Z. Huang, "Game analysis of conflicts between pedestrians and vehicular traffic on unsignalized road sections based on cumulative prospect theory," *China Journal of Highway and Transport*, vol. 35, no. 1, p. 119, 2022.
- [10] N. Abedi, A. Bhaskar, E. Chung, and M. Miska, "Assessment of antenna characteristic effects on pedestrian and cyclists travel-time estimation based on Bluetooth and WiFi MAC addresses," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 124–141, 2015.
- [11] P. B. Ohlms, L. E. Dougald, and H. E. MacKnight, "Bicycle and pedestrian count programs: scan of current U.S. Practice," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 3, pp. 74–85, 2019.
- [12] J. Li, C. Shao, W. Xu, and J. Li, "Real-time system for tracking and classification of pedestrians and bicycles," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2198, no. 1, pp. 83–92, 2010.
- [13] Z. Huang, L. Xu, and Y. Lin, "Multi-stage pedestrian positioning using filtered WiFi scanner data in an urban road environment," *Sensors*, vol. 20, no. 11, p. 3259, 2020.
- [14] A. Kurkcu and K. Ozbay, "Estimating pedestrian densities, wait times, and flows with wi-fi and bluetooth sensors," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2644, no. 1, pp. 72–82, 2017.
- [15] G. Asci and M. A. Guvensan, "A novel input set for LSTM-based transport mode detection," in *Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 107–112, Kyoto, Japan, March 2019.
- [16] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, vol. 49, pp. 312–321, Seoul, South Korea, September 2008.
- [17] T. Sohn, "Mobility detection using everyday GSM traces," *Lecture Notes in Computer Science*, vol. 4206, pp. 212–224, 2006.
- [18] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 1–27, 2010.
- [19] Z. Huang, L. Xu, Y. Lin, P. Wu, and B. Feng, "Citywide metro-to-bus transfer behavior identification based on combined data from smart cards and GPS," *Applied Sciences*, vol. 9, no. 17, p. 3597, 2019.
- [20] Z. Huang, X. Zhu, Y. Lin, L. Xu, and Y. Mao, "A novel WIFI-oriented RSSI signal processing method for tracking low-speed pedestrians," in *Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS 2019)*, pp. 1018–1023, Liverpool, UK, July 2019.
- [21] T. M. Brennan, J. M. Ernst, C. M. Day, D. M. Bullock, J. V. Krogmeier, and M. Martchouk, "Influence of vertical sensor placement on data collection efficiency from bluetooth MAC address collection devices," *Journal of Transportation Engineering*, vol. 136, no. 12, pp. 1104–1109, 2010.
- [22] A. Lesani and L. Miranda-Moreno, "Development and testing of a real-time WiFi-bluetooth system for pedestrian network monitoring, classification, and data extrapolation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1484–1496, 2019.
- [23] Y. Zhu, C. Markos, and J. Q. James, "Improving transportation mode identification with limited GPS trajectories," in *Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, Washington, DC, USA, November 2021.
- [24] 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, 2017.
- [25] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proceedings of the GIS '03: 11th ACM international symposium*, pp. 54–63, Arlington, Virginia, November 2011.
- [26] P. Wu, L. Xu, J. Li, H. Guo, and Z. Huang, "Recognizing real-time transfer patterns between metro and bus systems based on spatial-temporal constraints," *Journal of Transportation Engineering, Part A: Systems*, vol. 148, no. 9, Article ID 04022065, 2022.
- [27] S. Yang and Y. J. Wu, "Travel mode identification using bluetooth technology," *Journal of Intelligent Transportation Systems*, vol. 22, no. 5, pp. 407–421, 2018.
- [28] H. Zou, Y. Zhou, R. Arghandeh, and C. J. Spanos, "Multiple kernel semi-representation learning with its application to device-free human activity recognition," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7670–7680, 2019.
- [29] W. Xi, "Electronic frog eye: counting crowd using WiFi," in *Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, pp. 361–369, Toronto, Canada, April 2014.
- [30] L. Li, X. Guo, N. Ansari, and H. Li, "A hybrid fingerprint quality evaluation model for WiFi localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9829–9840, 2019.
- [31] T. H. Vu, L. Dung, and J. C. Wang, "Transportation mode detection on mobile devices using recurrent nets," in *Proceedings of the 2016 ACM workshop on Vision and Language Integration Meets Multimedia Fusion*, pp. 392–396, Amsterdam, The Netherlands, October 2016.
- [32] P. Wu, Z. Huang, Y. Pian, L. Xu, J. Li, and K. Chen, "A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting," *Journal of Advanced Transportation*, vol. 2020, Article ID 8863724, 15 pages, 2020.

- [33] Ieee Computer Society Lan Man Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *ANSI/IEEE Std.* vol. 802, pp. 11–1999, 1999.
- [34] J. Li, P. Wu, R. Li et al., "ST-CRMF: compensated residual matrix factorization with spatial-temporal regularization for graph-based time series forecasting," *Sensors*, vol. 22, no. 15, p. 5877, 2022.
- [35] M. Cunche, "I know your MAC address: targeted tracking of individual using Wi-Fi," *Journal of Computer Virology and Hacking Techniques*, vol. 10, no. 4, pp. 219–227, 2014.
- [36] P. Wu, L. Xu, and Z. Huang, "Imputation methods used in missing traffic data: a literature review," in *International Symposium on Intelligence Computation And Applications*, pp. 662–677, Springer, 2019.
- [37] G. Lan, W. Xu, D. Ma, S. Khalifa, M. Hassan, and W. Hu, "EnTrans: leveraging kinetic energy harvesting signal for transportation mode detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1–12, 2019.
- [38] B. Assemi, H. Safi, M. Mesbah, and L. Ferreira, "Developing and validating a statistical model for travel mode identification on smartphones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1920–1931, 2016.
- [39] Y. Lin, Z. Huang, P. Wu, and L. Xu, "RSSI positioning method of vehicles in tunnels based on semi-supervised extreme learning machine," *Journal of Traffic and Transportation Engineering*, vol. 21, pp. 243–255, 2021.
- [40] J. Parkin and J. Rotheram, "Design speeds and acceleration characteristics of bicycle traffic for use in planning, design and appraisal," *Transport Policy*, vol. 17, no. 5, pp. 335–341, 2010.
- [41] Q. Dong and W. Dargie, "Evaluation of the reliability of RSSI for indoor localization qian dong and waltenegus dargie," in *Proceedings of the 2012 International Conference on Wireless Communications in Unusual and Confined Areas (ICWCUCA 2012)*, pp. 1–6, Clermont-Ferrand, France, August 2012.
- [42] J. Li, R. Li, Z. Huang, P. Wu, and L. Xu, "Dynamic adaptive generative adversarial networks with multi-view temporal factorizations for hybrid recovery of missing traffic data," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7677–7696, 2022.
- [43] H. Cho and S. W. Kim, "Mobile robot localization using biased chirp-spread-spectrum ranging," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2826–2835, 2010.
- [44] J. Wang, Q. Gao, Y. Yu, H. Wang, and M. Jin, "Toward robust indoor localization based on Bayesian filter using chirp-spread-spectrum ranging," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1622–1629, 2012.
- [45] H. A. Nguyen, H. Guo, and K. S. Low, "Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3619–3628, 2011.
- [46] A. Jahangiri and H. A. Rakha, "Applying machine learning techniques to transportation mode recognition using mobile phone sensor data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2406–2417, 2015.
- [47] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification," *Computers, Environment and Urban Systems*, vol. 36, no. 6, pp. 526–537, 2012.
- [48] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, "Overcoming the myopia of inductive learning algorithms with RELIEFF," *Applied Intelligence*, vol. 7, no. 1, pp. 39–55, 1997.
- [49] X. Shi, X. Zhao, Z. Ye, N. Shiwakoti, and J. Lu, "Estimating pedestrian walking characteristics by use of smartphone sensing: an experimental study," in *Proceedings of the 17th COTA Conference International Conference of Transportation Professionals (CICTP2017)*, American Society of Civil Engineers, Shanghai, China, July 2017.
- [50] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, "Testing deep neural networks," 2018, <https://arxiv.org/abs/1803.04792>.
- [51] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, <https://arxiv.org/abs/1506.00019>.
- [52] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [53] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Article ID 132306, 2020.
- [54] J. Li, L. Xu, R. Li, P. Wu, and Z. Huang, "Deep spatial-temporal bi-directional residual optimization based on tensor decomposition for traffic data imputation on urban road network," *Applied Intelligence*, vol. 13, pp. 1–19, 2022.
- [55] F. Elmaz and R. Eyckerman, "CNN-LSTM architecture for predictive indoor temperature modeling," *Building and Environment*, vol. 206, Article ID 108327, 2021.
- [56] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, pp. 596–620, 2016.
- [57] H. F. Alan, B. Arnrich, C. Ersoy, and B. Cinaz, "Sensor log: bir mobil veri toplama ve etiketleme uygulaması," in *Proceedings of the 2014 22nd Signal Processing and Communications Applications Conference (SIU)*, pp. 1375–1378, Trabzon, Turkey, April 2014.
- [58] G. H. Bham, B. S. Javvadi, and U. R. R. Manepalli, "Multinomial logistic regression model for single-vehicle and multivehicle collisions on urban U.S. Highways in Arkansas," *Journal of Transportation Engineering*, vol. 138, no. 6, pp. 786–797, 2012.
- [59] C. C. Chang and C. J. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.