

Online Public Transit Ridership Monitoring through Passive WiFi Sensing

Wenbo Chang, Baoqi Huang, *Member, IEEE*, Bing Jia, *Member, IEEE*, Wuyungerile Li, and Gang Xu, *Member, IEEE*

Abstract—Online public transit ridership information is helpful to enhance the service quality of urban public transportation and the travel experiences of passengers. Passive WiFi sensing collects WiFi probe (request) frames sent by nearby mobile devices in a non-intrusive manner, and can thus be employed to monitor ridership. Compared with the existing non-WiFi based approaches, passive WiFi sensing based approaches demonstrate the advantages of limited interferences, large coverage, low costs and lightweight calculations. More recently, although some dedicated passive WiFi sensing based methods have been proposed in an offline mode, due to sniffing opportunistically, unknown dynamic transmission boundary, MAC randomization and the difficulty in online feature extraction, how to utilize limited sensing data to provide accurate online ridership information is still challenging. To this end, an innovative public transit ridership monitoring system built upon a customized WiFi sniffer and an online ridership estimation algorithm is developed. In the algorithm, a convolutional neural network (CNN) module and a bidirectional long short-term memory (BiLSTM) neural network module are first adopted to find correlations among inputs and capture the bidirectional time-series features, respectively; furthermore, an attention module is incorporated to determine the importance of an input sequence at different times. Real-world experiments are carried out on 8 buses corresponding to 4 bus routes in Hohhot, China. The evaluation results show that the proposed algorithm outperforms the other 4 online algorithms and the state-of-the-art offline algorithm.

Index Terms—Public transit, online ridership, CNN, BiLSTM, attention mechanism

I. INTRODUCTION

NOWADAYS, WiFi network infrastructures such as access points (APs) and WiFi enabled mobile devices have penetrated our daily life. In light of the IEEE 802.11 Standard, a mobile device actively and periodically broadcasts probe (request) frames across different channels to associate with an AP or switch between different APs. Since a probe frame involves spatial-temporal information about the user carrying this mobile device, a special kind of WiFi APs, termed WiFi sniffers, can thus be leveraged to passively sense user's behaviors and activities in public places, such as campuses, shopping malls, metro stations, public transit,

(Corresponding author: Baoqi Huang)

This work is supported by the National Natural Science Foundation of China (Grant No. 62262046, 41871363, 42161070 and 62061036), the Key Science-Technology Project of Inner Mongolia Autonomous Region (Grant No. 2021GG0163), and the Natural Science Foundation of Inner Mongolia Autonomous Region of China (Grant No. 2021ZD13 and 2019MS06030).

The authors are with Engineering Research Center of Ecological Big Data, Ministry of Education, Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing, College of Computer Science, Inner Mongolia University, Hohhot 010021, China (email:cshbq@imu.edu.cn)

etc. [1] Moreover, it is feasible to develop various novel applications in a non-intrusive manner, such as monitoring pedestrian flows [2], tracking and localization [3], [4], [5], [6], [7], [8], unveiling social relationships [9], measuring queueing time [10], understanding urban scenes [11], etc.

Public transit ridership information is generally a prerequisite for public transportation network planning, route optimization, improvement of service quality and travel scheduling [12]. However, non-WiFi based approaches, such as smart cards [13][14][15], video [16][17][18] and other types of sensors [19][20][21], often suffer from intractable limitations, such as insufficient information, deployment inconveniences and environmental dependence. In contrast, passive WiFi sensing is immune to these constraints to some extent, and thus becomes appealing in the field.

Intuitively, passive WiFi sensing has the advantages of low costs, large coverage, good scalability and non-intrusive detection. The relevance between ridership and WiFi sniffing data was already corroborated in [22][23]. Thereafter, although various passive WiFi sensing based methods relying on, e.g. a fixed threshold [24], statistical models [25][26] and machine learning [27], have been proposed to estimate the ridership in an offline manner, it is still challenging to accomplish similar tasks in an online manner due to the fact that several practical limitations, including sniffing opportunistically, unknown dynamic transmission boundary, MAC randomization, etc., are much more difficult to be tackled, especially given very limited WiFi sniffing data in the online case in comparison with the offline case.

This paper proposes an online public transit ridership estimation algorithm based on the long short-term memory (LSTM) neural network. Unlike the offline methods producing estimated number of passengers between two adjacent bus stops after collecting all WiFi sniffing data from a complete bus trip [22], [23], [24], [25], [26], [27], the proposed online estimation algorithm is able to predict the current ridership given only an input sequence containing the preprocessed WiFi sniffing data in a very short period of time in the past. Moreover, in order to reduce the impact of limited WiFi sniffing data caused by opportunistic sniffing and sufficiently exploit the temporal features contained in the input sequence, a bidirectional LSTM (BiLSTM) neural network is adopted to extract contextual features from both forward and backward time series. In addition, a convolutional neural network (CNN) is employed to capture intrinsic correlations among elements of one input, involving both fixed and random MAC addresses. Finally, an attention module is incorporated to form the

attention based convolutional BiLSTM (termed AConvBiLSTM) model, so as to distinguish different importances of an input sequence at different times by automatically assigning different weights, thus enhancing the estimation performance of the original model. To the best of our knowledge, this is the first to estimate online public transit ridership based on passive WiFi sensing and deep learning models.

In order to verify the estimation performance of the proposed algorithm, an experimental public transit ridership monitoring system is deployed on 8 buses to collect WiFi sniffing data in Hohhot, China. The experimental results show that the proposed algorithm can provide online public transit ridership information accurately and effectively. Specifically, the proposed algorithm achieves the root mean squared error (RMSE) and mean absolute error (MAE) of the ridership estimation as low as 3.30 and 2.61, respectively, which outperforms the other 4 online algorithms and the state-of-the-art offline algorithm.

The core contributions are summarized as follows.

- We resolve the online public transit ridership estimation problem by utilizing an efficiently designed deep learning network to capture the in-depth spatio-temporal features in WiFi sniffing data.
- We design an innovative online public transit ridership monitoring system using a customized WiFi sniffer, which can be easily generalized to new routes, thus making the deployment of the system in large-scale real-world scenarios readily available.
- We deploy an experimental monitoring system in authentic scenarios and conduct extensive experiments. Encouraging results demonstrate that our system makes a great progress towards achieving accurate online public transit ridership estimation through passive WiFi sensing.

The rest of this paper is organized as follows. Section II reviews the literature related to public transit ridership estimation. Section III introduces the public transit ridership monitoring system. Section IV describes the proposed online ridership estimation algorithm. Section V reports the experimental design and evaluation results. Section VI concludes this paper and points out the future research direction.

II. RELATED WORK

This section briefly reviews the literature on public transit ridership estimation based on non-WiFi and WiFi based approaches.

A. Non-WiFi based Ridership Estimation

Initially, smart cards were employed to estimate ridership, and the performance was restricted by the lack of passengers' alighting data [13][14][15]. Thereafter, many camera based solutions were developed, in which Hsu et al. [16] utilized a zenithal camera, and introduced a convolutional autoencoder to estimate ridership, however, it suffered from the occlusion and the poor illumination at night. Besides, Bernini et al. [17] implemented a stereo vision system to count passengers of the region of interest (ROI), and the performance was limited due to inadequate discriminating capabilities between human and nonhuman elements and high computational complexities.

In addition to the above approaches, other types of sensors were also deployed to monitor ridership. In [19], light-sensitive wireless sensors were employed to detect the passengers who pass through the front and the rear door, however, the accuracy declines evidently in high passenger densities. In [20], passive infrared (PIR) sensors were arranged on the roof above each seat to detect occupancy, but they can not count standing passengers, and the installation of many sensors increases the cost. In [21], an RFID distance scan module was integrated into a people counting system to locate and monitor passengers, and more practical tests need to be conducted to improve its accuracy.

It follows from the above introduction that non-WiFi based approaches are restricted by insufficient information, environmental dependency, high cost and low accuracy.

B. WiFi based Ridership Estimation

WiFi sniffing data contains spatial-temporal features of passengers' mobile devices, and can thus be used to estimate ridership. Moreover, sniffing data related to non-passengers, e.g. pedestrians and passengers in other vehicles, can also be detected by WiFi sniffers. To this end, some dedicated works utilized the characteristics of one MAC address, such as the median value of received signal strength (RSS) measurements, detection times and the detection duration, to filter out useless sniffing data, which are termed the fixed threshold based methods [24], [28], [29], [30]. However, the accuracy of these methods is restricted by some practical issues, including sniffing opportunistically and MAC randomization.

In order to improve the accuracy, some works further combined other data sources to build mathematical models, in which Ji et al. [26] proposed a hierarchical bayesian model and utilized WiFi sniffing data and smart card reader data to estimate ridership, achieving better performance. Besides, Håkegård et al. [31] exploited the characteristics of MAC addresses to establish a statistical distribution model to estimate ridership, and then combined the boarding data provided by the Automatic Passenger Counting (APC) system to calibrate the results. In addition to the mathematical models, machine learning models were also developed, in which Pu et al. [27] utilized the Fuzzy C-Means (FCM) clustering algorithm to filter out the useless sniffing data and the Random Forest (RF) regression model to estimate ridership. However, the time interval when one MAC address is detected is irregular, and the characteristics, such as detection duration, detection times, etc., can not be extracted until completing a whole bus journey. Thus, the above methods based on fixed thresholds, mathematical models and machine learning can only estimate ridership in offline scenarios.

In summary, neither the non-WiFi based approaches nor the emerging WiFi based approaches can address the issues confronted by online public transit ridership estimation. However, the WiFi based approaches demonstrate the advantages of low cost, easy deployment, large coverage and good performance in poor illumination conditions, which are major shortages of the non-WiFi based approaches. As such, it is imperative to understand the rules governing the relationship between the

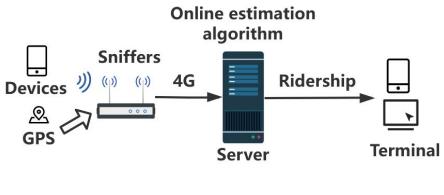


Fig. 1. The system architecture of the online public transit ridership monitoring system.



passenger number and the WiFi sniffing data as well as to estimate ridership in online scenarios.

III. SYSTEM DESIGN

This section introduces the system architecture and the customized WiFi sniffer in detail.

A. System Architecture

The system architecture presented in Fig. 1 consists of three components, including a sniffer, a server end and a front end.

Probe frames broadcasted by WiFi enabled mobile devices inside and outside a bus can be detected and stored by the WiFi sniffer which is deployed in the bus. Moreover, GPS information of the bus, including the latitude, longitude and speed, is also recorded every second by the WiFi sniffer. After that, WiFi sniffer rearranges the sniffing data and GPS data with the same timestamp into one item and sends it to the remote server through e.g. a 4G data link, where the proposed online ridership estimation algorithm is fed with these data to predict ridership. In the end, the online ridership estimates can be published or integrated with other public transportation application systems if needed, for the purposes of public transportation services and management.

B. Customized WiFi Sniffer

As shown in Fig. 2, the customized WiFi sniffer is designed to collect WiFi sniffing data, and is composed of the following three components.

Data processing component: A Raspberry Pi 4B is employed for processing sniffed frames. The Raspberry Pi 4B is a low-cost computing platform with a 1.5 GHz quad-core ARM Cortex A72 and a 4GB LPDDR4-3200 RAM, supports 802.11 b/g/n/ac Wireless LAN, and further installs the OpenWrt 18.06 as its operating system.

Sensing component: This component includes the following three modules.

- WiFi sniffing module: Since the WiFi adapter in Raspberry Pi 4B cannot operate in the monitor mode, an external USB WiFi adapter (TENDA U3) is used to implement passive WiFi scanning. It is noted that mobile devices send probe frames on all channels in the supported spectrum (typically both 2.4 GHz and 5 GHz), and the USB WiFi adapter is set to listen to Channel 6 in 2.4 GHz in this paper. Besides, to protect passengers' privacy, only encrypted MAC addresses, RSS values and

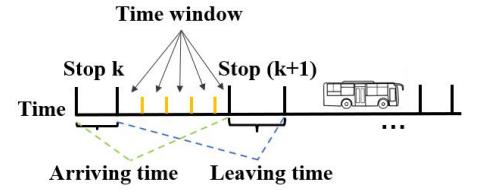


Fig. 3. The correlation between time windows and bus stops.

timestamps in the frames are reserved, other information will be discarded.

- GPS module: To record high-resolution longitude, latitude and speed data every second, an external USB GPS adapter (HUIHANG USB-Port-GPS U) with 165dBm sensitivity and 5Hz data update rate is used.
- Time synchronization module: If the WiFi sniffer is stopped, its clock will not work until it is restarted, so it is necessary to synchronize its clock with the GPS clock. Thus, an external clock module (Soumiety DS3231) with low power consumption and $\pm 5ppm$ timing accuracy is used.

Power supply: In order to support the WiFi sniffer to continuously work during at least one bus trip, a high capacity portable charger (ROMOSS sense6) with 20000 mAh is used.

IV. ONLINE RIDERSHIP ESTIMATION ALGORITHM

In this section, the online ridership estimation algorithm, including data preprocessing and the AConvBiLSTM model will be introduced in detail.

A. Data Preprocessing

In this subsection, how to process sniffing data into model inputs will be described.

1) *Time window mechanism:* In order to capture spatial-temporal features contained in the WiFi sniffing data, the time window shown in Fig. 3 is set as the basic unit to process sniffing data. As can be seen, the period when a bus travels between two adjacent stops is separated into several time windows, and the sniffing data during every time window is exploited for the next data filtering operation; in contrast, the sniffing data when the bus stops at a stop is discarded because the passenger number is constantly changing. In order to determine whether a bus is arriving at a bus stop, the GPS data collected by the WiFi sniffer is used. Specifically, the latitude and longitude data of each stop is imported into the server in advance, and a bus is inferred to be at a stop when the distance between the bus and the stop is less than 20m and the bus speed is less than 1km/h.

2) *Data filtering:* Since the detection range of the WiFi sniffer is much larger than the cross-sectional area (CSA) of a bus, the collected WiFi sniffing data includes not only useful frames sent by mobile devices inside the bus but also useless frames sent by various devices outside the bus. Therefore, it is imperative to filter out useless frames from the collected sniffing data.

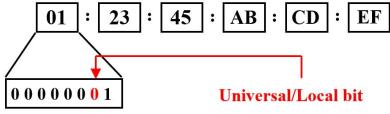


Fig. 4. The illustration of the Universal/Local bit in Fig. 5. The proposed AConvBiLSTM model structure.

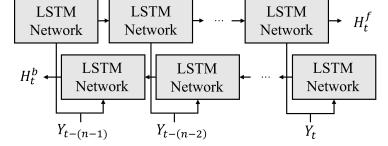
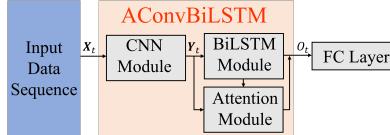


TABLE I
ELEMENTS IN EACH INPUT DATA

Element	Definition
MACNum	The number of MAC addresses
fixedMACNum	The number of fixed MAC addresses
randomMACNum	The number of random MAC addresses
avgFixedMACProbeNum	The number of probe frames with fixed MAC addresses divided by the size of the current time window
avgRandomMACProbeNum	The number of probe frames with random MAC addresses divided by the size of the current time window
meanProbeRSS medianProbeRSS	Mean, median, standard deviation and mode of the average of RSS values in the probe frames corresponding to each MAC address (dBm)
stdProbeRSS modeProbeRSS	
meanFrameRSS medianFrameRSS	Mean, median, standard deviation and mode of the average of RSS values in all frames corresponding to each MAC address (dBm)
stdFrameRSS modeFrameRSS	
meanProbeBusSpeed medianProbeBusSpeed	Mean, median, standard deviation and mode of the average bus speed when probe frames corresponding to each MAC address are detected (km/h)
stdProbeBusSpeed modeProbeBusSpeed	

The RSS value associated with a probe frame from a mobile device normally decreases with the increasing distance between the WiFi sniffer and the mobile device, which is often utilized to filter out useless sniffing data by establishing a proper threshold [31][32]. However, experiments revealed that frames other than probe frames can also be sniffed, and extra RSS values associated with such frames can be utilized to improve the sample average of RSS values corresponding to one MAC address. Therefore, when this improved average of RSS values in a time window is less than the threshold, the sniffing data associated with this MAC address is filtered out.

3) *Organizing input data:* In order to sufficiently capture spatial-temporal features contained in the sniffing data, seventeen elements defined in Tab. I are included in each input data, and are categorized into 4 groups according to their physical meanings.

The first group includes the numbers in relation to different types of MAC addresses and the corresponding probe frames. These numbers are intuitively correlated with the number of ridership, and moreover, the numbers of fixed and random MAC addresses are separately considered to mitigate the influence of MAC randomization. Therein, a MAC address is fixed when the value of the Universal/Local (U/L) bit shown in Fig. 4 is set as 1 [33]. In addition, because the size of the last time window between two adjacent bus stops might be less than the default setting, the last two of the five elements are derived by dividing the size of the current time window.

The remaining three groups reflect the spatial features of the sniffing data, so as to amplify the variability among inputs and make full use of sniffing data. Therein, the work in [27] has confirmed that the WiFi sniffer in a bus often collects a large percentage of probe frames related to non-passengers when the bus drives at a slow speed; therefore, the elements related to the bus speed are included in the input data to reflect such

characteristics.

B. AConvBiLSTM Model

This subsection describes the ridership prediction by the AConvBiLSTM model and the functions of different modules.

1) *Problem formulation:* The purpose of the AConvBiLSTM model is to predict the current ridership given the input sequence $\mathbf{X}_t = [X_{t-(n-1)}, X_{t-(n-2)}, \dots, X_t]$ corresponding to the past n time windows, in which $X_t = [f_t^1, f_t^2, \dots, f_t^{17}]^T$ denotes all elements of the input data at the t -th time window. Afterwards, the ridership estimate, denoted \hat{R}_k , between the k -th stop and $(k+1)$ -th stop shown in Fig. 3 can be calculated by averaging the predictions within these two stops.

2) *Model structure:* As shown in Fig. 5, the AConvBiLSTM model including a CNN module, a BiLSTM module and an attention module is proposed to predict public transit ridership. The CNN module is exploited to extract the most representative features and capture correlations among different elements, and then is connected by the BiLSTM module to further obtain the bidirectional time-series features of the input sequence. In addition, the attention module is integrated to automatically distinguish different levels of importances of the input sequence at different times. Finally, the BiLSTM module is followed by a fully connected (FC) layer which is the regression layer to capture the non-linear relationship between elements and the ridership. Each module shall be introduced in the following parts in detail.

3) *CNN module:* The multivariate input includes several key elements, i.e. the number of fixed MAC addresses, random MAC addresses and the corresponding number of probe frames, which generally vary with the number of mobile devices, the portion of devices adopting MAC randomization, and specific the MAC randomization strategies. Precisely identifying which random MAC address comes from which

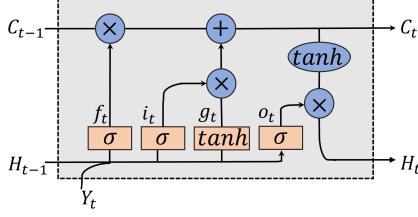


Fig. 7. The LSTM cell.

mobile device is not only hard and even impossible, but also unnecessary for only estimating the ridership. Therefore, the CNN module is utilized to capture correlations among these elements, thus reducing the influence of MAC randomization.

Given the input sequence \mathbf{X}_t , one dimensional convolution operation is performed over one element of the input sequence, denoted X_t , with the following convolution kernel filter to acquire the local perceptual domain by a sliding filter:

$$Y_t = \sigma(W * X_t + b), \quad (1)$$

where W is the weights of the filter, b is the bias, X_t is the input at the t -th time step, the symbol $*$ represents the convolution operation, σ is the activation function, and Y_t is the output.

It is noted that the pooling layer is not attached to the convolutional layer, considering the fact that the dimension of the input sequence is not large and the pooling layer will lead to feature loss and worse prediction performance.

4) *BiLSTM module*: Compared the RNN [34], the LSTM introduces an adaptive gate mechanism to solve the problem of gradient disappearance and explosion in the long sequence training process [35], and the BiLSTM can further capture contextual features [36]. Moreover, temporal features are contained in the input sequence. Therefore, the BiLSTM module is connected after the CNN module to capture two directional time-series features contained in the input sequence. As shown in Fig. 6, the BiLSTM module contains two directional LSTMs, one for forward propagation and another for backward propagation. The input data of the BiLSTM module is the output of the CNN module, denoted $\mathbf{Y}_t = [Y_{t-(n-1)}, Y_{t-(n-2)}, \dots, Y_t]$, and the hidden states for each time step, denoted $\mathbf{H}_t = [H_{t-(n-1)}, H_{t-(n-2)}, \dots, H_t]$, will be computed using the forward state H_t^f and the backward state H_t^b :

$$H_t = H_t^f \oplus H_t^b, \quad (2)$$

where \oplus represents element-wise summarizing.

As shown in Fig. 7, the hidden state H_t at the t -th time step is calculated as follows:

$$i_t = \sigma(W_{xi}Y_t + W_{hi}H_{t-1} + b_i), \quad (3)$$

$$f_t = \sigma(W_{xf}Y_t + W_{hf}H_{t-1} + b_f), \quad (4)$$

$$g_t = \tanh(W_{xg}Y_t + W_{hg}H_{t-1} + b_c), \quad (5)$$

$$C_t = i_t \odot g_t + f_t \odot C_{t-1}, \quad (6)$$

$$o_t = \sigma(W_{xo}Y_t + W_{ho}H_{t-1} + b_o), \quad (7)$$

$$H_t = o_t \odot \tanh(C_t), \quad (8)$$

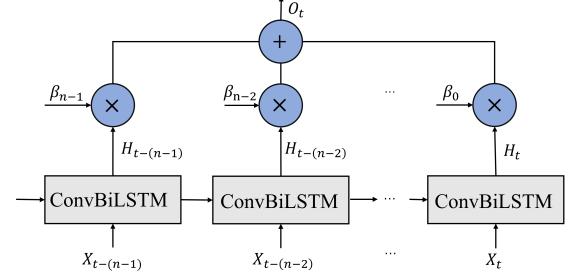


Fig. 8. The attention mechanism with ConvBiLSTM networks.

where Y_t is the input of the LSTM at the t -th time step, σ is the activation function, i_t, f_t, o_t are respectively the input gate, the forget gate and the output gate, respectively, C_t is the cell state, \odot represents the matrix element-wise product, and W 's and b 's are the weights and bias, respectively.

5) *Attention module*: Since the inputs in historical time windows have different levels of importances for predicting current ridership, the attention module which can discover associations among different features [37][38] is integrated into the ConvBiLSTM model to distinguish different importances by automatically assigning different weights.

As shown in Fig. 8, the output of the AConvBiLSTM model at the t -th time step is computed as a weighted summation of the output of the BiLSTM network \mathbf{H}_t as follows:

$$O_t = \sum_{k=0}^{n-1} \beta_k H_{t-k}, \quad (9)$$

where n is the length of the input sequence and β_k is the attention value at the $(t-k)$ -th time step. The attention value β_k can be calculated as

$$\beta_k = \frac{\exp(s_k)}{\sum_{k=0}^{n-1} \exp(s_k)}. \quad (10)$$

The score $\mathbf{s} = (s_0, \dots, s_{n-1})^T$ indicates the importance of each part in the input sequence, and can be computed as

$$s_k = V_s^T \sigma(W_{ys}Y_{t-k} + W_{hs}H_{t-k}), \quad (11)$$

where $k = 0, \dots, (n-1)$, and V_s, W_{ys}, W_{hs} are the learnable parameters.

V. EXPERIMENTS

In this section, the experiments are conducted to verify the effectiveness and superiority of the proposed algorithm.

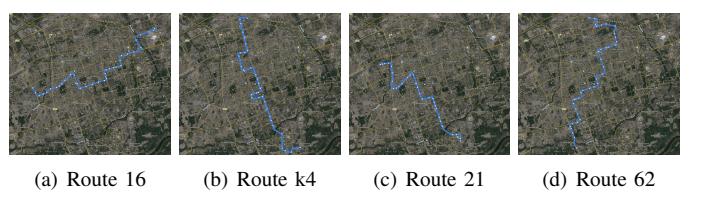


Fig. 9. Four bus lines for data collection.

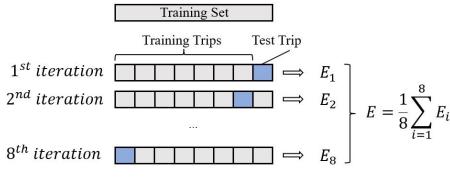


Fig. 10. The method of data set partitioning.

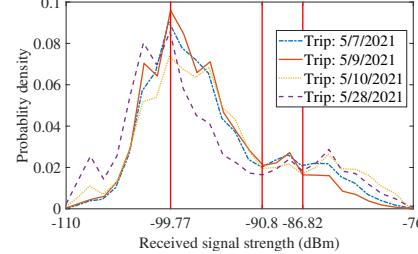


Fig. 11. Probability densities of RSS values associated with all probe frames sniffed in four trips.

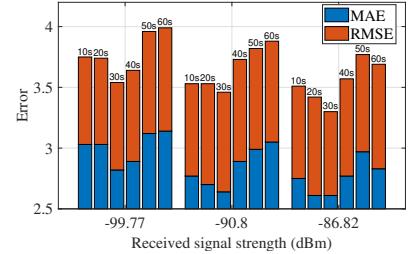


Fig. 12. Performance comparison of the proposed algorithm with different sizes of the time window and RSS threshold values (the number of time windows is 5).

TABLE II
STATISTICAL SUMMARY OF THE COLLECTED EXPERIMENTAL WiFi SNIFFING DATA

Trip No.	Route No.	Trip Date	The Day of The Week	Trip Start Time	Trip End Time	The Number of Probe Frames
1	16	5/7/2021	Friday	14:12:05	15:20:00	33391
2	16	5/9/2021	Sunday	16:19:00	17:37:10	53037
3	16	5/10/2021	Monday	15:11:20	16:17:15	38602
4	k4	3/13/2021	Saturday	15:12:15	16:20:00	26070
5	k4	5/25/2021	Tuesday	17:09:20	18:23:30	48018
6	k4'	5/28/2021	Friday	17:32:25	18:41:25	45171
7	21	5/16/2021	Sunday	16:41:25	18:02:55	36401
8	62	7/17/2021	Saturday	13:31:05	14:47:30	37642

A. Setup

In order to evaluate the estimation performance of the proposed algorithm, real-world sniffing data was collected in Hohhot, Inner Mongolia Autonomous Region, China. Fig. 9 shows the selected four bus lines which pass through not only urban areas with big populations but also second-ring roads with small populations, reflecting the overall public transportation situations in Hohhot.

Tab. II shows the statistical summary of the dataset, in which k4 and k4' are essentially the same bus lines in the opposite directions. Notice that, each data collection period covered the whole trip of one bus line, during which a WiFi sniffer was carried by a volunteer sitting in the middle of a bus. Since passengers alighting buses do not swipe cards, the ground truth ridership (including the bus driver) is obtained through manual counting.

Root mean squared error (RMSE) and mean absolute error (MAE) are calculated to compare the estimation performance of different algorithms. In addition, in order to tune the hyperparameters of the model, K -fold cross-validation is utilized, as shown in Fig. 10, where E represents the overall estimation performance of the algorithm, e.g. RMSE and MAE.

For ease of comparison, both the proposed algorithm and its counterparts are executed in an offline fashion on a laptop (Windows 11, AMD Ryzen 7 5800H @3.20 GHz, NVIDIA GeForce RTX 3060 laptop), and the python library, i.e. Keras 2.1.6, is utilized to build models. It is noticeable that, given a trip, the online algorithms will produce a sequence of ridership estimates during this trip, whereas the offline algorithms will produce only one estimate at the end of this trip. Moreover, the loss function is formulated using MSE and is further solved by the Adam optimizer [39].

B. Parameter Selection

This subsection investigates how to determine various parameters, including the RSS threshold for data filtering, the size of the time window for data preprocessing and the number of inputs in the input data sequence.

Firstly, the RSS threshold for data filtering is empirically determined according to the following experiments. Fig. 11 illustrates the probability densities of RSS values associated with all probe frames sniffed in four trips. As can be seen, most of the probe frames are distributed with the RSS values less than -90.8 dBm, indicating that all these frames are sniffed from mobile devices far from the sniffers; that is to say, these mobile devices are most likely to be outside of the buses, so that these frames should be excluded. In contrast, probability density peaks occur between -86.82 dBm and -76 dBm, meaning that the mobile devices generating the frames with their RSS values falling between this range are largely close to the sniffers; in other words, these mobile devices are most likely to be inside of the buses, so that these frames should be included for use in our algorithm. On these grounds, we speculate that the appropriate threshold should reside between -90.8 dBm and -86.82 dBm. In order to further verify the above analysis, the estimation performance of the proposed algorithm with the RSS threshold values taken from -99.77 dBm, -90.8 dBm and -86.82 dBm is presented in Fig. 12, in which each result is averaged over three estimates to mitigate the influence of the randomness of the deep learning model. As can be seen, the estimation performance of the proposed algorithm is evidently better with the threshold of -90.8 dBm than with the threshold of -99.7 dBm, but is slightly worse than with the threshold value of -86.82 dBm, confirming the above analysis. Therefore, the RSS threshold is set as -

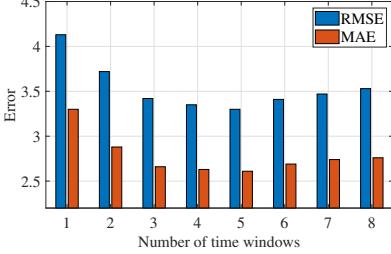


Fig. 13. Performance comparison of the proposed algorithm with respect to different numbers of inputs estimates under different modules for all trips. in the input data sequence (RSS threshold is -86.82dBm and the size of the time window is equal to 30 seconds).

86.82dBm.

Secondly, the size of the time window for data preprocessing is determined according to the following experiments. Fig. 12 illustrates the estimation performance of the proposed algorithm with different sizes of the time window, and as can be seen, regardless of the threshold value, the proposed algorithm produces the smallest errors when the size is equal to 30 seconds, indicating that the input data contains enough features and the corresponding training data is sufficient when the size is equal to 30 seconds. In contrast, increasing the size from 30 seconds results in insufficient model training, and reducing the size results in insufficient features in the input data and small feature differences among inputs.

Finally, the number of inputs in the input data sequence is determined according to the following experiments. Fig. 13 shows the estimation performance of the proposed algorithm with respect to different numbers of inputs, and as can be seen, minimal errors are obtained when the number is 5, indicating that although the LSTM module mitigates the influence of the gradient disappearance and explosion to some extent, the estimation performance of the proposed algorithm will decrease if the number is greater than 5.

C. Ablation Experiment

This subsection verifies the effectiveness of different modules used in the AConvBiLSTM model and explains the necessity of using all frames to improve the sample average of RSS values corresponding to one MAC address for data filtering.

Fig. 14 illustrates the cumulative density functions (CDFs) of the absolute errors of all estimates under different modules for all trips to verify their effectiveness. As can be seen, when the attention module or CNN module is removed from the AConvBiLSTM model, the estimation performance of the algorithm degrades to some extent, revealing that including the two modules contributes to improving the estimation performance. In contrast, inserting pooling layers does not evidently improve the performance of the AConvBiLSTM model.

In addition, the effectiveness of exploiting all frames to improve the sample average of RSS values corresponding to one MAC address for data filtering is evaluated by comparing

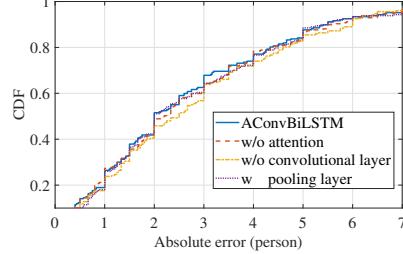


Fig. 14. The CDF of the absolute errors of all estimates under different modules for all trips.

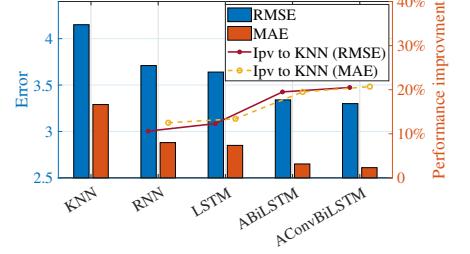


Fig. 15. Estimation performance comparison of different online algorithms.

the estimation performance of the proposed algorithm using only probe frames or using all frames. The results indicate that the RMSE and MAE of the proposed algorithm using all frames are 3.30 and 2.61, respectively, which are 3.2% and 4% lower than those using only probe frames, confirming the effectiveness of using all frames.

D. Performance Comparison with Different Algorithms

This subsection compares the estimation performance of the proposed algorithm with that of other online algorithms and the state-of-the-art offline algorithm.

Specifically, the proposed algorithm is compared with the online algorithms relying on the popular models, including the KNN model [40], the RNN model [41], the LSTM model [42] and the attention based BiLSTM (ABiLSTM) model [43]. Fig 15 presents estimation errors of different online algorithms and the degree of performance improvement (Ipv) with the algorithm relying on the KNN model as the baseline. As can be seen, the estimation performance of the proposed algorithm is best, and is respectively 20.5% and 20.7% higher than those of the baseline in terms of the RMSE and MAE.

In order to have a clear observation, Fig. 16 visualizes the estimated riderships of different online estimation algorithms and the corresponding ground truth in different trips, from which the following four conclusions can be drawn.

Firstly, since the CNN module in the AConvBiLSTM model captures the spatial features among different elements of the input data, which cannot be extracted by models in other algorithms, the proposed algorithm often performs best during periods of high passenger fluctuations and periods of low passenger flow, such as in the beginning and ending of a journey.

Secondly, the algorithms based on the AConvBiLSTM model and the ABiLSTM model attain smaller estimation errors than those of other online algorithms, which is attributable to the fact that the attention module is able to improve the estimation performance by distinguishing different importances of an input sequence at different times and the BiLSTM module is able to extract contextual features in the input data sequence.

Thirdly, the KNN based algorithm produces significantly larger estimation errors in the beginning and ending of a trip than other algorithms, and is extremely insensitive to input

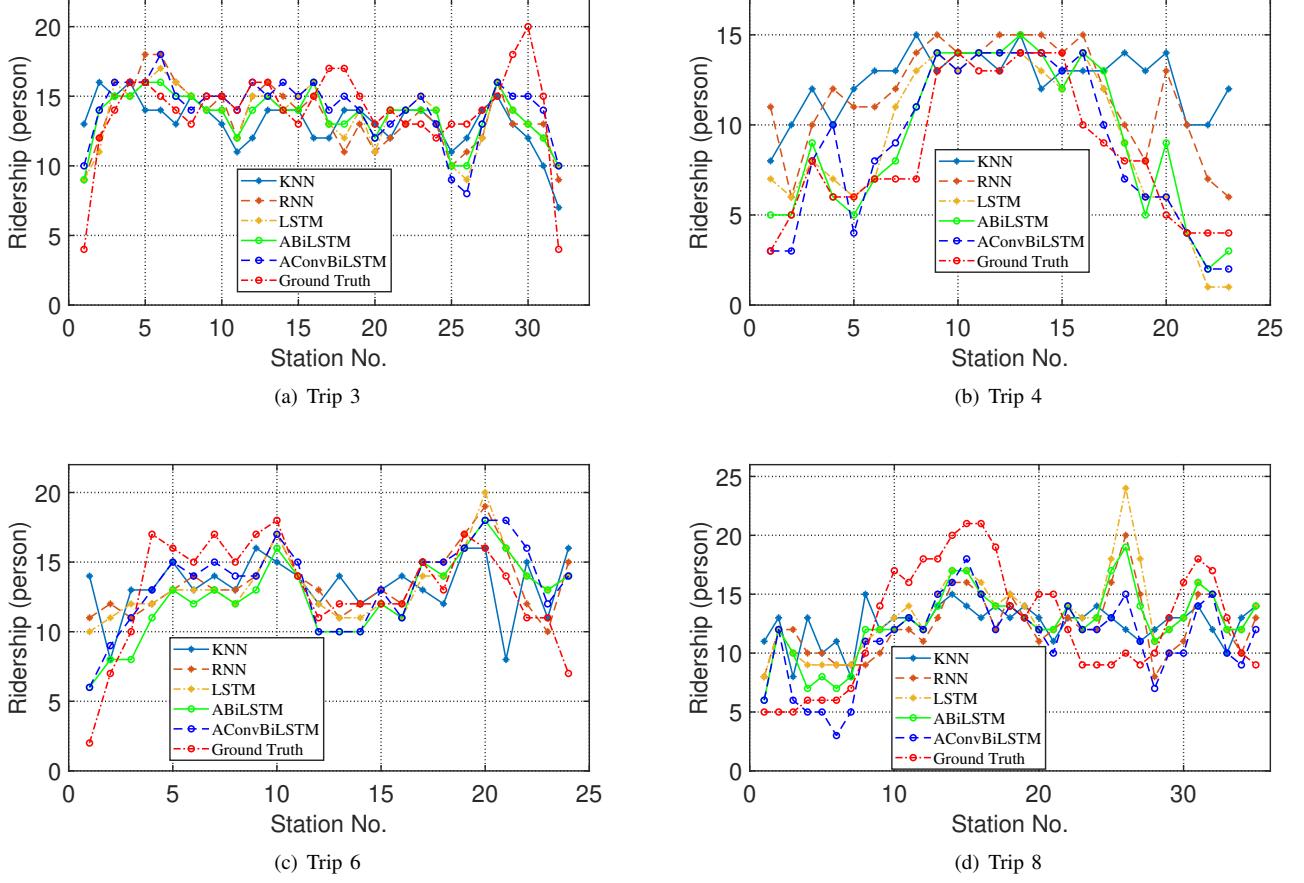


Fig. 16. Estimation performance comparison of different online algorithms in different trips.

changes because it is not able to capture time-series features in the input sequence.

Finally, all the algorithms including the proposed algorithm do not perform well near certain stations (e.g. schools, hospitals, etc.), which is probably degraded by interferences caused by specific passengers. For example, the bus stations between Station No. 10-17 in Fig. 16 (d) are flanked by middle schools and hospitals, where many students on the buses normally do not bring mobile devices and many elders carry phones without WiFi enabled, thus disrupting the mapping from WiFi sniffing data to the number of passengers.

In addition, the standard deviations of the RMSE and MAE of all trips with different algorithms are plotted in Fig. 17 to evaluate the robustness of different algorithms. As can be seen, the standard deviations of the proposed algorithm are comparable to and even better than those of its counterparts.

TABLE III
PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART OFFLINE ALGORITHM

Algorithm	Scenario	RMSE	MAE
Algorithm in [27]	Offline	3.80	2.93
The proposed algorithm	Online	3.30	2.61

In what follows, the proposed online algorithm is compared with the state-of-the-art offline algorithm. Tab. III presents the

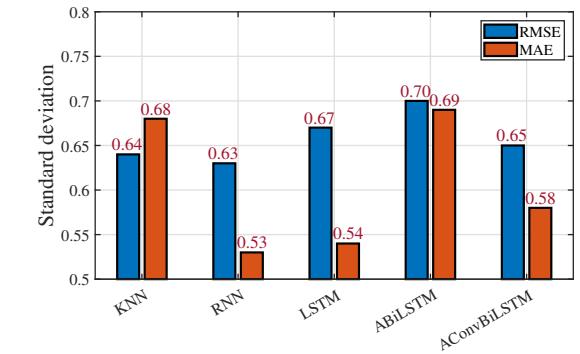


Fig. 17. The standard deviation of the RMSE and MAE of all trips with different algorithms.

estimation errors of the proposed algorithm and the offline algorithm proposed in [27], and as can be seen, the RMSE and MAE of the proposed online algorithm are respectively 13.2% and 10.9% smaller than those of the offline algorithm. It turns out that the proposed algorithm is not only capable of estimating ridership with low errors in an online manner, but will be applied to an increasing number of online real-world scenarios.

E. Performance Comparison on Weekends and Weekdays

TABLE IV
AVERAGE ESTIMATION ERRORS COMPARISON ON WEEKENDS AND WEEKDAYS

Trip time	Trip No.	RMSE	MAE
Weekend	2,4,7,8	3.72	2.99
Weekday	1,3,5,6	2.88	2.23

The estimation performance of the proposed algorithm on weekends and weekdays are listed in Tab. IV, which shows that the RMSE and MAE on weekdays are respectively 22.58% and 25.41% lower than those on weekends. The major reason is that, a large number of mobile devices outside the buses broadcast a particularly large amount of useless sniffing data, which is collected by sniffers on the buses and thus disrupts the features in the input data sequence; in particular, this situation is more likely to happen on weekends, when there are more vehicles on the road and more pedestrians on the pedestrian streets next to some stops compared to weekdays. The above analysis is verified by comparing various data on weekends and weekdays when the buses in Route 16 passed through two adjacent stops next to a busy mall: the elapsed time, the number of collected probe frames and the corresponding absolute error on weekends are 565 seconds, 10145 and 5, respectively, which are 98.2%, 227.2% and 400% bigger than those on weekdays, confirming the influences of a large amount of interference data.

VI. CONCLUSION AND FUTURE WORK

This paper dealt with the problem of estimating public transit ridership through passive WiFi sensing in the online scenario. To be specific, the online public transit ridership monitoring system consists of a customized WiFi sniffer and an online ridership estimation algorithm using the AConvBiLSTM model, in which the CNN module captures correlations among different elements of the input data and finds the most representative elements, the BiLSTM module extracts contextual time-series features, and the attention module distinguishes the most important parts of the input data sequence, enhancing the estimation performance. The experimental system temporarily deployed on 8 buses corresponding to 4 routes in Hohhot, China, confirmed the feasibility of estimating online ridership through passive WiFi sensing as well as the effectiveness and superiority of the proposed algorithm.

There are some future studies that can be extended to the online public transit ridership estimation system. First, it is worthwhile to extend the proposed algorithm to other public transportation scenarios, such as subways; second, it is attractive and meaningful to fuse video-based estimation methods with wireless-based estimation methods to obtain better estimation performance.

REFERENCES

- [1] Y. Fukuzaki, M. Mochizuki, K. Murao, and N. Nishio, “A pedestrian flow analysis system using wi-fi packet sensors to a real environment,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 721–730.
- [2] B. Huang, G. Mao, Y. Qin, and Y. Wei, “Pedestrian flow estimation through passive wifi sensing,” *IEEE Transactions on Mobile Computing*, vol. 20, no. 04, pp. 1529–1542, 2021.
- [3] A. Musa and J. Eriksson, “Tracking unmodified smartphones using wifi monitors,” in *Proceedings of the 10th ACM conference on embedded network sensor systems*, 2012, pp. 281–294.
- [4] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie, “A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252–1266, 2015.
- [5] B. Huang, Z. Xu, B. Jia, and G. Mao, “An online radio map update scheme for wifi fingerprint-based localization,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6909–6918, 2019.
- [6] B. Huang, R. Yang, B. Jia, W. Li, and G. Mao, “A theoretical analysis on sampling size in wifi fingerprint-based localization,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3599–3608, 2021.
- [7] L. Hao, B. Huang, B. Jia, and G. Mao, “Dhcloc: A device heterogeneity tolerant and channel adaptive passive wifi localization method based on dnn,” *IEEE Internet of Things Journal*, 2021.
- [8] Z. Xu, B. Huang, and B. Jia, “An efficient radio map learning scheme based on kernel density function,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 13 315–13 324, 2021.
- [9] A. Di Luzio, A. Mei, and J. Stefa, “Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests,” in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.
- [10] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, “Measuring human queues using wifi signals,” in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 235–238.
- [11] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Deneux, “Multimodal information fusion for urban scene understanding,” *Machine Vision and Applications*, vol. 27, no. 3, pp. 331–349, 2016.
- [12] L. Liu, L. Sun, Y. Chen, and X. Ma, “Optimizing fleet size and scheduling of feeder transit services considering the influence of bike-sharing systems,” *Journal of Cleaner Production*, vol. 236, p. 117550, 2019.
- [13] X. Ma, C. Liu, H. Wen, Y. Wang, and Y.-J. Wu, “Understanding commuting patterns using transit smart card data,” *Journal of Transport Geography*, vol. 58, pp. 135–145, 2017.
- [14] L.-M. Kieu, A. Bhaskar, and E. Chung, “A modified density-based scanning algorithm with noise for spatial travel pattern analysis from smart card afc data,” *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 193–207, 2015.
- [15] K. Lu, J. Liu, X. Zhou, and B. Han, “A review of big data applications in urban transit systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2535–2552, 2020.
- [16] Y.-W. Hsu, Y.-W. Chen, and J.-W. Perng, “Estimation of the number of passengers in a bus using deep learning,” *Sensors*, vol. 20, no. 8, p. 2178, 2020.
- [17] N. Bernini, L. Bombini, M. Buzzoni, P. Cerri, and P. Grisleri, “An embedded system for counting passengers in public transportation vehicles,” in *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2014, pp. 1–6.
- [18] J. M. Grant and P. J. Flynn, “Crowd scene understanding from video: a survey,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, pp. 1–23, 2017.
- [19] P. Chen, H. Chen, C. Zhang, and Z. Tang, “Real-time bus passenger flow statistics scheme based on light-sensitive wireless sensor network,” in *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 8490–8494.
- [20] A. Rahmatulloh, F. M. Nursuwars, I. Darmawan, and G. Febrizki, “Applied internet of things (iot): The prototype bus passenger monitoring system using pir sensor,” in *2020 8th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2020, pp. 1–6.
- [21] R. Khoebatl, T. Laohapensaeng, and R. Chaisricharoen, “Passenger monitoring model for easily accessible public city trams/trains,” in *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTICON)*. IEEE, 2015, pp. 1–6.
- [22] V. Kostakos, T. Camacho, and C. Mantero, “Wireless detection of end-to-end passenger trips on public transport buses,” in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1795–1800.

- [23] T. Oransirikul, R. Nishide, I. Piumarta, and H. Takada, "Measuring bus passenger load by monitoring wi-fi transmissions from mobile devices," *Procedia Technology*, vol. 18, pp. 120–125, 2014.
- [24] D. B. Paradeda, W. K. Junior, and R. C. Carlson, "Bus passenger counts using wi-fi signals: some cautionary findings," *Transportes*, vol. 27, no. 3, pp. 115–130, 2019.
- [25] T. A. Myrvoll, J. E. Häkegård, T. Matsui, and F. Septier, "Counting public transport passenger using wifi signatures of mobile devices," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.
- [26] Y. Ji, J. Zhao, Z. Zhang, and Y. Du, "Estimating bus loads and od flows using location-stamped farebox and wi-fi signal data," *Journal of Advanced Transportation*, vol. 2017, 2017.
- [27] Z. Pu, M. Zhu, W. Li, Z. Cui, X. Guo, and Y. Wang, "Monitoring public transit ridership flow by passively sensing wi-fi and bluetooth mobile devices," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 474–486, 2020.
- [28] A. Hidayat, S. Terabe, and H. Yaginuma, "Determine non-passenger data from wifi scanner data (mac address), a case study: Romango bus, obuse, nagano prefecture, japan," *International Review for Spatial Planning and Sustainable Development*, vol. 6, pp. 154–167, 07 2018.
- [29] A. Hidayata, S. Terabea, and H. Yaginumaa, "Time travel estimations using mac addresses of bus, passengers: A point to path-qgis analysis," *Geoplanning: Journal of Geomatics and Planning*, vol. 5, no. 2, pp. 259–268, 2018.
- [30] T. Oransirikul, I. Piumarta, and H. Takada, "Classifying passenger and non-passenger signals in public transportation by analysing mobile device wi-fi activity," *Journal of Information Processing*, vol. 27, pp. 25–32, 2019.
- [31] J. E. Häkegård, T. A. Myrvoll, and T. R. Skoglund, "Statistical modelling for estimation of od matrices for public transport using wi-fi and apc data," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1005–1010.
- [32] S. Ryu, B. B. Park, and S. El-Tawab, "Wifi sensing system for monitoring public transportation ridership: a case study," *KSCE Journal of Civil Engineering*, vol. 24, no. 10, pp. 3092–3104, 2020.
- [33] Y. Zhou, B. P. L. Lau, Z. Koh, C. Yuen, and B. K. K. Ng, "Understanding crowd behaviors in a social event by passive wifi sensing and data mining," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4442–4454, 2020.
- [34] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.
- [35] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.
- [36] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6910–6920, 2020.
- [37] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4634–4643.
- [38] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 1, pp. 154–163, 2018.
- [39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [40] S. Yu, Y. Li, G. Sheng, and J. Lv, "Research on short-term traffic flow forecasting based on knn and discrete event simulation," in *International Conference on Advanced Data Mining and Applications*. Springer, 2019, pp. 853–862.
- [41] A. Belhadi, Y. Djennouri, D. Djennouri, and J. C.-W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," *Applied Intelligence*, vol. 50, no. 10, pp. 3252–3265, 2020.
- [42] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–6.
- [43] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Multivariate time series forecasting via attention-based encoder-decoder framework," *Neurocomputing*, vol. 388, pp. 269–279, 2020.



Wenbo Chang received the B.E. degree in 2018 and the M.S. degree in 2022 from Inner Mongolia University, Hohhot, China, where he is currently working toward the Ph.D. degree in computer science. His research interests include passive WiFi sensing and mobile computing.



Baoqi Huang (Member, IEEE) received the B.E. degree in computer science from Inner Mongolia University (IMU), Hohhot, China, in 2002, the M.S. degree in computer science from Peking University, Beijing, China, in 2005, and the Ph.D. degree in information engineering from The Australian National University, Canberra, ACT, Australia, in 2012. He is currently a Professor with the College of Computer Science, IMU. His research interests include indoor localization and navigation, wireless sensor networks, and mobile computing. He was the recipient of the Chinese Government Award for Outstanding Chinese Students Abroad in 2011.



Bing Jia (Member, IEEE) received the Ph.D. degree from Jilin University, Changchun, China, in 2013. She is currently an Associate Professor with the College of Computer Science, Inner Mongolia University, Hohhot, China. Her current research interests include indoor localization, crowdsourcing, wireless sensor networks, and mobile computing.



Wuyungerile Li received the Ph.D. degree from Shizuoka University, Hamamatsu, Japan, in 2013. She is currently an Associate Professor with the School of Computer Science, Inner Mongolia University, Hohhot, China. Her research interests include wireless sensor networks and mobile computing. She was the recipient of some research grants from the National Science Foundation of China and Inner Mongolia as a Principal Investigator.



Gang Xu (Member, IEEE) received the M.S. degree in computer science from Harbin Institute of Technology, Harbin, China, in 2007, and the Ph.D. degree in computer science from Inner Mongolia University (IMU), Hohhot, China, in 2016. He is currently an Associate Professor with the College of Computer Science, IMU. His research interests include wireless sensor networks, DTN, and mobile computing.