BusWTE: Realtime Bus Waiting Time Estimation of GPS Missing via Multi-Task Learning

Yuecheng Rong[1,2] ✉, Jun Liu[1,3], Zhilin Xu[2], Jian Ding[2], Chuangming Zhang[2], and Jiaxiang Gao[2]

[1] College of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China
[2] Baidu Inc., Beijing, China
{rongyuecheng, xuzhilin01, dingjian01, zhangchuanming, gaojiaxiang}@baidu.com
[3] National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, Xi'an, China
liukeen@mail.xjtu.edu.cn

Abstract. Realtime bus waiting time is of great importance to the intelligent public transportation system and is beneficial for improving user satisfaction by online map services. While there are limited realtime bus waiting time services in a city, because of the expensive cost of GPS sensor deployment and realtime service operation. To address the above problem, we propose a novel end-to-end multi-task framework named BusWTE, which estimates bus waiting time for those bus routes without GPS sensors deployed. BusWTE utilizes a variety of urban datasets, including historical bus trip data reported by a limited number of GPS equipped buses, road network data, traffic condition data, and mobility data. Specifically, we firstly use a classical BiLSTM architecture to encode the sequence of bus route related features, and employ two fully-connected layers to embed the stop related features and temporal features, respectively. Then a temporal attention mechanism is proposed to capture the dynamic correlation between the route features and temporal features. Furthermore, we employ multi-task learning to estimate the bus waiting time and the bus interval simultaneously, which highly improves the model performance. Finally, extensive experiments conducted on two large-scale real-world datasets demonstrate the effectiveness of BusWTE. In addition, BusWTE has been deployed on Baidu Map app, servicing over twenty major cities in China.

Keywords: Bus Waiting Time · DNN · LSTM · Attention · Multi-Task.

1 Introduction

With the rapid expansion of public transportation network, bus navigation has become an essential service for urban residents. As a core function, effective realtime bus waiting time estimation can significantly improve user satisfaction and ultimately optimize the public transportation system [5].

Traditionally, the bus waiting time can be calculated by the estimated travel time and the collected bus realtime location. However, the realtime services with high coverage of bus routes are still limited [3], due to the cost of GPS sensor deployment and maintenance, and the dispersion of operators.

The average waiting time for passengers is considered as a measure of quality for the public transportation service [6,9]. Therefore, it is meaningful yet difficult to estimate the realtime bus waiting time for arbitrary bus stops without GPS sensors in the city. Specifically, the challenges of the above problem lie in two aspects. First, the result of waiting time estimation is affected by many complex factors, including traffic condition, spatial context and temporal dependencies. Existing headway-based methods deduce the static average waiting time as half of the departure interval, assuming that passengers arrive randomly at bus stops and passengers can be served by the earliest arriving bus [2]. However, the static estimation cannot meet the demand for realtime and highly accurate waiting time. Second, staged approaches estimate essential information (e.g., bus departure schedule) separately, which may introduce cumulative error. In practice, the bus schedule information has a great significance on waiting time estimation. However, it is very difficult to reduce the cumulative error while fully leveraging the bus schedule information.

Recent advances of location-acquisition and wireless communication technologies have resulted in massive spatial-temporal data, which provide great potentials to estimate the realtime information in metropolis [11, 13, 16, 17]. To tackle the above challenges, in this paper, we propose BusWTE, a novel end-to-end multi-task framework to estimate bus waiting time for those bus routes without GPS sensors using a variety of urban datasets (e.g., traffic condition data, road network data and mobility data). Specifically, we firstly use a classical BiLSTM architecture to encode the sequence of bus route features and employ two fully-connected layers to embed the stop features and temporal features, respectively. Then, we propose a temporal attention mechanism to capture the dynamic correlation between the bus route features and temporal features. Finally, we employ multi-task learning to estimate the bus waiting time and the bus interval simultaneously, which obviously improves the performance.

To verify the effectiveness of the proposed framework BusWTE, we conduct extensive experiments on two large-scale real-world datasets collected from Baidu Maps. The experimental results demonstrate that BusWTE significantly outperforms the baseline approaches in terms of multiple metrics. In addition, it has already been deployed on Baidu Maps which is one of the world's largest online map services, serving over twenty major cities in China. Figure 1 shows an illustrative example of bus waiting time estimation service on Baidu Maps.

In summary, our main contributions are as follows:

- To the best of our knowledge, we present the first attempt to formally study the problem of estimating waiting time for those bus routes without GPS sensors, in a realtime fashion.

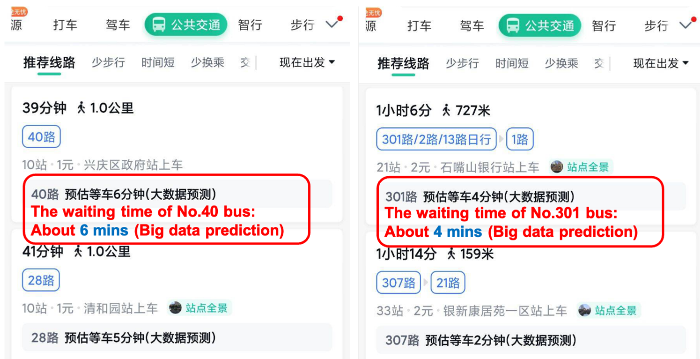


Fig. 1. The bus waiting time application of BusWTE on Baidu Maps. The figure illustrates Baidu Maps provide realtime waiting time estimation service for No.40 and No.301 bus without GPS information.

- We propose an end-to-end multi-task framework that learns to estimate the bus waiting time and the bus interval simultaneously, which reduces the cumulative error caused by staged estimation.
- We extract discriminative spatial-temporal related features contributing to our model. Moreover, we design a temporal attention mechanism to adaptively model the dynamic correlation between the bus route features and temporal features, therefore, leading to a high estimation accuracy.
- We conduct extensive experiments on two real-world urban-scale datasets, which demonstrate the effectiveness of BusWTE and its components and features. The successful deployment of BusWTE at Baidu Maps further shows that it is a large-scale practical solution for real-world bus waiting time estimation services.

The rest of this paper is organized as follows. In Section 2, we discuss the related work of the proposed approach. Section 3 presents the definitions and problem statement. We elaborate on the detailed methodologies of BusWTE in Section 4. Experimental results are presented in Section 5. Finally, we conclude this paper and suggest future work in Section 6.

2 Related Work

In this section, we mainly discuss the relevant work of bus waiting time estimation. In addition, we also discuss the related work of estimating the realtime information for those entities missing hardware sensors by fusing multi-source spatial-temporal data.

2.1 Bus Waiting Time Estimation

Reliable and realtime waiting time of the bus can help passengers plan their trips better, which would be an effective way to improve the service of public transportation systems. Bus waiting time estimation methods can be organized into the following two categories:

Realtime location based methods: Realtime location based methods acquire the vehicle realtime location using the hardware devices like GPS and then calculate the waiting time by estimating the travel time from the realtime location to the waiting stop. The realtime location-based methods rely on the bus location information, which can be collected by GPS devices or other available sensing resources, including cell tower signals, movement statuses, audio recordings, etc [18]. However, the realtime bus information is limited due to the expensive cost of GPS sensor deployment and maintenance.

Headway based methods: Headway based methods deduce the static average waiting time using the headway distributions through some assumptions, such as passengers arrive randomly at bus stops and passengers can be served by the earliest arriving bus [2]. Under the abovementioned assumptions, the average waiting time is half that of the departure interval. However, sometimes the assumption of regular service cannot be completely reliable and some methods have been proposed to address cases where some degree of irregularity is involved in bus arrivals [1, 2]. The static average waiting time is not always applicable, because the punctuality and regularity of bus travel may be heavily affected by traffic and other external fluctuations, which directly impacts the waiting time [10].

It is extremely valuable but hard to estimate the realtime waiting time without directly tracking the bus in real time and timetable information, which is even considered infeasible [3]. However, compared to existing approaches, we propose an end-to-end multi-task learning framework to estimate bus realtime waiting time for those bus routes without GPS sensors.

2.2 Spatial-temporal Data Estimation

Due to the cost or data constraints, it is a very critical issue to estimate the realtime information by spatial-temporal data without hardware sensors, such as air quality inference [17] and parking difficulty estimation [11, 16].

Recently, deep learning techniques have enjoyed considerable success due to their powerful hierarchical feature learning ability in spatial-temporal data estimation [13]. U-Air [17] incorporates a neural network into the co-training framework to inference air quality for any location based on the air pollutant of some monitoring stations and a variety of urban datasets. SHARE [16] employs a semi-supervised hierarchical recurrent graph neural network to predict parking availability for the parking lots without parking sensors, based on historical data reported by a limited number of existing sensors and a variety of datasets observed in the city.

Compared with the prediction tasks of missing sensors at fixed positions, it is more complex to estimate bus realtime waiting time when GPS information is completely missing.

3 Preliminaries

We first introduce some important definitions and formally define the bus waiting time estimation problem.

Definition 1: *Bus Waiting Time.* Consider a set of bus routes $R = R_l \cup R_u = \{r_1, r_2, ..., r_L\}$, where L is the total number of bus routes, R_l and R_u denote a set of bus routes with and without position sensors, respectively. Given current time t, the route $r \in R$, the k-th stop $stop_{rk}$ of the route r, the earliest bus b arrival time $arrtime^b_{rk}$ at $stop_{rk}$ since t, the bus waiting time can be given by $arrtime^b_{rk} - t$.

Let $X_{rkt} \in \mathbb{R}^M$ and $Y_{rkt} \in \mathbb{R}$ denote observed M dimensional feature vectors and bus waiting time for the stop $stop_{rk}$ at time t, respectively.

Definition 2: *Bus Departure Interval.* Bus departure interval is the duration between the departure times of two adjacent buses of the same route. In this paper, we assume that bus departure interval is constant in each time period (e.g., an hour), but may vary in different time periods.

Let $Y^{val}_r = (y^{val}_{r1}, y^{val}_{r2}, ..., y^{val}_{rT}) \in \mathbb{R}^T$ denote the bus departure interval for bus route $r \in R$ at T time intervals in one day.

Problem: *Bus Waiting Time Estimation.* Suppose we have the feature vector set for all bus routes $X_R \subset \mathbb{R}^M$, partially bus waiting times $Y_{R_l} \subset \mathbb{R}$ and partially bus intervals $Y^{val}_{R_l} \subset \mathbb{R}^T$. We aim to estimate the bus waiting time with the given current time t, and the bus stop $stop_{rk}$ of the bus route $r \in R_u$.

4 BusWTE

As shown in Figure 2, our framework consists of two major parts, feature extraction and waiting time estimation model. We extract discriminative features from the crowdsourcing data, mobility data and transportation network data of Baidu Maps. See Section 4.1 for details. The waiting time estimation model is designed as an end-to-end multi-task learning network, as detailed in Section 4.2.

4.1 Feature Extraction

We introduce the process of constructing and transforming feature vectors below. Table 1 lists the features we construct with a detailed description.

Bus Route Features The bus route departure interval has a great influence on the waiting time at the bus stops. In the case of regular bus services, the average waiting time of a bus stop in a time interval is close to half of the departure interval, assuming that the traffic condition remains stable and the time for passengers to arrive at the stop is random [2]. For the route feature extraction, we pay more attention to the bus route departure interval features. Our insight into the departure interval features is that the departure intervals of a bus route

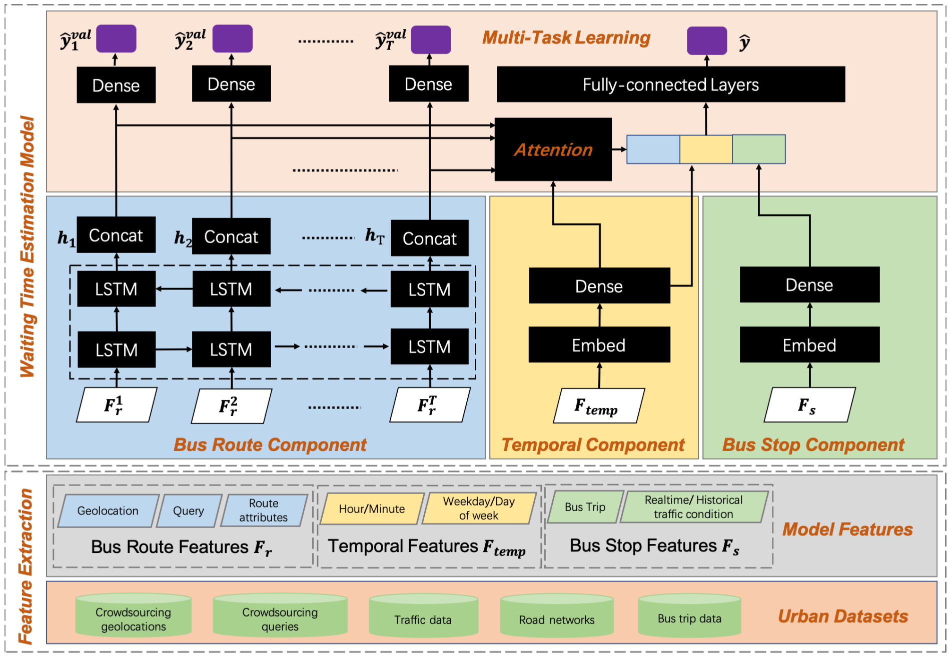


Fig. 2. The framework of BusWTE.

Table 1. The Description of Features.

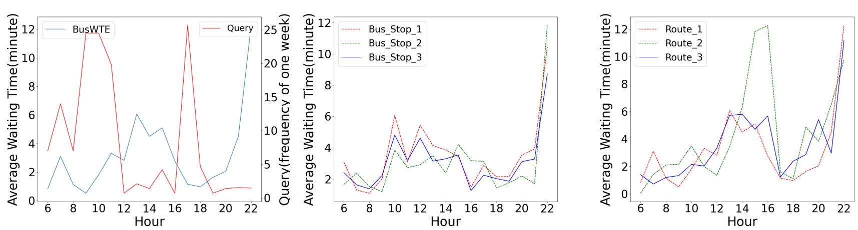
Feature Type	Feature	Description
Bus Route (F_r)	Geolocation (F_{rg})	The popularity of visitors located in the region of a bus route in a time interval
	Query (F_{rq})	The popularity of a bus route search queries in a time interval
	Static attributes (F_{ra})	The length and the number of bus stops in the bus route on the road network
Bus Stop (F_s)	Bus Trip (F_{strip})	The length and the number of bus stops in the bus trip from the first stop to the corresponding stop on the road network
	Realtime traffic condition (F_{srtc})	The total current traffic travel time for each road segment in the bus trip
	Historical traffic condition (F_{shtc})	The total Historical traffic travel time for each road segment in the bus trip
Temporal (F_{temp})	Minite (F_{tm})	The corresponding time period in a hour
	Hour (F_{th})	The corresponding time period in a day
	Day of week (F_{td})	The ordinal number of the day in a week
	Workday (F_{tw})	Whether the day is a workday

must match the actual travel demand, which can be represented by the human mobility data in the city, such as crowdsourcing map queries.

Bus route features F_r are comprised of three features: geolocation feature F_{rg}, query feature F_{rq} and route static attributes F_{ra}. F_{rg} is the frequency of visitors located in each bus stop region of a bus route in a time interval, which presents human mobility of the areas crossed by the bus route. F_{rq} is the popularity of bus route queries representing the demand of passengers to take the bus on this route. Figure 3(a) shows a strong correlation between the query feature and bus average waiting time for the same bus route. F_{ra} includes the total length and the total number of stops in each bus route, which are considered in the design of the bus route departure interval.

Bus Stop Features The travel time from the first stop to the waiting stop has a great significance on the bus waiting time assuming that the departure interval is known in advance. For the stop feature extraction, we focus on the bus travel time from the first stop to each stop. Figure 3(b) shows that there is a very significant difference in the average waiting time distributions between different bus stops on the same route.

Bus stop features F_s are comprised of bus trip feature F_{strip}, realtime traffic condition feature F_{srtc} and historical traffic condition feature F_{shtc}. We use the distance and the stop number of the bus trip as bus trip feature F_{strip}. The bus travel time is highly correlated with the route that bus travels through and the bus stops for the bus trip. We use the realtime traffic travel time as F_{srtc} and the historical average traffic travel time as F_{shtc} to capture the realtime and historical pattern of traffic conditions, respectively.



(a) The daily distributions of query feature and average waiting time for the same bus route.

(b) The average waiting time distributions of three stops in the same route.

(c) The daily average waiting time distributions of three bus routes.

Fig. 3. The correlation between features and bus waiting time on Xiamen City.

Temporal Features The waiting time of one bus stop could be affected by lots of temporal information. The start waiting time is one of the most important factors. Figure 3(c) shows the strong correlation between time (hour in day) and bus average waiting time. In fact, the average waiting time of the bus stop changes periodically as long as time. We exploit hour of day F_{th}, minute of hour F_{tm}, day of week F_{td} and weekday F_{tw} as the temporal features F_{temp} to estimate bus waiting time.

4.2 Waiting Time Estimation Model

Figure 2 shows the high-level overview of the proposed model, which is comprised of three major components, modeling bus route scheduling patterns, bus stop spatial-temporal information and general temporal factors, respectively.

The route features in each time interval are fed into the route component, which uses the classical BiLSTM to model the temporal dependencies among features at different time intervals. In the stop component and temporal component, the features are fed into a two-layer fully-connected neural network, respectively. Then we propose a temporal attention mechanism to capture the dynamic correlation between the latent representations of the bus route component and temporal component. The outputs of the temporal attention, stop component and temporal component, are concatenated and fed through the fully-connected layer to output the bus waiting time result. Finally, we employ a multi-task mechanism to estimate the bus intervals and the bus waiting time simultaneously, capable of leveraging the operation patterns of bus routes.

Bus Route Component In this paper, we denote the bus route interval feature at time interval t as $F_r^t = (F_{rg}^t, F_{rq}^t, F_{ra}^t)$. We employ the Bidirectional Long-Short Term Memory (BiLSTM) architecture to encode the sequence of departure features, generating the latent vector representation for each time step feature.

The bus departure interval continuously changes over time, companing the fluctuation of temporal factors that affect it. Intuitively, the previous interval features may influence on the current departure interval, which can be effectively handled by the recurrent neural network (RNN) [12].

Therefore, we employ BiLSTM in the proposed model which can be trained using all the available input temporally-related information from two directions to improve the estimation performance.

A BiLSTM consists of a forward and backward LSTM. The forward $\overrightarrow{f}$ reads the input interval temporally-related feature sequence from F_r^1 to F_r^T and outputs a sequence of forward hidden states $(\overrightarrow{\boldsymbol{h}}_1, \overrightarrow{\boldsymbol{h}}_2, ..., \overrightarrow{\boldsymbol{h}}_T)$. The backward LSTM $\overleftarrow{f}$ reads the input feature sequence in the reverse order, i.e., from F_r^T to F_r^1, resulting in a sequence of backward hidden states $(\overleftarrow{\boldsymbol{h}}_1, \overleftarrow{\boldsymbol{h}}_2, ..., \overleftarrow{\boldsymbol{h}}_T)$.

We concatenate the forward hidden state $\overrightarrow{\boldsymbol{h}}_t$ and the backward one $\overleftarrow{\boldsymbol{h}}_t$, which becomes the final latent vector representation as $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$.

Bus Stop Component We use a neural network model, which can effectively capture the relationship among different information, to represent bus stop features. In this paper, the bus stop features is denoted as $F_s = (F_{strip}, F_{srtc}, F_{shtc})$.

F_{strip}, F_{srtc} and F_{shtc} are first fed into the embedding layer followed by an activation, respectively. Then we concatenate the output of each sub embedding layer as $\boldsymbol{H}_{se}$, followed by fully-connected layer as:

$$\boldsymbol{H}_{ostop} = ReLU\left(\boldsymbol{W}_{hse}\boldsymbol{H}_{se} + \boldsymbol{b}_{hse}\right), \tag{1}$$

where $\boldsymbol{W}_{hse}$ and $\boldsymbol{b}_{hse}$ are the parameters to be learned.

Temporal Component Temporal information is essential for bus waiting time estimation. We use fully-connected neural network component to represent temporal information. In this paper, the temporal features are denoted as $F_{temp} = (F_{th}, F_{tm}, F_{td}, F_{tw})$. Then F_{th}, F_{tm}, F_{td} and F_{tw} are first fed into the embedding layer which is followed by an activation, respectively. Then we concatenate the output of each sub embedding layer as $\boldsymbol{H}_{te}$, which is followed by fully-connected layer as:

$$\boldsymbol{H}_{otemp} = ReLU\left(\boldsymbol{W}_{hte}\boldsymbol{H}_{te} + \boldsymbol{b}_{hte}\right), \tag{2}$$

where $\boldsymbol{W}_{hte}$ and $\boldsymbol{b}_{hte}$ are the parameters to be learned.

Temporal Attention Mechanism We employ an attention mechanism to adaptively model the dynamic correlation between the bus route interval features and temporal features. We introduce a temporal attention setting to compute the attention vector for the temporal hidden representation $\boldsymbol{H}_{otemp}$. In this setting, the temporal hidden representation $\boldsymbol{H}_{otemp}$ is taken as the query of the attention mechanism. The bus route feature hidden states $\boldsymbol{h}_t (t \in [1, T])$ are taken as the *keys* and *values* of the attention mechanism. To be specific, the attention mechanism is formulated as:

$$\boldsymbol{Q} = \boldsymbol{H}_{otemp}, \tag{3}$$

$$\boldsymbol{K}_t = \boldsymbol{h}_t, \tag{4}$$

$$\boldsymbol{V}_t = \boldsymbol{h}_t, \tag{5}$$

$$\boldsymbol{f}(\boldsymbol{Q}, \boldsymbol{K}_t) = \frac{\boldsymbol{Q}^\top \cdot \boldsymbol{K}_t}{\sqrt{d^{(H)}}}, \tag{6}$$

$$\alpha(\boldsymbol{Q}, \boldsymbol{K}_t) = \frac{exp(f(\boldsymbol{Q}, \boldsymbol{K}_t))}{\sum_{t'} exp(f(\boldsymbol{Q}, \boldsymbol{K}_{t'}))}, \tag{7}$$

$$\boldsymbol{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \sum_{t'} \alpha(\boldsymbol{Q}, \boldsymbol{K}_{t'})\, \boldsymbol{V}_{t'}, \tag{8}$$

where the $d^{(H)}$ denotes the hidden size of the *keys* and *values*. Then, the dynamic correlation between the bus route interval features and temporal features can be encoded as $\boldsymbol{H}_{oatt} = \boldsymbol{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ by Equation (8).

Multitask Learning Intuitively, the departure interval changes over time and has a great significance on waiting time estimation. Staged approaches estimate bus departure schedule separately, which may introduce cumulative error. Therefore, we designed a multi-task structure to reduce the cumulative error while fully leveraging the bus interval information. More specifically, we estimate bus waiting time and each departure interval of a bus route using a sequence model simultaneously, which is similar to the sequence labeling task in natural language processing (NLP).

Based on the bus route component, we have latent representations of the bus route feature $\boldsymbol{h}_t (t \in [1, T])$ at time interval t. $\boldsymbol{h}_t$ is fed through the fully-connected layer to output new hidden state, defined as:

$$\boldsymbol{h}_{to} = ReLU \left(\boldsymbol{W}_{ht} \boldsymbol{h}_t + \boldsymbol{b}_{ht} \right). \tag{9}$$

Then, $\boldsymbol{h}_o$ is fed through the fully-connected layer to output the bus departure interval result $\hat{\boldsymbol{y}}_t^{val}$, defined as:

$$\hat{\boldsymbol{y}}_t^{val} = \boldsymbol{V}_{hto}^{\top} \boldsymbol{h}_{to} + \boldsymbol{b}_{hto}, \tag{10}$$

where the $\boldsymbol{W}_{ht}$, $\boldsymbol{b}_{ht}$, $\boldsymbol{V}_{hto}$ and $\boldsymbol{b}_{hto}$ are the parameters to be learned. Finally, we use a linear transformation to generate the final output result.

Training and Optimization Based on the above components, we concatenate all the latent representation layers $\boldsymbol{H}_{oatt}$, $\boldsymbol{H}_{ostop}$ and $\boldsymbol{H}_{otemp}$ as $\boldsymbol{H}_f$, which is then fed into the fully-connected layer to output the bus waiting time result, defined as:

$$\hat{\boldsymbol{y}} = \boldsymbol{V}_{hf}^{\top} \boldsymbol{H}_f + \boldsymbol{b}_{hf}, \tag{11}$$

where the $\boldsymbol{V}_{hf}$ and $\boldsymbol{b}_{hf}$ are the parameters to be learned. Finally, we use a linear transformation to generate the final output result.

Our proposed model aims to minimize the mean squared error (MSE) between the ground truth bus waiting time $\boldsymbol{y}$ and the estimated bus waiting time $\hat{\boldsymbol{y}}$:

$$L_1 = \| \boldsymbol{y} - \hat{\boldsymbol{y}} \|_2^2. \tag{12}$$

In addition, the bus interval estimation auxiliary task aims to minimize the mean squared error (MSE) between the ground truth departure interval $\boldsymbol{y}_t^{val}$ and the estimated departure interval $\hat{\boldsymbol{y}}_t^{val}$, defined as:

$$L_2 = \sum_{t=1}^{T} \| \boldsymbol{y}_t^{val} - \hat{\boldsymbol{y}}_t^{val} \|_2^2. \tag{13}$$

By considering the MSE loss and auxiliary task loss, our model aims to jointly minimize the following objective:

$$L(\theta) = L_1 + \lambda L_2, \tag{14}$$

where θ are all learnable parameters in our model, λ is the hyper-parameter controls the importance of the auxiliary task loss.

5 Experiments

A set of experiments are employed in this section to measure the performance of BusWTE and verify the effectiveness of each component in BusWTE. All of our approaches are deployed on Baidu PaddlePaddle deep learning platform [15].

5.1 Datasets

In the evaluation, we use the following 2 real datasets in the experiments. Table 2 shows the statistical details of the datasets.

Bus trip data: Two datasets are used to evaluate our solutions of this problem. Both of them are acquired from Baidu maps, from December 1st 2021 to December 28th 2021.

Mobility data: We also employ sampled geolocation data and map query data from crowdsourcing data of Baidu Maps.

Traffic data: The realtime traffic data and historical traffic data are also from Baidu Maps.

Road network data: The public transportation network containing the geolocation information of bus routes and stops, is acquired from Baidu Maps.

Table 2. Detail of Dataset.

Data description		Xiamen City	Nanjing City
bus trip data	bus trip records	4,042,772	3,919,149
road network data	bus routes	665	870
	bus stops	16,802	19,096
mobility data	crowdsourcing queries	334,241	402,118
	crowdsourcing geolocations	19,917,326	27,915,180

The ground truth of bus waiting time, bus departure interval and bus travel time are all produced by the bus trip datasets mentioned above. We use the data from December 1st 2021 to December 21th 2021 for training, and the data from December 22th 2021 to December 28th 2021 is used for testing. We also guarantee that the bus routes of test dataset are not in the training dataset.

5.2 Experimental Settings

Evaluation Metrics We use three metrics including root mean square error (RMSE) [7], mean absolute error (MAE) [14] and mean absolute percentage error (MAPE) [8] to evaluate all tasks (e.g., bus waiting time estimation and bus interval estimation). We use second as the unit of bus waiting time. For the above-mentioned evaluation metrics, a smaller evaluation metric value means better performance in the following experiments.

Implementation details The time period of bus departure interval is set to an hour, which indicates the total number T of time intervals is 24. The super parameter λ in Equation (14) is set to 0.35 in our multi-task learning model. The number of hidden states in BiLSTM layer is 64. Each of the two layers has 64 neurons in the stop component the same as the temporal component. The hidden state size of the output layer is 64 in both bus waiting time task and bus interval task. To optimize the model, we choose Adam as the optimizer and set the learning rate to 0.001. Each of the two layers has 128 neurons in the DNN model. We also employ Adam as the optimizer of the DNN model, and the learning rate is set to 0.0006. We choose ReLU as the activation function of all the hidden layers.

5.3 Baselines And Variants

Baselines We compare our proposed BusWTE with following approaches:

Historical Average(HA): The historical average waiting time of all the bus stops, covered by bus trip datasets in a time interval in the city.

Waiting Time Based Interval(WTBI): In the case of regular bus services, the average waiting time of passengers is estimated assuming that passengers arrive randomly at bus stops and passengers can be served by the earliest arriving bus [2], and is given by: $E(W) = \frac{1}{2}H$, where H is bus departure interval.

Waiting Time Based Pipeline(WTBP): We also compared BusWTE with pipeline based method, which estimates bus waiting time based on approximating the realtime locations of buses, using the estimated bus intervals and realtime travel times.

Linear Regression(LR): Linear regression is widely used to model the relationship of multiple independent variable and single dependent variable [19].

Gradient Boosting Decision Tree(GBDT): GBDT is well-known for its outstanding performance and efficiency. The XGBoost (eXtreme Gradient Boosting) is an open source gradient boosting library which also provides an optimized distributed version [4].

Deep Neural Network(DNN): We also use a two fully-connected layers neural network with ReLU activation to estimate the bus waiting time.

Variants. To evaluate each component of our proposed model, we also compare it with different variants of BusWTE:

BusWTE-noRoute: BusWTE-noRoute removes the bus route component.

BusWTE-noStop: BusWTE-noStop removes the bus stop component.

BusWTE-noTemp: BusWTE-noTemp removes the temporal component.

BusWTE-noAttn: BusWTE-noAttn removes the attention mechanism.

BusWTE-noMul: BusWTE-noMul removes the multil-task mechanism.

5.4 Overall Performance

A set of experiments compares the performance of BusWTE and several baseline methods. Table 3 shows the experimental results. From the results, we have the following observations:

(1) From Table 3, we can see that GBDT algorithm outperforms Linear Interpolation. Both Xiamen City and Nanjing City present good performance on the MAE and MAPE. Thus our features are general and robust for different cities.

(2) BusWTE significantly outperforms other methods on the two datasets. HA is a simple baseline and works the worst. The main reasons for such improvement lie in two aspects. First, we introduce several feature components and the attention mechanism to extract more useful spatial-temporal information from the designed features. Second, we propose an end-to-end multi-task network to estimate bus waiting time and bus interval simultaneously, which is able to reduce the cumulative error compared with pipeline based methods.

Table 3. Performance of BusWTE and Baseline Methods.

Methods	Xiamen City			Nanjing City		
	MAPE	MAE (sec)	RMSE (sec)	MAPE	MAE (sec)	RMSE (sec)
HA	93.9%	359.81	450.86	98.0%	328.58	407.24
LR	70.6%	231.81	286.23	71.3%	202.45	254.08
GBDT	66.1%	227.80	280.19	70.9%	200.36	252.63
DNN	54.8%	228.10	300.36	63.3%	198.46	257.01
WTBT	72.1%	231.60	288.77	75.7%	230.46	312.72
WTBP	67.0%	232.00	325.67	72.1%	228.01	348.91
BusWTE	**52.4%**	**220.28**	**276.82**	**50.2%**	**196.47**	**252.30**

5.5 Ablation study

In this section, we conduct ablation studies on BusWTE, including model ablation and feature ablation, to further verify the effectiveness of each component. The experiments are finished for three metrics on both Xiamen City and Nanjing City datasets. Table 4 shows the experimental results of ablation study.

Feature Ablation To examine the performance impact of feature components, we evaluate BusWTE with complete features and its three variants: BusWTE-noRoute, BusWTE-noStop and BusWTE-noTemp.

Effectiveness of the Route Component: We evaluate the relevance of the route component by removing all the route features. Table 4 shows that the MAE and MAPE of BusWTE-noRoute declines significantly compared with BusWTE. The contribution of the route component is significant. The main reason is bus route interval information has a great influence on the result of waiting time

Table 4. Performance of BusWTE and Variants.

Methods	Xiamen City			Nanjing City		
	MAPE	MAE (sec)	RMSE (sec)	MAPE	MAE (sec)	RMSE (sec)
BusWTE-noRoute	61.7%	233.34	298.12	71.2%	204.46	254.08
BusWTE-noStop	59.0%	235.43	298.62	62.2%	209.31	272.31
BusWTE-noTemp	56.1%	231.47	299.05	58.2%	198.01	260.05
BusWTE-noAttn	57.8%	226.05	293.07	55.2%	196.49	272.31
BusWTE-noMul	54.9%	228.29	300.36	63.3%	198.46	257.01
BusWTE	**52.4%**	**220.28**	**276.82**	**50.2%**	**196.47**	**252.30**

estimation. In fact, in the case of regular bus services, the average waiting time of passengers is estimated by the bus departure interval [2]. As can be seen in Table 4, the results of WTBI and WTBP are also effective, which indicates that the route interval information is always beneficial for this problem.

Effectiveness of the Stop Component: We also evaluate the relevance of the stop component by removing all the stop features. As shown in Table 4, the results of BusWTE-noStop drops significantly compared with BusWTE. The contribution of stop component is important, i.e., the MAPE of Xiamen City and Nanjing City increases 12% and 23% respectively, after removing the stop component. Intuitively, there is a significant difference in average waiting time distribution between different bus stops, which is caused by the spatial-temporal factors conditions along bus route, such as dynamic traffic conditions.

Effectiveness of the Temporal Component: We also evaluate the relevance of the stop component by removing all the temporal features. Table 4 shows that the results of BusWTE-noTemp drops obviously compared with BusWTE. The contribution of temporal component is also important, i.e., the MAE and MAPE of Xiamen City and Nanjing City increase to a certain extent, after removing the temporal component. Therefore, temporal information is critical for waiting time estimation.

Model Ablation We evaluate the performance of BusWTE and it's two variants, which are BusWTE-noAttn and BusWTE-noMul.

Effectiveness of the Attention Mechanism: We remove the attention mechanism from BusWTE to test its contribution. As illustrated in Table 4, the results of BusWTE-noAttn falls obviously compared with BusWTE. Particularly, the MAPE of Xiamen City and Nanjing City increases 9% and 10% respectively, after removing the attention mechanism. A possible reason is that temporal attention mechanism can effectively capture the realtime bus routes departure information.

Effectiveness of the Multil-Task Mechanism To evaluate the importance of multil-task mechanism, we compare BusWTE-noMul to BusWTE. As can be seen from Table 4, when the multil-task mechanism is removed, the performance declines significantly. Especially, the MAE, MAPE and RMSE of Xiamen City and Nanjing City increase in varying degrees, after removing the multil-Task mechanism. This is because the multitask mechanism can reduce cumulative error and fully leverage the valuable information of bus route interval, which is beneficial for bus waiting time estimation.

5.6 Application and Deployment

We applied BusWTE to provide the realtime bus waiting time service in Baidu Maps, in more than twenty major cities in China. We build online service based on BRPC (https://github.com/brpc/brpc), a scalable RPC framework used throughout Baidu. We can acquire the waiting time query information such as the query route, stop and current time from Baidu map app. First, we retrieve the route related features, stop related features, and temporal features from database, which are extracted in advance. Then all above features are fused into a single feature vector for the bus stop at current time. Finally, the bus waiting time is estimated by the trained model and the online service sends the estimated travel time to Baidu map app.

Table 5 presents the online efficiency of our approach, which was tested on a 64-bit server with 8-core 2.4G CPU, 64 GB RAM and NVIDIA A100 GPU. The feature processing accounts for up to more than 90% of the total online processing time.

Table 5. Efficiency study

Procedures		Time (ms)
Feature processing (per query)	Online process	6.37
Inference (per query)	BusWTE-CPU	0.76
	BusWTE-GPU	0.24

6 Conclusion

We propose BusWTE, an end-to-end multi-task model to estimate bus waiting time for those bus routes without GPS sensors. BusWTE utilizes historical bus trip data reported by a few existing buses with GPS sensors and various datasets, such as traffic condition data, map mobility data and road network data. Then we propose a temporal attention mechanism to capture the dynamic correlation between the bus route features and temporal features. Furthermore, we employ multi-task learning to estimate the bus waiting time and the bus interval simultaneously, which reduces the cumulative error caused by staged estimation. Experimental results on two real-world datasets prove the effectiveness of BusWTE. We applied it to provide the realtime bus waiting time services on Baidu Maps, serving over 20 major cities in China. In the future, we will try to model the dynamic temporal autocorrelation inside of and between bus routes (stops) with and without GPS sensors, to improve the estimation performance.

Acknowledgment

This work was supported by National Key Research and Development Program of China (2020AAA0108800), National Natural Science Foundation of China (62137002, 61721002), Innovation Research Team of Ministry of Education (IRT_17R86), and Project of China Knowledge Centre for Engineering Science and Technology.

References

1. Amin-Naseri, M.R., Baradaran, V.: Accurate estimation of average waiting time in public transportation systems. Transportation Science **49**, 213–222 (08 2014)
2. Ansari Esfeh, M., Wirasinghe, S., Saidi, S., Kattan, L.: Waiting time and headway modelling for urban transit systems - a critical review and proposed approach. Transport Reviews (08 2020)
3. Barnes, R., Buthpitiya, S., Cook, J., Fabrikant, A., Tomkins, A., Xu, F.: Bustr: Predicting bus travel times from real-time traffic pp. 3243–3251 (2020)
4. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y.: xgboost: Extreme gradient boosting (2015)
5. Chu, K., Lam, A.Y.S., Loo, B.P.Y., Li, V.O.K.: Public transport waiting time estimation using semi-supervised graph convolutional networks. In: ITSC (2019)
6. Hsu, S.: Determinants of passenger transfer waiting time at multi-modal connecting stations. Transportation Research Part E **46**, 404–413 (05 2010)
7. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting **22**(4), 679–688 (2006)
8. de Myttenaere, A., Golden, B., Le Grand, B., Rossi, F.: Mean absolute percentage error for regression models. Neurocomputing **192**, 38–48 (2016)
9. Politis, I., Papaioannou, P., Basbas, S., Dimitriadis, N.: Evaluation of a bus passenger information system from the users' point of view in the city of thessaloniki, greece. Research in Transportation Economics **29**, 249–255 (12 2010)
10. Ramli, M.A., Jayaraman, V., Chee, K., Heong, T., Khoon, G., Monterola, C.: Improved estimation of commuter waiting times using headway and commuter boarding information. Physica A **501** (12 2017)
11. Rong, Y., Xu, Z., Yan, R., Ma, X.: Du-parking: Spatio-temporal big data tells you realtime parking availability. In: SIGKDD. ACM (2018)
12. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NeurIPS (2014)
13. Wang, S., Cao, J., Yu, P.: Deep learning for spatio-temporal data mining: A survey. IEEE Transactions on Knowledge and Data Engineering (2020)
14. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate Research **30**(1), 79–82 (2005)
15. Yanjun Ma, Dianhai Yu, T.W.H.W.: Paddlepaddle: An open-source deep learning platform from industrial practice. Frontiers of Data and Domputing **1**(1) (2019)
16. Zhang, W., Liu, H., Liu, Y., Zhou, J., Xiong, H.: Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. Proceedings of the AAAI Conference on Artificial Intelligence **34**, 1186–1193 (04 2020)
17. Zheng, Y., Liu, F., Hsieh, H.P.: U-air: When urban air quality inference meets big data. In: SIGKDD. ACM (2013)

18. Zhou, P., Zheng, Y.: How long to wait? predicting bus arrival time with mobile phone based participatory sensing. TMC **13** (06 2012)
19. Zou, K.H., Tuncali, K., Silverman, S.G.: Correlation and simple linear regression. Radiology **227**(3), 617–628 (2003), pMID: 12773666