Learning Heterogeneous Traffic Patterns for Travel Time Prediction of Bus Journeys

Peilan He, Guiyuan Jiang, Siew-Kei Lam, Yidan Sun

School of Computer Science and Engineering,
Nanyang Technological University, 639798 Singapore

Abstract

In this paper, we address the problem of travel time prediction of bus journeys which consist of bus riding times (may involve multiple bus services) and also the waiting times at transfer points. We propose a novel method called Traffic Pattern centric Segment Coalescing Framework (TP-SCF) that relies on learned disparate patterns of traffic conditions across different bus line segments for bus journey travel time prediction. Specifically, the proposed method consists of a training and a prediction stage. In the training stage, the bus lines are partitioned into bus line segments and the common travel time patterns of segments from different bus lines are explored using Non-negative Matrix Factorization (NMF). Bus line segments with similar patterns are classified into the same cluster. The clusters are then coalesced in order to extract data records for model training and bus journey time prediction. A separate Long Short Term Memory (LSTM) based model is trained for each cluster to predict the bus travel time under various traffic conditions. During prediction, a given bus journey is partitioned into the riding time components and waiting time components. The riding time components are predicted using the corresponding LSTM models of the clusters while the waiting time components are estimated based on historical bus arrival time records. We evaluated our method on large scale real-world bus travel data involving 30 bus services, and the results show

Email addresses: phe002@e.ntu.edu.sg (Peilan He), gyjiang@ntu.edu.sg (Guiyuan Jiang), siewkei_lam@pmail.ntu.edu.sg (Siew-Kei Lam), ysun014@e.ntu.edu.sg (Yidan Sun)

that the proposed method notably outperforms the state-of-the-art approaches for all the scenarios considered.
Keywords: Journey time prediction, bus journey, riding time, waiting time, traffic pattern.

1. Introduction

Efficient and easy-to-use public transportation system is an important element in sustainable cities as it can boost the reduction in traffic congestion and lower carbon emissions from vehicles [2]. A key enabler to the success of public transportation system lies in the provision of accurate travel time information for travelers to make reliable journey planning. This is especially vital for bus services which typically account for the majority ridership among all public transport journeys [13]. Travel time prediction is also elementary to dynamic route guidance systems that provide intermodal transport options and recommended routes to travellers based on real-time data.

Accurately predicting travel time is challenging as it is not only affected by environmental and periodical factors such as weather conditions (e.g., a strong storm), time of day (e.g., rush hours) and holidays, but also by many complex dynamics such as the dynamic traffic conditions, uncertainty of the driving behavior, the fluctuations in travel demand and supply, stochastic arrivals and departures at signalized intersections, etc. Moreover, the traffic conditions of the road segments usually follow periodic patterns (e.g. daily, weekly periodicity) which change over time and vary geographically [24]. These characteristics make travel time in urban area intrinsically uncertain and difficult to predict.

Existing methods typically estimate the total journey time of a path by decomposing it into a sequence of individual road segments or path segments (each segment consists of a collection of individual segments). The *individual-based* approaches estimate the travel time of each road segment individually and aggregate the travel time for each road segment to compute the total travel time [6], while the *collective-based* methods treat a sequence of road segments (or the

entire path) as the basic prediction elements [26]. In general, the collective-based approaches perform better than individual-based ones, as they can better characterize the complex traffic conditions within the entire path and eliminate some errors accumulated by those individual-based approaches [6, 10, 34].

The factors that affect individual passengers' journey time using bus services (JT-BS) are significantly different from those affecting the travel time of general vehicles. Specifically, JT-BS is affected not only by the traffic conditions (e.g. traffic flow, traffic speed) but also by other factors such as the dwell time at each bus stop, bus service frequency, etc. Also, the waiting time at interchange station during transfer needs to be considered, as this is a non-negligible part of a passengers' total journey time. The waiting time may fluctuate drastically due to dynamic traffic demand and supply. These unique features make travel time prediction of bus journeys significantly different from travel time prediction of general vehicles in the existing works. As such, there is a need to develop efficient methods for travel time prediction which can cater to the unique characteristics of bus services.

The existing works on bus travel time prediction [1, 5, 14, 19] typically employ a single prediction model for the entire bus line (or bus service route). This could lead to unreliable prediction results as a bus usually travels through road segments that have large variances in traffic conditions (e.g. congestions, travel demands, traffic signals). On the other hand, segments from different bus lines could exhibit similar travel time patterns if they are subjected to similar traffic conditions. As such, a prediction model that is constructed based on bus line segments with similar traffic patterns (even though they are not spatially connected) could lead to more accurate prediction. There are many other advantages for identifying and employing traffic patterns for bus journey time prediction: 1) It only needs to train a few prediction models (one for each cluster of a specific travel time pattern) for all the bus lines, and thus is of lower cost to retrain the prediction models to adapt to the evolving traffic conditions. 2) Some bus lines that lack sufficient training data can benefit from other bus lines that share the same traffic patterns.

The main contributions of this work are summarized as follows:

1. We propose a novel bus travel time prediction method that fully considers the heterogeneity in travel time patterns of bus line segments within the same bus line, while simultaneously exploring the commonalities of traffic patterns across bus line segments in different bus lines. We first identify the common travel time patterns (in changing trends) shared by different bus line segments via a Non-negative Matrix Factorization (NMF) based method. In the NMF, we employ $l_{2,1}$-norm minimization on regularization to generate sparse solutions so that each bus line segment shares similar travel time trend with only a few of the identified patterns. The projected gradient descent algorithm is applied to solve the NMF problem [15]. Then, we classify all the segments in bus lines into clusters such that each cluster is associated with a specific travel time pattern.
2. We train a separate Long Short Term Memory (LSTM) based prediction model for each cluster that captures the travel time pattern associated with that cluster. The clusters are coalesced to extract journey records of various distance for training the LSTM network. Features that can characterize roadway characteristics (e.g. distance), traffic conditions, bus dwelling time at bus stops, delays at intersections and bus speeds are extracted from historical bus trajectories, bus line information, road network, and weather data. These features are fed into the LSTM network for training. Our work is the first to demonstrate that bus travel time prediction models do not need to be confined to the spatial connectivity of the bus lines and exploiting common traffic patterns across different bus line segments can lead to better prediction accuracy.
3. Unlike existing works, we focus on predicting the bus journey travel time for passengers that involve not only the riding times of multiple bus services but also the waiting times at transfer points. During prediction, the total journey time is calculated by aggregating the riding time components and the waiting time components. The riding time components are

estimated using the LSTM models and the waiting time components are calculated using historical average method relying on a large volume of historical bus arrival time records.

4. We conduct extensive experiments to evaluate our proposed approach using large scale real bus travel data involving 30 bus services, bus route information, road networks of Singapore, and weather condition data. The experimental results show that the proposed method significantly and consistently outperforms the baseline approaches.

Section 2 discusses related works and highlight the differences between our work and the existing ones. Section 3 introduces important definitions and the problem description, and Section 4 presents the proposed method for travel time prediction of bus journeys. The benefits of the proposed approach are evaluated in Section 5, and Section 6 concludes the paper.

2. Related Works

Bus Travel Time Prediction. In general, efforts to predict bus travel times can be categorized into the following categories: 1) *historical average* (HA) approach [14] predicts the travel time of a journey by relying on the historical average travel time for the same daily period over different days. It builds a non-parametric model that does not make any assumptions on the underlying data, and does not use any explicit training data. However, it is typically difficult to collect sufficient journey records for each origin-destination pair. 2) *Kalman Filters* (KF) approaches use a series of travel time records observed over time to produces estimates of unknown travel times, by estimating a joint probability distribution over the travel time records for each time frame [25]. Typically, a KF model cannot be generalized to the prediction of different time series [5]. 3) *Time Series Analysis* uses models to predict future values based on previously observed values by taking into account possible internal structure in the data [19]. However, this method is sensitive to complex scenarios with anomalies, which are common in bus journeys due to uncertainties caused by

bunching, delays at intersections, etc. 4) *ML* based models such as LR, SVM, and NN have been proposed for travel time prediction. LR models capture the linear relationship between travel times and the related factors [20]. This model is computationally efficient but usually, produce undesirable results for nonlinear systems. SVR methods have been used for predicting travel-time of cars on highways [29] and buses in city road network [33], due to its greater generalization capacity and can guarantee global minima for given training data. However, this model suffers from high computation overhead. Many NN approaches have been developed to predict bus travel time using both historical and real-time data [1, 4]. It is shown that NN based models have demonstrated an advantage over the KF model, HA model, ARIMA and classical regression models, because they have a better ability to model the traffic dynamics in road networks.

Journey Travel Time Prediction Many works have been reported to estimate the travel time of vehicles (taxis or private cars) between an origin-destination pair [29, 21, 27, 26]. However, these approaches are not suitable for bus journey time prediction because, the travel time of bus journey is affected not only by the traffic conditions (e.g. traffic flow, traffic signal) but also by other factors such as the dwell time at each bus stop, bus service frequency, etc. Also, the waiting time at interchange station during transfering need to be considered, which is a non-negligible part of a passengers' total journey time. The work in [6] predicted the travel time of bus journeys by partitioning a bus line into segments based on bus stops. The travel times over all segment are estimated separately and summed up to obtain the total journey time. However, it is shown that simply summing up the travel time of each route segments does not result in high prediction accuracy [10]. Moreover, the transfer time at interchange stations along the journey has not been taken into consideration. The work in [9] proposed to partition a bus journey into riding and waiting components based on transfer points. Since a separate prediction model is trained for each bus route, several hundreds of prediction models are needed for an entire city. It also neglects the heterogeneity in traffic patterns of bus line segments

within the same bus line and the commonalities of traffic patterns across route segments in different bus lines, which are important for accurate travel time prediction.

Traffic Situation-aware Prediction Situation-aware prediction methods have been used in traffic speed prediction of road segments [30, 3], where different roads have different speed patterns. These works proposed methods to predict the traffic speed of the same road segments using different prediction models while different road segments can share the same models if they are subjected to the same traffic condition. In our paper, we investigated the problem of bus journey time prediction, which is a more complicated scenario as a journey route cannot be simply treated as a single entity. This is because segments of the same bus line may behave significantly different in travel time due to heterogeneous traffic conditions while segments of different bus lines could also exhibit similar patterns. We propose a novel framework that explores the commonalities of travel time patterns among the segments and coalesces the journey route at unit segment level based on traffic pattern similarity of bus line segments, with the aim of improving prediction accuracy. For each traffic pattern (corresponding to a cluster), we rely on an attention-aware LSTM network to build a single prediction model, instead of using multiple prediction models as in [3].

Table 1 presents a detailed qualitative comparison of existing works with ours for travel time prediction of a journey route (the works in [17, 19, 20, 21, 29], which are restricted to travel time/speed prediction of only single road segments are not included in the table). We classified the existing works into three categories based on the vehicle type: 1) *general vehicle*: estimates the vehicle travel time for an input journey path or OD pair; 2) *bus*: predicts the travel (or arrival) time of a bus service which has fixed travel route; 3) *passenger*: estimates the individual passengers' total travel time using public transport (e.g. bus services) that may include both the riding time on buses and the waiting time for the bus services. In general, the existing works vary significantly in the type of transport network (road network, bus network, and grid network), estimation strategy (individual-based or collective-based), datasets (e.g. taxi trajectories,

Table 1: Detailed comparison of existing methods for predicting travel time of a journey path.

Problem category	general vehicle					bus travel time				individual passenger	
Existing works	[26]	[33]	[28]	[10]	[12]	[1]	[5]	[14]	[33]	[9]	our
road network	√		√	√							
bus network						√	√	√	√	√	√
grid network		√			√						
consider waiting time										√	√
individual-based				√							
collective-based	√	√	√		√	√	√	√	√	√	√
taxi trajectory	√			√	√						
bus trajectory							√	√	√	√	√
trip data		√	√			√					
Neighbor-based		√					√	√			
KF							√				
SVR									√		
LR				√		√					
(deep) neural network	√		√		√	√	√			√	√

bus trajectories, trip data, etc.), and prediction models (e.g. neighbor-based method, KF, SVR, LR, NN, etc.).

For the datasets, a trajectory consists of a sequence of points, where each point typically contains GPS location and the corresponding timestamp. Some works also include extra features for each trajectory point, such as the bus speed, number of boarding passengers, bus dwell time etc., to achieve more accurate prediction. On the other hand, a trip record typically consists of the trip information including origin, destination, journey start time, total travel time, etc., and no trajectory points are stored. With regards to the prediction model, the traditional methods (such as KF, LR, SVR, ARIMA, etc.) typically take into account the environmental factors such as weather condition (e.g., heavy storm), network characteristics (e.g., road type, distance, connectivity), time of day (e.g., rush hours) and holidays. However, these methods face difficulty in modelling the complex nonlinear spatiotemporal correlations. The deep learning models provide a promising way to capture spatiotemporal correlations for

traffic prediction. For example, the traffic conditions have inherent temporal patterns (e.g. daily and weekly periodicity), which can be learned using recurrent neural network (e.g. LSTM), while the convolutional neural networks (CNN) can capture the spatial dependency among neighboring journey route segments. This motivates the approach taken in our work where we also rely on the LSTM network to capture the temporal correlations for bus riding time prediction.

3. Definitions and Problem Description

Definition 1 (Bus Line): A bus line is a fixed route that is regularly traveled by the bus, and it can be represented by a sequence of points $BL_l =< p_1^l, p_2^l, \ldots, p_{n_l}^l >$ where $p_i^l = (x_i, y_i)$, for $i = 1 \ldots n_l$, is the GPS location of the i-th bus stop along the bus line BL_l, and n_l is the number of bus stops in BL_l. We use bus stops as points to represent a bus line/route as predicting the arrival time at a bus stop is usually desired. Bus passengers tend to be only interested in the arrival time at a bus stop rather than a random point along the route. In this paper, the notation bus line, bus route, and bus service are used interchangeably. A *bus line segment* is a set of connected points, e.g. $R_{i,j}^l =< p_i^l, p_{i+1}^l, \ldots, p_j^l > (i < j)$ indicating the segment from stop p_i^l to stop p_j^l of the bus line BL_l. Particularly, the bus line segment $R_{i,j}^L$ is called a ***unit segment*** if p_i^L and p_j^L are consecutive bus stops of the bus line BL_l.

Definition 2 (Bus Trajectory): A *bus trajectory* T is a sequence of consecutive GPS points that record a bus' travel information, i.e. $BT = \{p_1, p_2, ..., p_{|BT|}\}$. Each point p_i contains the latitude information, longitude information, and timestamp information. A bus trajectory records the arrival time of the bus at each of the bus stops along the bus line. Based on the bus trajectory, the actual bus travel time between any segment of the trajectory can be derived.

Definition 3 (Bus Journey): A *bus journey* signifies a complete travel from the passenger's origin to the destination, which may involve multiple

journey-segments using different bus lines/services. Passengers typically need to wait for bus service at the origin stop as well as the intermediate bus stops/interchange station. Without loss of generality, the bus stop/interchange station where a passenger waits for the first bus service is also regarded as a transfer point.

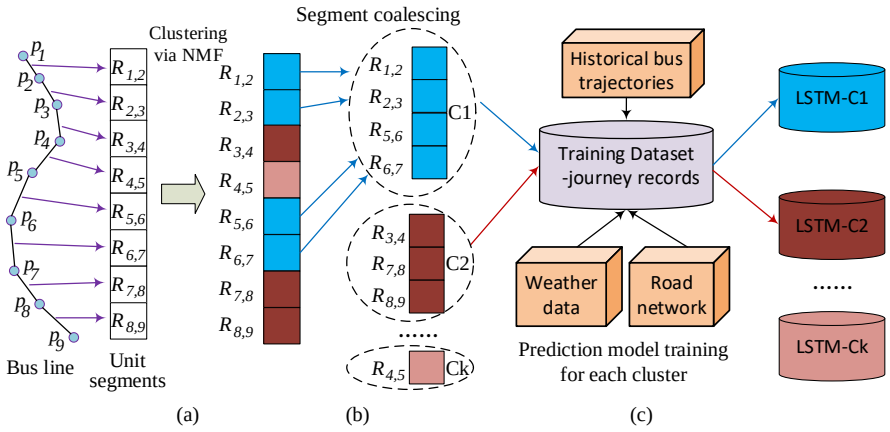


Figure 1: Proposed framework: Model training stage.

4. Proposed Method

4.1. Framework

The proposed framework, Traffic Pattern centric Segment Coalescing Framework (TP-SCF), for bus journey time prediction consists of three major stages.

1. The first stage aims to explore common travel time patterns across different bus line segments. This is based on the premise that segments from different bus lines may exhibit similar travel time patterns if they are subjected to similar traffic conditions. We first identify the hidden travel time patterns that are shared by bus line segments. We then cluster the bus line segments with similar travel time pattern, as shown in Fig. 1 (a) and (b).
2. Each of the obtained clusters is associated with a specific traffic pattern that is shared among segments from the same or different bus lines. We

train a separate LSTM-based prediction model for each cluster that captures the travel time pattern associated with that cluster, as shown in Fig. 1 (c). The models are trained using features that can characterize traffic conditions, bus dwell times at bus stops, delays at intersections as well as roadway characteristics.

3. The third stage performs bus journey time prediction, as shown in Fig. 2. We partition a given journey into riding time components and waiting time components. The riding times are predicted using the LSTM models obtained in the second stage, while the waiting times are calculated using the historical average (HA) method. Finally, the riding times and waiting times are aggregated to calculate the total journey time.

4.2. Traffic Pattern Clustering

The aim of this stage is to identify travel time patterns that are shared among bus line segments, and cluster bus line segments with similar patterns.

4.2.1. Data Representation of Bus Line Segment

For each unit segment $R^l_{i,i+1}$ of a bus line BL_l, we use a travel time vector $\mathbf{X}^{i,l} \in R^M$ to characterize the traffic situation (travel time pattern) associated with the segment. $\mathbf{X}^{i,l}$ contains M values corresponding to the historical travel times of M time intervals through a day, which can reflect the difference in traffic condition across different time intervals. Each value is the average of all historical travel times of the same interval over all days. In this paper, $M = 36$ as the time span of observation is from 6:00 am to 24:00 pm with a time interval of 30 minutes. The vector $\mathbf{X}^{i,l}$ represents a changing trend of travel time (*pattern*) associated with the bus line segment $R^l_{i,i+1}$ (*segment pattern* for short). We combine the vectors of each unit segment $R^l_{i,i+1}$ into a matrix $\mathbf{X}$, where $\mathbf{X} \in R^{N \times M}$, $N = \sum_{l=1}^{|BL|} n_l - 1$, $n_l - 1$ is the number of unit segments in bus line BL_l, BL is the set of involved bus lines.

4.2.2. Traffic Pattern Identification

We rely on Non-negative Matrix Factorization (NMF) technique to identify the hidden patterns (i.e. changing trends) of travel times associated with the bus line segments. The NMF method has been used to describe the typical temporal patterns of the global traffic states (or human mobility flows) and achieves long-term prediction of the large-scale traffic evolution [7, 23, 18]. In this paper, we aim to identify K hidden patterns so that all segments can be grouped into K clusters and each cluster is associated with a specific pattern, where K is a hyperparameter.

With the matrix $\mathbf{X} \in R^{N \times M}$ and the cluster number K ($K < M$), we use NMF to find two non-negative matrices $\mathbf{V} \in R_+^{N \times K}$ and $\mathbf{H} \in R_+^{K \times M}$ to approximate the original matrix $\mathbf{X}$. The NMF is formulated as

$$\arg\min_{\mathbf{V},\,\mathbf{H}} \parallel \mathbf{X} - \mathbf{V}\mathbf{H} \parallel_F^2 + \rho_1 \parallel \mathbf{V} \parallel_{2,1} \tag{1}$$

where the $l_{2,1}$ norm is given by $\parallel \mathbf{V} \parallel_{2,1} = \sum_{i=1}^{N} \sqrt{\sum_j^K V_{i,j}^2}$. Existing research has shown that minimizing $l_{2,1}$-norm usually generates sparse solutions [32, 16]. Each row of $\mathbf{H}$ represents a specific traffic pattern associated with the corresponding cluster (*cluster pattern* for short). The cluster pattern is the common knowledge regarding the travel time sequences of all segments in the cluster (each sequence contains M values corresponding to M time intervals), which is obtained by solving the NMF problem. The obtained matrix $\mathbf{V}$ has very few components with relatively large values in each row, meaning that the travel time pattern of a bus line segment (each row of $\mathbf{X}$) is similar to the patterns of only a few clusters. The projected gradient descent algorithm is applied to solve the NMF problem [15].

4.2.3. Cluster Label Assignment

After matrix factorization, each row $\mathbf{X}_i$ of matrix $\mathbf{X}$ can be represented as an additive combination of the rows in matrix $\mathbf{H}$ (or each segment pattern can be represented as a combination of the cluster patterns). The row $\mathbf{V}_i$ of matrix $\mathbf{V}$ contains the weights of the linear combination of each cluster pattern to form

$\mathbf{X}_i$. The *Cluster Label Assignment* problem considered in this section assigns a cluster label for each row $\mathbf{X}_i$ of matrix $\mathbf{X}$, based on the matrix $\mathbf{V}$.

The NMF approach has been used in [31, 3] to cluster document data and sensor data. In those works, after the matrix $\mathbf{V}$ is obtained, the cluster label for row $\mathbf{X}_i$ is assigned to cluster j maximizing $V_{i,j}$. In this work, for a data record $\mathbf{X}_i$, we do not simply assign cluster $c = \arg\max_j V_{i,j}$ to it. We impose a restriction that all segments of the same bus line are associated with at most k clusters (corresponding to at most k traffic patterns). By limiting the parameter k to a small value, fewer LSTM models are required to predict the total journey time.

We formulate the cluster label assignment as an optimization problem as follows. The binary matrix $\mathbf{Z} \in \{0,1\}^{n_l-1,K}$ is used to represent the assignment of cluster labels to the rows of $\mathbf{X}$ (affiliated with bus line BL_l), where $Z_{i,c} = 1$ if cluster label c is assigned to row $\mathbf{X}_i$, and 0 otherwise (n_l is the number of bus stops in BL_l and K is the number of clusters). The vector $\mathbf{Q} \in \{0,1\}^K$ is used to indicate which cluster labels are assigned to segments of bus line BL_l, where $Q_c = 1$ if cluster label c is assigned to at least one segment of BL_l, and 0 otherwise. The problem is formulated as

$$\max_{1 \le i < n_l, 1 \le j \le K, Z_{i,c} \le Q_c, \|\mathbf{Q}\|_1 \le k} Z_{i,j} V_{i,j} \tag{2}$$

where $\| \mathbf{Q} \|_1 \le k$ ensures that at most k clusters can be selected for the bus line while $Z_{i,c} \le Q_c$ ensures that a cluster label can be used only if it is selected for the bus line. The problem can be efficiently solved since the problem size is small ($K < 20$, $n_l < 100$ for each bus line).

4.3. Prediction Model for Each Cluster

In this section, we describe the training prediction model of each cluster, which captures the travel time patterns associated with various traffic conditions. We first discuss how to build the training dataset and introduce the features that are used to train the LSTM network. Next, we present our LSTM network for travel time prediction. The discussions in the following sub-sections

will be restricted to a single cluster (i.e. $C1$), and other clusters can be processed in the same way.

4.3.1. Training Dataset

We can easily reconstruct a pseudo bus line by coalescing (concatenating) multiple segments from the same bus line as shown in Fig. 1 (b). Since a practical journey usually contains 1, 2 or 3 bus lines, we further concatenate every three pseudo lines to form one relatively longer pseudo bus lines. We treat the segments of each pseudo bus line as connected even though they are physically not. The journey records of the pseudo bus line are then extracted as follows.

For each pseudo bus line BL, we extract training data as a matrix $\mathbf{x} \in R^{F \times D}$, where F is the number of journey records and D is the dimension (number) of features. Each row of matrix $\mathbf{x}$ is a feature vector associated with a journey made at a certain time interval. The matrix of journey records is obtained in the following way. For a given route (a segment of a pseudo bus line), we obtain a journey record x_t at each 30-minutes time interval for a period of 63 days, resulting in a sequence of 63×36 journey records, where 36 is the number of time intervals in each day (from 06:00 am to 12:00 pm). The pseudo bus line BL with n_{BL} unit segments has $\frac{n_{BL} \cdot (n_{BL}-1)}{2}$ different route segments in total, hence the total number of journey records collected for pseudo bus line BL is $63 \times 36 \cdot \frac{n_{BL} \cdot (n_{BL}-1)}{2}$. As a result, the matrix $\mathbf{x}$ has $F = \sum_{BL \in C1} 63 \times 36 \cdot \frac{n_{BL} \cdot (n_{BL}-1)}{2}$ rows, where $C1$ is the cluster obtained via NMF and BL is a pseudo bus line. The ground truth vector of the journey travel time is denoted as $\hat{\mathbf{y}} \in R^F$. In addition, we use $\mathbf{y} \in R^F$ to denote the target vector. One sub-matrix is extracted for each pseudo bus line, and the matrices of all pseudo bus lines are combined together to train a prediction model for the cluster $C1$.

Each row of the matrix $\mathbf{x}$ is a feature vector containing the following features that impact the journey travel time:

- *Time of day*, i.e. journey start time. This can be used as an indicator to characterize the variance of traffic conditions over a day.

- *Day of week*, i.e. day that journey will be made. This can be used to differentiate the traffic conditions between working days and weekends.

- *Travel distance*, i.e. total distance of the journey route.

- *Number of bus stops*, i.e. number of stops between the origin stop and the end stop along the journey route. This reflects the expected number of bus stopping and the bus dwelling time.

- *Number of intersections*, i.e. number of intersections along the journey route, including pedestrian crossings. Note that buses typically slow down at intersections.

- *Number of traffic signals*, i.e. number of traffic signals along the journey route. Buses often need to stop at the intersections with signals.

- *Weather condition*. This affects the bus moving speed and travel demands. There are 14 categories of weather conditions, including heavy thunderstorms, rain showers, light rain, sunny, etc. We arranged all the categories in the order of good conditions (e.g. sunny) to bad conditions (e.g. strong thunderstorms), as this will represent the conditions that will progressively impact the journey time. The ordered weather conditions are denoted as numbers from 1 to 14.

4.3.2. LSTM based Network Structure

The input matrix is fed into two stacked LSTM layers, where each LSTM layer has 128 neurons. The LSTM memory cell can be described with the

following equations:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\widetilde{\mathbf{C}_t} &= \tanh(\mathbf{W}_{Cx}\mathbf{x}_t + \mathbf{W}_{Ch}\mathbf{h}_{t-1} + \mathbf{b}_C) \\
\mathbf{C}_t &= \mathbf{i}_t * \widetilde{\mathbf{C}_t} + \mathbf{f}_t * \mathbf{C}_{t-1} \\
\mathbf{h}_t &= \mathbf{o}_t * \tanh(\mathbf{C}_t)
\end{aligned}
\tag{3}
$$

where t indicates the t-th timestamp, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ refer to the output of the input gate, forget gate and output gate respectively. $\mathbf{x}_t$, $\mathbf{c}_t$, $\mathbf{h}_t$ are the input vector, state vector and hidden vector respectively, and $\mathbf{h}_{t-1}$ is the former output of $\mathbf{h}_t$. $\widetilde{\mathbf{C}_t}$ and $\mathbf{C}_t$ are the input state and output state of the memory cell, and $\mathbf{C}_{t-1}$ is the former state of $\mathbf{C}_t$. σ is a sigmoid function. $\mathbf{W}_{ix}, \mathbf{W}_{fx}, \mathbf{W}_{ox}, \mathbf{W}_{Cx}$ are the weight matrices connecting $\mathbf{x}_t$ to the three gates and the cell input, $\mathbf{W}_{ih}, \mathbf{W}_{fh}, \mathbf{W}_{oh}, \mathbf{W}_{Ch}$ are the weight matrices connecting $\mathbf{x}_{t-1}$ to the three gates and the cell input, $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$, $\mathbf{b}_C$ are the bias terms of the three gates and the cell gates. All the above-mentioned parameters are initialized randomly and learned automatically through backpropagation during the learning stage.

The output of the LSTM layers goes into several fully-connected layers, where each layer is of size 128. The fully-connected layers are connected with residual connections, which is shown to be effective for training a very deep neural network [8]. The residual connection adds shortcuts between different layers, thus previous information flow can skip one or more non-linear layers through the shortcut and the skipped layers just need to learn the 'residual' of the non-linear mapping. For the first fully connected layer, its input is the output of the last LSTM layer. Let σ_{f_i} be the i-th residual fully-connected layer, then the output of the first layer is $\sigma_{f_i}(\mathbf{o}_z)$, where $\mathbf{o}_z$ is the output of the LSTM layer. For the rest of the residual layers, let $\mathbf{o}_{f_i}$ be the output of the i-th layer, then the output of the $(i+1)$-th layer can be represented as $\mathbf{o}_{f_{i+1}}$=$\mathbf{o}_{f_i} \oplus \sigma_{f_{i+1}}(\mathbf{o}_{f_i})$, where $\oplus$ is an element-wise add operation.

Finally, we apply a tanh activation function and obtain the prediction results.

In order to prevent overfitting, two widely used regularization techniques are employed: dropout and L_2 regularization. The dropout mechanism is applied to each hidden layer, where the rate of dropout is set to 0.5 [22]. Moreover, we apply L_2 regularization on model weights to prevent possible overfitting. Formally, the loss function used for training the model is:

$$L_{loss} = \sum_{i=1}^{F} (\hat{y_i} - y_i)^2 + \lambda \parallel \mathbf{W} \parallel^2 \tag{4}$$

where λ is a hyper-parameter to control the regularization strength and $\mathbf{W}$ denotes all weights in the network. The Adam optimizer is utilized as the gradient descent optimization algorithm. The training process repeats for 50 epochs.

4.4. Bus Journey Time Prediction

During the prediction stage, the entire journey time is partitioned into riding time components (moving along the bus line segments) and waiting time components (at origin stop or transfer points). Then the total journey time is calculated as the sum of all riding time components and waiting time components,

$$T = \sum_{p \in TP} Waiting(p) + \sum_{bl \in BL'} Riding(bl) \tag{5}$$

where TP is the set of transfer points (including the origin stop), $Waiting(p)$ is the waiting time at bus stop p, BL' is the set of coalesced journey routes, and $Riding(bl)$ is the required riding time for travel over bus line segment bl.

4.4.1. Bus Riding Time Prediction

The journey route is reconstructed by coalescing the corresponding line segment clusters as shown in Fig. 2, based on the cluster labels of bus line segments obtained via the NMF method as discussed in Section 4.2. The travel time of the bus line segments is predicted using the prediction model trained for the corresponding cluster. As shown in Fig. 2, the input journey route is partitioned into three route segments, bl_1, bl_2, bl_3, and two waiting time components

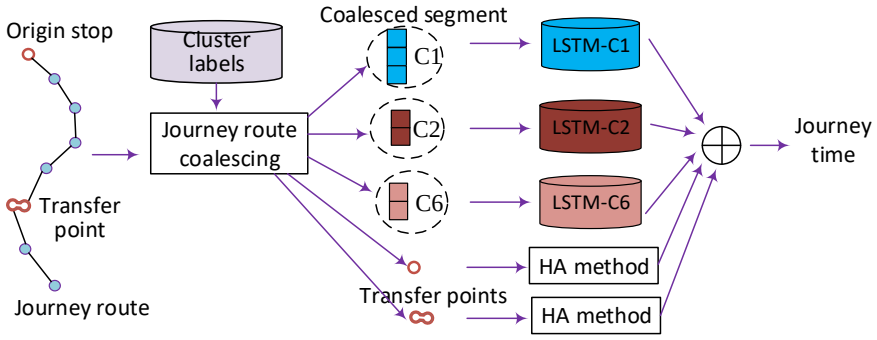


Figure 2: Proposed framework: Prediction stage.

(i.e. waiting at the origin stop and the transfer point). The riding times of three route segments, $Riding(bl_i)$ $(1 \leq i \leq 3)$, are estimated using the LSTM models of their corresponding clusters, i.e. C1, C2 and C6, respectively.

4.4.2. Waiting Time Prediction

We rely on a large number of historical bus trajectories (containing the information of the arrival times of buses at each bus stop) to estimate the waiting times at the transfer points (including the origin stop for waiting the first bus service). The Historical Average (HA) method is utilized, which is a data-driven approach. This avoids the assumption of a fixed distribution of waiting times at a bus stop. Let's assume that a passenger is expected to arrive at the bus stop p at time t_0 to wait for the bus (e.g. bus 179). Let BAT_p be the dataset of historical bus arrival times at bus stop p containing data of d days, then the historical average waiting time can be calculated as

$$Waiting(p) = HA(p, t_0) = \frac{\sum_{i=1}^{d}(t_i|_{BAT_p} - t_0)}{d},$$

where t_i is the first time bus 179 arrives after time t_0 in the i-th day of the dataset BAT_p. Thus $t_i|_{BAT_p} - t_0$ is the historical waiting time on the i-th day. For estimating the waiting time at an intermediate transfer point p', the passenger's arrival time t'_0 at this stop can be simply calculated as t_0 plus the estimated journey time between origin stop and the transfer point. Then the waiting time $Waiting(p')$ at p' is estimated as $HA(p', t'_0)$.

The HA method is further optimized as follow: 1) the historical dataset of bus arrival time is partitioned into two groups based on weekday and weekend, as bus frequencies on weekends are much lower than weekdays; 2) the existence of noise and missing values in the dataset of bus arrival time results in many incorrect records of historical waiting times. To mitigate this influence, only records that are smaller than the 90 percentile are utilized in the HA method; 3) the average time interval between two consecutive bus arrivals (during a period of 1 hour) is calculated, then all records that have waiting time larger than two times of the average time interval are removed.

5. Results and Analysis

5.1. Dataset

Road Networks. The road network of Singapore, obtained from OpenStreetMap[1], is utilized to derive the information of intersections as well the number of traffic signals for any journey routes.

Bus Route. The bus route information[2] includes the ID (a five-digit number) of each bus stop in sequential order, the GPS location (latitude and longitude) of each bus stop, and the travel distance between any two consecutive bus stops. We map the bus routes to the road network using the GPS locations of bus stops to determine the sequence of road segments traveled by the bus line. The results are verified by comparing with Google Map via visualization. The bus route data, together with road network information, is used to calculate the number of intersections and traffic signals covered by a journey route. 30 bus services are used in the experiment, which are shown in Fig. 3.

Bus Trajectories. A bus trajectory dataset is derived based on the real-world Bus Arrival Time dataset (the arrival time of the next bus for each bus stop, at every minute) provided by the Land Transport Authority, Singapore[2]. The dataset contains bus trajectory data of 30 bus lines from May 06 to July 07,

[1] https://www.openstreetmap.org/export

[2] https://www.mytransport.sg/content/mytransport/home/dataMall.html

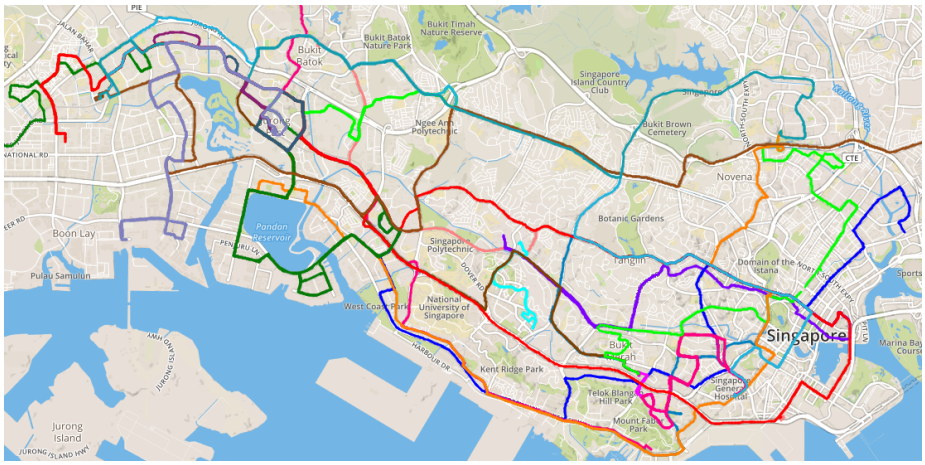


Figure 3: The spatial distribution of the bus routes utilized in the experiment.

2017 (63 days in total). Each bus trajectory is a sequence of points, and each point contains the information of the stop ID, the GPS location of the bus stop, the timestamp (arrival time of the bus at the stop), and the bus line ID. With the trajectories, the following features are extracted for each journey record: the day-of-week, the journey start time, the journey travel time.

Pseudo Journey Records. It is challenging to obtain sufficient journey records of individual passengers due to privacy issues. However, since the historical bus arrival time for each bus service at each bus stop is known, it is easy to generate the correct journey record if the journey travel route, journey start time, and bus arrival time at each bus stop (along the journey route) are known. The generated journey records can be used to evaluate the performance of our proposed method as well as train the baseline methods that rely on passengers' journey records. Please refer to [9] for more discussion on generating journey records.

We first select a journey route (which may involve multiple bus services) on the bus network and then generate a certain amount of journey records by randomly selecting journey start time over a period of 63 days. The journey records are generated based on real-world historical bus trajectories involving 30

bus lines in Singapore. Two types of journey records are generated: 1) journeys on single bus lines without transfers and 2) journeys on multiple bus lines with transfers. In the first category, we randomly select five journey routes (origin-destination pairs) that cover each bus line and generate 100 journey start times on each day (from Jun. 24 to Jul. 07, 2017). Hence, the testing set contains $5 \times 30 \times 100 \times 14 = 210,000$ journey records without transfers. In the second category, we first identify valid transfer stops between any bus line pair, and then randomly select the origin, destination and journey start time based on the transfer points using a method similar to the first category. There are $77 \times 100 \times 14 = 107,800$ journey records with single transfers in total, where 77 is the number of valid transfer points identified. We also identify 30 journey routes involving twice transfers; thus $30 \times 100 \times 14 = 42,000$ journey records with twice transfers are generated. The length of the generated journeys ranges from 1.1 km to 51.6 km, with an average of 15.1 km.

Weather data. Weather condition influences the bus travel speed by affecting the bus stopping time at bus stops as well as the moving speed of vehicles. Hourly-grained weather data are collected during the same time period, i.e. from May 06 to July 07, 2017[3]. There are 14 types of weather conditions, such as thundershowers, strong thunderstorms, rain showers, light rain, sunny, etc.

5.2. Baseline Methods for Bus Travel Time Prediction

1) *Historical Average* (HA) : It predicts the journey travel time as the average of all historical travel records of the same period that have the same origin and destination [14]. HA is commonly used as a baseline for travel time prediction [14]. Given the origin, destination, journey start time (interval), and historical journey records, the predicted travel time is the average of all historical travel records of the same period that have the same origin and destination. In this work, the HA uses all the historical records that fall into the same time interval of journey start time with the journey to be predicted. 2) *Auto Re-*

[3]https://www.timeanddate.com/weather/singapore/singapore

Table 2: Comparison of results on MAE (minutes), MAPE (%) and RMSE (minutes), EPKM (minutes/kilometer).

	metrics	HA	ARIMA	TFTS	LR	SVR	DNN	PCF	TP-SCF
	MAE	6.047	7.714	6.466	5.901	5.904	6.002	5.046	**4.863**
overall	MAPE	11.363	13.181	11.837	11.086	11.049	11.147	9.343	**8.603**
performance	RMSE	7.677	9.212	8.241	7.501	7.512	7.658	6.673	**6.389**
	EPKM	0.395	0.504	0.423	0.386	0.386	0.392	0.330	**0.318**
journeys	MAE	5.288	6.398	5.624	5.123	5.125	5.221	4.164	**3.865**
without	MAPE	11.962	13.829	12.474	11.610	11.558	11.656	9.310	**8.421**
transfer	RMSE	6.666	7.842	7.106	6.470	6.477	6.617	5.634	**5.154**
	EPKM	0.384	0.465	0.409	0.372	0.372	0.379	0.303	**0.281**
journeys	MAE	6.935	8.62	7.45	6.809	6.814	6.914	6.263	**5.913**
with	MAPE	10.663	12.424	11.092	10.474	10.455	10.552	9.389	**8.816**
transfer	RMSE	8.711	10.591	9.396	8.550	8.564	8.718	8.132	**7.582**
	EPKM	0.412	0.512	0.442	0.404	0.405	0.411	0.372	**0.351**

gression Integrated Moving Average (ARIMA): It is well-known for predicting time series data, which makes predictions solely based on historical data [17]. 3) *TensorFlow Time Series* (TFTS): We use the open source tool TFTS as one baseline [11]. 4) *Linear Regression* (LR): It is utilized to model the relationship between journey travel time and all the impact factors/features [20]. 5) *Deep Neural Network* (DNN): LSTM based neural networks have been used for travel time prediction and have achieved better performance in recent years [4]. 6) *Support Vector Regression* (SVR): Due to its high accuracy when trained with a sufficiently large dataset, SVR has been used for travel time prediction in some existing works such as [29]. 7) *Partitioning and Combination Framework* (PCF): It also partitions a journey into waiting time components (waiting times at the transfer stops) and riding time components (bus riding times on the used bus line segment) according to transfer points (not based on traffic patterns). Different from the method proposed in this paper, it trained one LSTM based prediction model for each bus line, which is used for predicting the riding time components [9].

5.3. Results Comparison

The performance measures used are the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and prediction error per kilometer (EPKM).

$$MAE = \frac{\sum_{i=1}^{F} |y_i - \hat{y}_i|}{F}$$

$$MAPE = \frac{100}{F} \sum_{i=1}^{F} |\frac{y_i - \hat{y}_i}{\hat{y}_i}|$$

$$RMSE = \sqrt{\frac{1}{F} \sum_{i=1}^{F} (y_i - \hat{y}_i)^2}$$

$$EPKM = \frac{1}{F} \sum_{i=1}^{F} |\frac{y_i - \hat{y}_i}{dist(i)}|$$

where F is the size of the testing set, $y_i \in R^F$ is the predicted value, $\hat{y}_i \in R^F$ is the actual value observed, and $dist(i)$ is the total distance of the $i-$th journey. In the experiments, the data collected from May 06, 2017 to June 23, 2017 are used for training the prediction model, while the data collected for the last two weeks (from June 24 to July 07, 2017) are used for testing. During training, 30% of the training set is used for validation.

5.3.1. Prediction Accuracy

Table 2 compares the performance of our proposed method TP-SCF and the 7 baseline methods. We observe that the two time series methods (i.e. ARIMA and TFTS) perform worse than other methods on all four metrics. This is because they need to update the prediction model with most recent observations and hence, they are not suitable for long-term prediction (i.e. for a journey with any start time within the testing period, e.g. next 14 days). Moreover, they are sensitive to anomalies and will become unreliable if there is a huge difference in travel time between two consecutive time steps. The classic regression methods (HA, LR, SVR) achieve relatively better results than time series methods because they train the prediction models using the entire training set,

while the time series methods rely on only the most recent observations. However, they still fail to characterize the complex nonlinear correlations among the historical data. Even though the DNN method relies on the LSTM network to capture the temporal patterns of the travel data (e.g. daily and weekly periodicity), it fails to properly model the waiting time at origin/transfer stops due to insufficient features. For example, among the features that impact the bus riding time, i.e. time of day, day of week, travel distance, number of bus stops/intersections/traffic signals, and weather conditions, only the first two features seem to impact the expected waiting time (for offline prediction where the bus location is not known). The proposed method produces better performance than all baselines on MAE, MAPE, RMSE as well as EPKM. For example, the average improvements compared with the baseline methods on MAPE are 17.3%, 29.0%, 20.5%, 15.8%, 15.7%, 16.5%, and 4.4%, respectively. This is because the proposed method can capture heterogeneous traffic situations along the journey routes. In particular, the comparison between the proposed method and the PCF method verifies that it is important to identify and employ traffic patterns for travel time prediction of bus journeys. In this way, it not only reduces the number of prediction models but also leads to better accuracy.

In addition, TP-SCF achieves better results on journeys without transfers than that on journeys with transfers. This is because TP-SCF partitions a given journey into multiple components (riding time and waiting time components), and the journeys with transfers typically have more components than journeys without transfers thus leading to larger accumulated errors. On the other hand, all the baseline methods obtained better MAE and RMSE on journeys without transfers, but better MAPE on journeys with transfers. Since journeys with transfers are typically longer than those without transfers, it is reasonable that larger MAE and RMSE are obtained on journeys with transfers for the baseline methods. The decrease in MAPE is due to the fact that the increase in prediction errors is not as large as the increase in journey length.

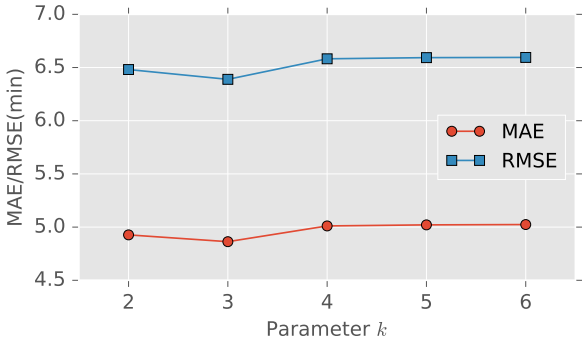
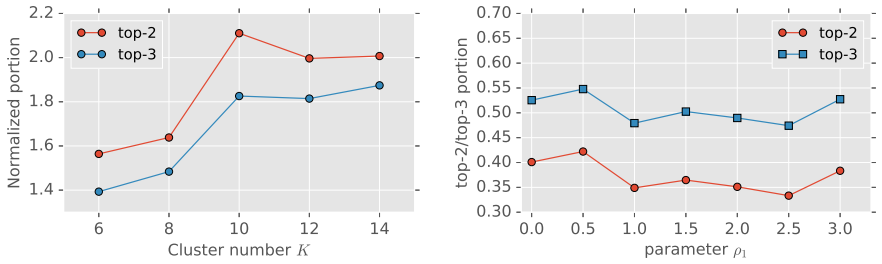


Figure 4: Effect of k on the performance of travel time prediction.

5.3.2. Effect of parameters k, K, and ρ_1

We now discuss the effect of hyperparameters and how to select the parameters. TP-SCF method restricts the segments of each bus line are associated with at most k clusters. Fig. 4 shows the effect of k on the prediction error, where K and ρ_1 are set to 10 and 0.5, and k ranges from 2 to 6 step by 1. It can be observed that the best MAE and RMSE value is achieved when $k = 3$. When k reduces, the heterogeneous traffic situations cannot be fully characterized, thus leading to lower prediction accuracy. However, large k will also result in low accuracy because a journey route will be partitioned into too many pieces leading to large accumulated errors for prediction.

As discussed before, two matrices $\mathbf{V}$ and $\mathbf{H}$ are obtained via the NMF method, and the bus line segment clustering relies on the matrix $\mathbf{V}$. The row $\mathbf{V}_i$ of matrix $\mathbf{V}$ contains the weights of the linear combination of each cluster pattern (i.e. each row of $\mathbf{H}$) to form X_i. Our goal is that, in each row, only two or three values are large while other values are relatively small. The rationale is that we do not simply assign the cluster c ($c = \arg\max_j V_{i,j}$) to the line segment of $\mathbf{X}_i$, thus $\mathbf{X}_i$ should have alternative choices, i.e. the second or the third largest weight. Fig. 5(a) shows the results of $\mathbf{V}$ with varying K, where 'top-2-portion' ('top-3-portion') indicates the portion of the largest two (three) weights over the total weights of the corresponding rows. Note the top-i-portion



(a) Effect of K (b) Effect of ρ_1

Figure 5: Effect of parameters K and ρ_1 on traffic pattern clustering.

is normalized by dividing i/K where K is the number of clusters, e.g. the top-2-portion for $K = 8$ is divided by '$2/8 = 0.25$'. We observe that the top-2-portion and the top-3-portion have similar changing trend and both of them achieve the highest value when $K = 10$. Thus we select $K = 10$ to characterize the traffic conditions of all bus lines and distinguish the underlying latent grouping. This indicates that we can characterize the entire traffic conditions of all bus lines and distinguish the underlying latent grouping with 10 traffic patterns.

Fig. 5(b) shows the 'top-2-portion' and 'top-3-portion' for $K = 10$ with varying ρ_1. The values in this figure are not normalized as they belong to the same cluster number. It can be observed that $\rho_1 = 0.5$ achieves the highest value for both the top-2-portion and top-3-portion. Also, since small ρ_1 allows the NMF to reduce the error between matrix $\mathbf{X}$ and the approximation $\mathbf{V}\times\mathbf{H}$, meaning that smaller ρ_1 is prefered. Thus ρ_1 is set to 0.5.

Figure 6 compares the effect of different methods of traffic pattern identification (i.e., NMF and K-means) for prediction performance in terms of (RMSE). The number of traffic patterns K is set to 8, 10, and 12, respectively. The parameter k is set to 3, meaning that all segments of a same bus line are associated with at most 3 clusters, since partitioning a single bus line into too many clusters leads to poor performance. We can observe that NMF performs better than K-means, which demonstrates the superiority of the NMF method for discovering latent traffic situations. This is because by using the NMF method, each

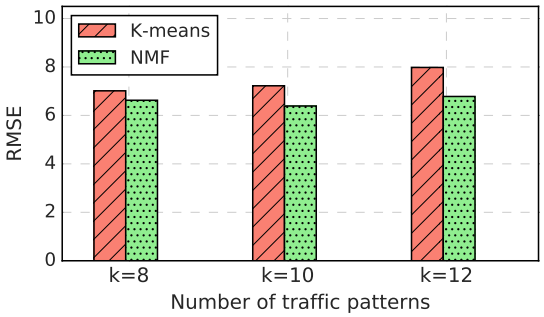


Figure 6: Effect of different methods of traffic pattern identification (i.e. NMF and K-means clustering) for prediction performance in terms of RMSE.

segment has multiple alternative clusters of similar traffic patterns. As such, some bus line segments can be assigned to alternative clusters (second/third best choice) if the restriction of parameter k is violated. On the other hand, using the K-means algorithm, some bus line segments do not have a satisfactory second choice, and assigning them to dissimilar clusters leads to performance loss in prediction.

5.3.3. Runtime of Prediction Model

We test the running time of journey time prediction on a PC with 3.50GHz CPU and 32GB RAM. Despite the longer training time of LSTM network, the bus journey prediction is very efficient. During the testing phase to estimate every 1000 journey queries, our TP-SCF method takes 0.567s, while the 6 baseline approaches, i.e. HA, ARIMA, TFTS, LR, SVR, DNN, and PCF, take 0.008s, 0.293s, 0.278s, 0.007s, 0.032s, 0.062s, 0.624s, respectively. Compared with the baselines, our method takes a little longer time as it needs to predict the travel time for each involved traffic condition and the waiting time at each transfer point, before merging them to obtain the final solution. It is worth noting that the running time can be significantly accelerated by deploying the predictors on a powerful server and running multiple predictors simultaneously. For example, when tested on the GPU (NVIDIA Quadro M4000) with 3.50GHz CPU and

32GB RAM, the TP-SCF method requires 0.153s to produce the results for the 1000 test instances, which provides an acceleration of 3.7 times.

6. Conclusions

In this paper, we investigated the problem of predicting bus journey time for passengers, that takes into account both the bus riding time and the waiting times at transfer points. We proposed an approach to automatically learn the heterogeneous traffic situations of different bus line segments, and train a separate prediction model for each disparate traffic pattern to improve prediction accuracy. We showed that, without using users' travel records, we can accurately predict the travel time of bus journeys by just relying on historical bus travel data. In addition, our work is the first to demonstrate that bus travel time prediction models do not need to be confined to the spatial connectivity of the bus lines, and exploiting common traffic patterns across different bus line segments can lead to better prediction accuracy. Our work also demonstrated that the sequence of bus line segments can be flexibly changed (segment coalescing) without strictly following the spatial connectivity of the bus lines. By conducting extensive experiments on large scale real-world bus travel data, we showed that our method can accurately predict the travel time for any given journeys and significantly outperforms the baseline approaches.

Predicting the waiting time at origin/transfer stops is challenging for offline scenarios due to the lack of effective features to characterize the dynamic situations. Currently, the proposed method for waiting time prediction has not fully considered the temporal dependency (e.g. daily and weekly periodicity). We plan to improve the algorithm for waiting time prediction in our future work by taking into account the frequency information of different bus services, as arrival times of a relatively low frequent bus service may follow a specific distribution (e.g. some time slots have much higher probability of bus arrival than others). In addition, the waiting time and bus riding time are currently predicted separately in our approach. In our future work, we plan to explore

prediction methods that jointly considers both the bus riding time and waiting time at the transfer stop. For example, one can generate a certain amount of individual passengers' journey records using the method presented in Section 5.1, and train a prediction model using trip records that contain both bus riding time and waiting time.

7. Acknowledgements

This research project is funded in part by the National Research Foundation Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme with the Technical University of Munich at TUMCREATE.

8. References

[1] Amita, J., Jain, S., Garg, P., 2016. Prediction of bus travel time using ann: a case study in delhi. Transp. Res. Procedia 17, 263–272.

[2] Beaudoin, J., Farzin, Y.H., Lawell, C.Y.C.L., 2015. Public transit investment and sustainable transportation: A review of studies of transit's impact on traffic congestion and air quality. Res. Transp. Econ. 52, 15–22.

[3] Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., 2017. Situation aware multi-task learning for traffic prediction, in: Proc. 2017 IEEE Int. Conf. Data Min. (ICDM), New Orleans, LA, USA, November 18-21, 2017, pp. 81–90.

[4] Duan, Y., Lv, Y., Wang, F., 2016. Travel time prediction with LSTM neural network, in: Proc.19th IEEE Int. Conf. Intell. Transp. Syst. (ITSC), 2016, Rio de Janeiro, Brazil, November 1-4, 2016, pp. 1053–1058.

[5] Fan, W., Gurmu, Z., 2015. Dynamic travel time prediction models for buses using only gps data. Int. J. Transp. Sci. Technol. 4, 353–366.

[6] Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A., Weidlich, M., 2017. Traveling time prediction in scheduled transportation with journey segments. Inf. Syst. 64, 266–280.

[7] Han, Y., Moutarde, F., 2016. Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization. Int. J. Intell. Transp. Syst. Res. 14, 36–49.

[8] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proc. 2016 IEEE Conf. Comput. Vis. Patt. Recog. (CVPR), Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778.

[9] He, P., Jiang, G., Lam, S., Tang, D., 2018. Travel-time prediction of bus journey with multiple bus trips. IEEE Trans. Intell. Transp. Syst. , 1–14.

[10] Jenelius, E., Koutsopoulos, H.N., 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. Transp. Res. PT B-Method. 53, 64–81.

[11] Jeon, T., 2019. Tensorflow tutorial for time series prediction. URL: `https://github.com/tgjeon/TensorFlow-Tutorials-for-Time-Series`.

[12] Lan, W., Xu, Y., Zhao, B., 2019. Travel time estimation without road networks: An urban morphological layout representation approach, pp. 1772–1778.

[13] Lee, E., 2012. Public Transportation in Singapore. URL: `http://worksingapore.com/articles/live_4.php`.

[14] Lee, W.C., Si, W., Chen, L.J., Chen, M.C., 2012. Http: a new framework for bus travel time prediction based on historical trajectories, in: Proc. 20th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst., ACM. pp. 279–288.

[15] Lin, C., 2007. Projected gradient methods for nonnegative matrix factorization. Neural Comput. 19, 2756–2779.

[16] Nie, F., Huang, H., Cai, X., Ding, C.H.Q., 2010. Efficient and robust feature selection via joint l2,1-norms minimization, in: Adv. Neural Inf. Process. Syst.,6-9 December 2010, Vancouver, British Columbia, Canada, pp. 1813–1821.

[17] Pan, B., Demiryurek, U., Shahabi, C., 2012. Utilizing real-world transportation data for accurate traffic prediction, in: Proc. 12th IEEE Int. Conf. Data Min. (ICDM), Brussels, Belgium, December 10-13, 2012, pp. 595–604.

[18] Qi, G., Huang, A., Guan, W., Fan, L., 2019. Analysis and prediction of regional mobility patterns of bus travellers using smart card data and points of interest data. IEEE Trans. Intell. Transp. Syst. 20, 1197–1214.

[19] Reza, R., Pulugurtha, S.S., Duddu, V.R., 2015. ARIMA model for forecasting short-term travel time due to incidents in spatio-temporal context. Technical Report.

[20] Rice, J., Van Zwet, E., 2004. A simple and effective method for predicting travel times on freeways. IEEE Trans. Intell. Transp. Syst. 5, 200–207.

[21] Salamanis, A., Kehagias, D.D., Filelis-Papadopoulos, C.K., Tzovaras, D., Gravvanis, G.A., 2016. Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction. IEEE Trans. Intell. Transp. Syst. 17, 1678–1687.

[22] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

[23] Sun, L., Axhausen, K.W., 2016. Understanding urban mobility patterns with a probabilistic tensor factorization framework. Transp. Res. PT B-Method. 91, 511–524.

[24] Sun, Y., Jiang, G., Lam, S.K., Chen, S., He, P., 2019. Bus travel speed prediction using attention network of heterogeneous correlation features, in: Proc. 2019 SIAM Int. Conf. Data Min., SIAM. pp. 73–81.

[25] Vanajakshi, L., Subramanian, S.C., Sivanandan, R., 2009. Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. IET Intell. Transp. Syst. 3, 1–9.

[26] Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y., 2018a. When will you arrive? estimating travel time based on deep neural networks, in: Proc. Thirty-Second AAAI Conf. Artif. Intell., (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 2500–2507.

[27] Wang, H., Tang, X., Kuo, Y.H., Kifer, D., Li, Z., 2019. A simple baseline for travel time estimation using large-scale trip data. ACM Trans. Intell. Syst. Technol. 10, Article No. 19.

[28] Wang, Z., Fu, K., Ye, J., 2018b. Learning to estimate the travel time, in: Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., ACM. pp. 858–866.

[29] Wu, C.H., Ho, J.M., Lee, D.T., 2004. Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. 5, 276–281.

[30] Xu, J., Deng, D., Demiryurek, U., Shahabi, C., van der Schaar, M., 2015. Mining the situation: Spatiotemporal traffic prediction with big data. J. Sel. Top. Signal Process. 9, 702–715.

[31] Xu, W., Liu, X., Gong, Y., 2003. Document clustering based on non-negative matrix factorization, in: Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., ACM. pp. 267–273.

[32] Yang, X., Deng, C., Liu, X., Nie, F., 2018. New $l_{2,1}$-norm relaxation of multi-way graph cut for clustering, in: Proc. Thirty-Second AAAI Conf. Artif. Intell. (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp. 4374–4381.

[33] Yu, B., Yang, Z., Yao, B., 2006. Bus arrival time prediction using support vector machines. J. Intell. Transp. Syst. 10, 151–158.

[34] Zhang, H., Wu, H., Sun, W., Zheng, B., 2018. Deeptravel: a neural network based travel time estimation model with auxiliary supervision. Proc. Twenty-Seventh Int. Joint Conf. Artif. Intell. (IJCAI-18), Stockholm, Sweden, 13-19 July 2018 , 3655–3661.