HTTP: A New Framework for Bus Travel Time Prediction Based on Historical Trajectories

Wang-Chien Lee[1†] Weiping Si[1†] Ling-Jyh Chen[2‡] Meng Chang Chen[2‡]
[†]Department of Computer Science & Engineering, The Pennsylvania State University, PA, USA.
[‡]Institute of Information Science, Academia Sinica, Taiwan.
[1]{wlee,wus129}@cse.psu.edu [2]{cclljj,mcc}@iis.sinica.edu.tw

In this paper, we develop a new bus travel time prediction framework, called ***H**istorical **T**rajectory based Travel/Arrival **T**ime **P**rediction (HTTP)* for real-time prediction of travel time over future segments (and thus the arrival time at stops) of an on-going bus journey. The basic idea behind HTTP is to use a collection of historical trajectories "similar" to the current bus trajectory to predict the future segments. Specifically, the HTTP framework (1) samples a set of similar trajectories as the basis for travel time estimation instead of relying on only one historical trajectory best matching the on-going bus journey; and (2) explores different prediction schemes, namely, passed segments, temporal features, and hybrid methods, to identify the sample set of similar trajectories. We conduct a comprehensive empirical experimentation using real bus trajectory data collected from Taipei City, Taiwan to validate our ideas and to evaluate the proposed schemes. Experimental result shows that the proposed prediction schemes significantly outperforms the state-of-the-art and baseline techniques.

1. INTRODUCTION

With the ever-growing number of vehicles running on roads, traffic congestion has become one of the most serious problems nowadays, driving more and more people to take the public transportation systems (e.g., buses, subways and trains). It can be expected that, especially in metropolitan areas, the public transportation will become increasingly important to many people. Subsequently, customer satisfactory is a high priority to various public transportation services. In today's busy society, information regarding *arrival time* or *travel time* of transport from a place to another is valuable.[1] Thus, there is a high demand for accurate estimate of arrival and travel times in various transportation services.

Accurate estimation of travel times of public transportation is a challenging research problem that remains open for the past thirty years in the transportation research community [1, 18].[2] To predict the travel time on a given path, a simple approach is to adopt the *average travel time* derived from historical data. This approach, making constant prediction of the travel time for a path, apparently does not capture the dynamic traffic situation very well. Thus, advanced prediction techniques for travel time estimation have been proposed [1, 18, 21, 19, 8, 17, 16]. Generally speaking, these techniques share a common idea, i.e., discover certain regular patterns from the historical data collected over time, even though the specific approaches adopted are different. For example, some propose to fit the historical data to statistical models such as Gaussian models, Bayesian network and Markov Chains in order to facilitate statistical analysis [1, 18, 21]. On the other hand, techniques based on regression models learn from historical data regression functions of estimated travel time in terms of various external factors [19, 8]. Thus, a prediction is made by using the values of those factors under current situation as input to the function. Moreover, techniques based on time-series models focus on discovering the internal relationship among historical time-series data in order to identify similar patterns from historical data to make prediction under the current situation [17, 16]. Notice that there is no clear winner among these techniques as their performances are highly constrained by the quality/quantity as well as the types of data available. For example, conventional collection of traffic data is typically conducted by surveys [1, 18, 21, 19] or using expensive sensors deployed along the roads at some locations to record arrival times, traffic flow volumes, and other statistics of vehicles [8, 17, 16].

In recent years, due to the rapid advent of positioning and wireless communication technologies, wireless devices equipped with Global Positioning System (GPS) have been widely deployed on various private and public vehicles, generating massive amount of vehicle data, including instant speeds, locations, and so on, for fleet management and other transportation applications. The vehicle data, usually represented in form of *trajectories*, also brings a great potential for real-time estimate of the vehicle travel times. Thus, in this paper, we propose a new prediction framework to estimate the travel time of buses by exploiting collected bus trajectory data. Among the public transportation systems, the travel times of buses, which drives along with other vehicles on roads, are more difficult to predict than trains and subways, which ride on fixed rails. First, the travel condition of a bus may easily get affected by various internal and exter-

[1]Accurate prediction of arrival/travel time allows users to make timely plans of their upcoming activities and business.

[2]As the arrival time can be derived from travel time mostly, we focus on the travel time for the rest of the paper.

SIGSPATIAL GIS'12 November 6-9, 2012. Redondo Beach, CA, USA

nal factors, including accidents, weather, road construction, government policies and even temperature. Second, for vehicles in metropolitan areas, errors often exist in (positional) data acquisition and transmissions due to the interference and blocking of surroundings and other resources of errors. Fortunately, we do not need to capture all real-time data to make accurate travel time prediction. Historical trajectory data of buses can help!

Recently, research works on discovering traffic patterns from historical data collected from vehicles have received significant attentions [28, 33, 5, 13, 23, 33]. Particularly, these works show that traffic patterns exist in road segments and thus could be used to predict the future traffic condition on the same segment. This finding provides a concrete basis for using "similar trajectories" to predict the travel time of an on-going bus journey. In this paper, we develop a new bus travel time prediction framework, called ***H**istorical **T**rajectory based Travel/Arrival **T**ime **P**rediction (HTTP)* for real-time prediction of travel time at coming segments (and thus the arrival time at stops) of an on-going bus journey. The basic idea behind HTTP is to use a collection of historical trajectories "similar" to the current bus journey to predict the travel times in future segments of the bus journey. Specifically, the HTTP framework (1) samples a SET of similar trajectories as the basis for travel time estimation instead of relying on only ONE historical trajectory best matching the on-going bus journey; and (2) explores different features (e.g., travel times of passed segments as well as time/day of the bus trajectories) to identify the sample set of similar trajectories.

Several issues faced in design of the HTTP framework. For example, many features are associated with trajectories. Some of these features are categorical while the others are numerical. We need to use discriminative features and properly define similarity functions for those features in order to identify a sample set of similar trajectories effective for travel time prediction. To determine a set of similar trajectories based on travel time on passed trajectory segments, we consider the K-Means [15] and V-Clustering [32] algorithms to partition the whole spectrum of travel times into ranges of travel times. To determine a set of similar trajectories based on hours/days, we also use the K-Modes [10] algorithm to partition the hours/days feature space. Accordingly, the HTTP framework is able to retrieve the sample set of similar trajectories efficiently to estimate the travel times. To validate the proposed ideas and evaluate various prediction schemes proposed for HTTP, we conduct a comprehensive empirical experimentation using real bus trajectory data collected from Taipei City, Taiwan.[3]

This research work has made a number of significant contributions as summarized below.

- We propose a new system framework, namely, HTTP, for predicting the travel times over future segments of an on-going bus journey based on historical trajectory data. The HTTP framework consists of two major components: (i) similar trajectory retrieval; and (ii) travel time estimation.
- We perform a detailed data analysis to investigate the correlation between bus travel times in route segments and a number of trajectory features, e.g., passed segment travel time, hours, days, etc. Based on our analysis, we select a number of trajectory features to identify similar trajectories.
- We adopt clustering algorithms for different types of trajectory features in order to group similar trajectories together. These similar trajectory clusters allow us to efficiently and effectively retrieve a sample set of trajectories similar to the on-going bus journey.
- We study a number of travel time estimation schemes to derive the travel time prediction for future segments of an on-going bus journey.
- Through a comprehensive experimental study, using a real dataset collected from buses in Taipei City, Taiwan, we validate our proposed ideas and evaluate the HTTP framework in terms of prediction accuracy. The experimental results show that all the prediction schemes proposed under HTTP significantly outperforms the baseline and state-of-the-art schemes. Among our proposals, the *hybrid temporal features/passed segments* scheme achieves the best performance.

The remainder of this paper is organized as follows. In Section 2, introduce terminology, formulate the research problem, and review some related works. In Section 3, we analyze the collected historical trajectory data. Next, in Section 4, we give an overview of the HTTP framework, detailing its system design. In Section 5, we further discuss the proposed prediction schemes in details. In Section 6, we conduct a comprehensive experimental study using the collected real dataset of bus trajectories. Finally, we conclude this work in Section 7.

2. PRELIMINARIES

In this section, we first define some terms used in this paper, formulate the targeted research problem, and review relevant work in the literature.

2.1 Terminology

To facilitate our discussions in the paper, here we first introduce some basic terms, including *bus route*, *route segment* and *bus trajectory*. Notice that, in public transportation, a bus regularly runs on a fixed route, which can be represented by a polyline (i.e., a sequence of two-dimensional *route points* $p_1, p_2, ..., p_n$ where $p_i = (x_i, y_i)$ for $i = 1..n$; other points on the route can be obtained by interpolation). Those route points are connected by segments, denoted as $S_1, S_2, ...S_{n-1}$ where $S_i = \overline{p_i p_{i+1}}$ for $i = 1..(n-1)$. Accordingly, the length of a segment S is denoted as $|S|$. Naturally, a high-precision route presentation can be obtained by using a number of route points which keep route segments near linear. Logically, a number of points-of-interests can also be used to represent a route, e.g., bus stops. We use bus stops as the route points to represent a bus route as it is of high interest to predict the arrival time at a bus stop. Bus passengers tend to be interested in learning the arrival time at a bus stop rather than a random point along the route.

Definition 1. **Bus Route.** *A bus route R is represented as a sequence of points, $R = \langle p_0, ..., p_n \rangle$, where p_i stands for a bus stop. Additionally, $|p_i|$ denotes the total distance from the start of the route to the ith bus station. Thus, $|p_i| < |p_{i+1}|$.*

[3] http://www.ebus.taipei.gov.tw/

In this work, based on bus stops, we divide a route into route segments (or simply *segments* for short), which are partial routes between bus stops.

Definition 2. **Route Segment.** *A route segment S is a part of a route between two adjacent bus stops. Accordingly, a route R can be represented by a sequence of segments, $\langle S_0, ..., S_{n-1} \rangle$, where the length of the segment S_i is denoted as $|S_i| = |p_{i+1}| - |p_i|$.*

Due to the availability of positioning technology, buses equipped with GPS are able to update their positions (along with other bus status information) regularly and thus report the journey on a bus route as a trajectory, which consists of a time-stamped series of location points on the bus route. Notice that the location points in a trajectory are obtained in accordance with the GPS-dependent sampling scheme, i.e., these sample points may not be aligned with bus stops. To address this issue, for a given trajectory, the arrival time of a bus at a bus stop is obtained by interpolation. Therefore, the travel time for each route segment can be easily computed.

Definition 3. **Bus Trajectory.** *A bus trajectory T is represented as a sequence of $\langle (p_1, t_1), (p_2, t_2), ...(p_n, t_n) \rangle$, where $p_i \in R$ and t_i denotes the arrival time of a bus at the stop p_i. With the arrival time at each bus stop, it is straightforward to derive the travel time between two bus stops. Therefore, for a given route, a bus trajectory T can also be represented as a sequence $\langle \Delta t_1, \Delta t_2, ...\Delta t_N \rangle$, where Δt_i denotes the travel time on S_i and N is the total number of segments on this route (i.e., $N = n - 1$).*

A trajectory denotes a journey where a bus travels through the whole bus route to generate a travel time for each segment. Therefore, M historical trajectories provide M estimates of travel times for each segment, which may serve as a good basis for travel time prediction.

Definition 4. **Segment Travel Time.** *Given M trajectories of a particular bus route, for each segment, there are M travel times $\{\Delta t_1, \Delta t_2 ..., \Delta t_M\}$, where Δt_j is the travel time of a bus on this segment corresponding to the jth trajectory and M is the total number of historical trajectories on this bus route. The collection of segment travel times corresponding to S_i is denoted as ST_i.*

2.2 Problem Formulation

Definition 5. **Route Segment Travel Time Prediction.** *Consider a bus route $R = \langle S_1, ..., S_N \rangle$ with N segments. For a bus traveling on segment S_i of its bus route, its current (and incomplete) trajectory/journey $\dot{T}$ can be represented as a sequence of travel times of the passed segments, i.e., $\dot{T} = \langle \Delta \dot{t}_1, \dot{t}_2, ...\Delta \dot{t}_i \rangle$ where $1 \leq i \leq N$. Without loss of generality, the travel time prediction problem is to predict the travel times this on-going bus to spend in the remaining segments, i.e., $\Delta \dot{t}_{i+1} ... \Delta \dot{t}_N$, on the bus route.*

Problem Statement. Given a bus route, a repository of M historical trajectories on this route, and an on-going bus traveling on the route, we aim to develop an effective travel time prediction framework and prediction schemes by exploring the patterns hidden in the massive collection of historical trajectories on the route.

2.3 Related Works

There are many existing research works related to our study. In this section, we review some of them in i) arrival/travel time prediction; ii) trajectory similarity; and iii) trajectory patterns.

Arrival/Travel Time Prediction. Over decades, researchers have applied various models and methods on arrival/travel times prediction. In [34], mathematical models are developed by taking into account the travel times on links, dwell times at stops, and delays at intersections. In [14], algorithms are developed to provide bus arrival information based on bus location data, schedule information, the difference between scheduled and actual arrival times, and the waiting time at time-check stops. To a great extent, with the advances of artificial intelligence (AI), researchers have widely adopted AI methods (e.g., Kalman filtering [29, 20, 35] and artificial neural networks (ANNs) [24, 36, 3]) in real-time arrival/travel times predictions. Kalman filtering takes into account the stochastic properties of the process disturbance and the measurement noise. It works well for short-term prediction, but not for long-term prediction. ANNs has a great advantage in processing complex nonlinear relationships. However, these methods are limited by the extremely long training time. In recent years, other machine learning methods (e.g., Support Vector Regression (SVR) and Support Vector Machine (SVM)) have also become popular [27, 31]. Similar to ANNs, SVR is too expensive in training to deal with real-time updates. Recently, predicting methods based on historical trajectory data have also been developed in [11, 23, 22]. The authors show that the similarity between historical trajectories and current position data of a bus can be exploited to predict the bus arrival times at bus stations, which share the same intuition with our research work in this paper. In TransDB [22], the system searches the historical trajectory database for the "most similar" trajectory to the passed segments of the current bus trajectory in order to make a good prediction. The basic idea is that, based on the proposed trajectory similarity function, the nearest neighborhood trajectory (NNT) and the trajectory of current bus ride are anticipated to exhibit similar traveling behavior (in terms of travel time). Based on this assumption, the NNT serves as a good basis for predicting the future travel time of the current bus journey without explicitly taking into account various external and internal factors. The HTTP system proposed in this paper also aims to exploit the patterns in similar historical trajectories for making predictions. However, we argue that the historical trajectory "most similar" to the passed segments of the current bus trajectory alone may not provide the best prediction of the on-going bus ride. Thus, we collect a set of similar trajectories and adopt a statistical approach to make predictions. Additionally, we exploit different features associated with trajectories and develop different similarity functions to find similar trajectories that, as our experimental results show in Section 6, make significantly more accurate travel time predictions than TransDB.

Trajectory Similarity. Buses running back and forth on a fixed bus route naturally result in similar trajectories that capture hidden traffic factors in the historical data. Therefore, the similarity among trajectories provides a strong basis for predicting future travel times and thus plays an essential role in or system. Due to the availability of trajectory data, a great amount of research work has been conducted

to identify similar trajectories and time series (note that a trajectory can be seen as a time series). In [30], *Lp*-norm, by computing Manhattan Distance or Euclidean Distance, has been first proposed to measure trajectory similarity and widely applied in various applications. Over the years, various similarity measures have been developed and adopted. In [4], Dynamic Time Warping (DTW) is proposed and further enhanced in [2, 25]. Additionally, the concept of edit distance, introduced in [12], also receives significant attention, e.g., a widely used similarity measure based on edit distance is the Longest Common SubSequence (LCSS) distance, which has been widely used as the distance measure to fetch similar trajectories or time series [26, 7, 9]. While LCSS and DTW are applicable to our data, they are highly sensitive to noises and errors. Additionally, LCSS and DTW both emphasize on the overall similarity of the whole trajectory, without taking into account the similarity in individual segments, so they do not truly capture the similarity required for segment travel time prediction. In this paper, we propose our own similarity measures in HTTP.

Trajectory Patterns. Patterns of historical trajectories can be classified in two aspects: *trend* and *periodicity*. The trend refers a general, systematic component that changes over time and does not repeat (or at least does not repeat within the time range captured by data). The periodicity represents a component repeating itself in certain intervals over time. In [28], the daily periodicity from historical data of travel times around the same location is shown. In [33], an analysis is conducted to verify the existence of periodicity of speeds over time on a route segment. In [5, 13], patterns in trajectories are discovered by measuring the correlation between the traffic on a specific route and different time periods. Such patterns validate the ideas of using historical data associated with a certain segment to predict the future traffic condition on that segment.

3. DATA ANALYSIS

To better understand the characteristics of bus trajectory data, we perform an analysis on real data we collected from buses in Taipei, Taiwan, for more than one year (March 2010 to March 2011). Our goals for this data analysis are twofold: i) we would like to verify that there are evidences for supporting the ideas of using historical trajectory data to predict a current bus ride; and ii) we would like to explore patterns inside the data that can be used for the prediction.

The data consist of instant speeds, GPS coordinates and times of buses. Each bus and each route have their own IDs, respectively. We also obtain information about bus stops in the whole Taipei City, including the names, coordinates, and IDs of passing routes. Typically, there are multiple buses traveling on a route at different segments at the same time. There are totally 3,893 buses running on 394 routes daily. We perform an analysis on Route 10283 which consists of 64 stops (i.e., 63 segments), starting at stop Xinzhuang and ending at stop Dr.Sun Yat-Sen Memorial Hall. Its distance is 47.4 kilometers. From March 2010 to March 2011, there are totally 24,985 trajectories.

As our goal is to make travel time prediction on a segment, we are interested in observing i) whether there are correlations between segments (so we can use passed segments of an on-going bus journey to predict the travel times of the remaining segments) and ii) there are some patterns existing in the travel time of segments to be predicted. We first use

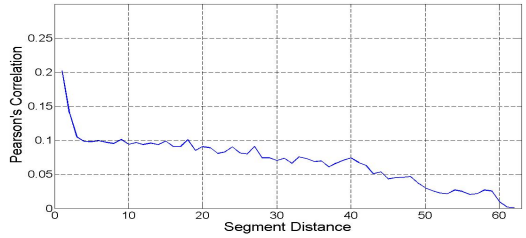
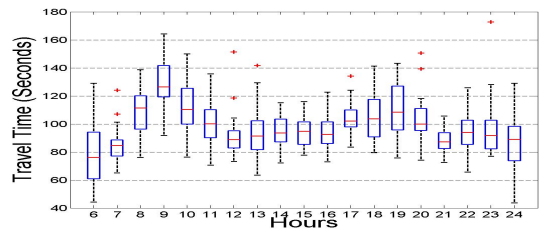
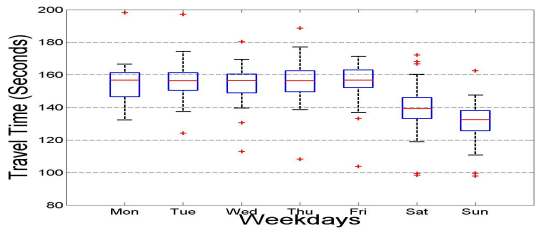


Figure 1: Travel time correlation between two segments

Pearson's correlation as the tool to measure the correlation between segments. As we anticipate that the correlation between any two segments to vary, we conduct the analysis by increasing the "segment distance" (e.g., the segment distance between Segment 20 and Segment 25 is 5) to observe how distance may affect the correlation.



(a) Analysis by hours



(b) Analysis by days

Figure 2: Travel Time Analysis by Hours and Days

As shown in Figure 1, correlations exist commonly between two arbitrary segments in a route but not very high for most pairs of segments. Not surprisingly, the closer two segments are, the higher their travel time correlation is. In other words, a segment is more related to near segments than farther ones. Additionally, there is an apparent rapid decline as the segment distance increases from 1 (i.e., adjacent segments). We conclude that using historical trajectories similar to the current trajectory in terms of passed segments "closeby" for prediction is more reliable than using all passed segments or those faraway.

Intuitively, travel times of a segment are not only related to that of near segments but also to some features. For example, in a big city, the traffic conditions during rush hours are usually worse than other time. Therefore, we may analyze the travel times to a hourly basis. Similarly, the travel times in the same segment may appear differently during weekdays and weekends. Figure 2 shows the trends of travel time at Segment 45 by hours of a day and days of

a week using one year of data.[4]

Figure 2(a) shows that, as expected, travel times in the rush hours (7am-9am and 4pm-7pm) are higher than other hours. It can be conclude that buses running in rush hours exhibit similar behavior, and those running in other (non-peak) hours are likely to be similar. Travel times also exhibit different patterns in different days of a week. As Figure 2(b) indicates, travel times in weekdays do not differ from each other too much. However, we do observe that the travel times drop dramatically during weekends. The difference between weekday and weekend are apparent. Such a trend supports the hypothesis that the travel times change in a similar way in a weekday (i.e., Monday to Friday). The travel times on Friday are slightly higher probably due to some additional traffic (e.g., vacations or out-of-town students/workers going home). Over the weekend, travel times are obviously lower than that in weekdays.

4. THE HTTP SYSTEM - AN OVERVIEW

According to the data analysis presented earlier, we exploit the travel time patterns exhibited in similar trajectories to develop a novel travel time prediction framework, called *Historical Trajectory based arrival/travel Time Prediction (HTTP)*, based on a large collection of historical bus trajectories. In this section, we first provide an overview of the proposed HTTP system framework and then, in Section 5, discuss a number of similar trajectory based prediction schemes proposed under this framework. Figure 4 shows our system design of the HTTP framework.

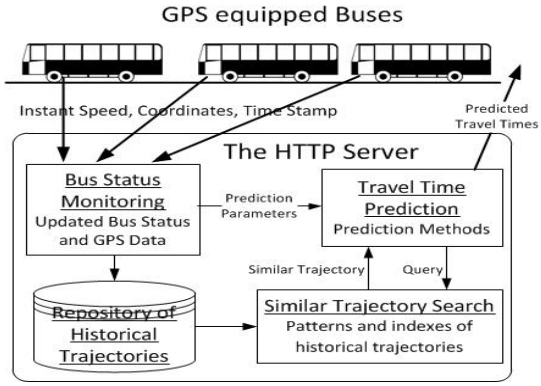


Figure 3: System architecture of HTTP

As illustrated, the proposed HTTP system (i.e., an location based service server) continuously collects bus trajectory data from GPS-equipped buses which report the latest bus status including time-stamped geographical coordinates of the bus and instant speed. The HTTP server is responsible for receiving and storing the trajectory data, monitoring the not-yet-completed trajectory journeys of on-going buses, and making prediction of bus travel time on the routes in responding to (i) passenger enquiries and (ii) real time update of bus arrival time at bus stops.[5] As shown in Figure 4, the HTTP server consists of three modules: a) Bus Status Monitoring (BSM) module,; b) Travel Time Prediction (TTP) module; and c) Similar Trajectory Search (STS) module.

The BSM module is responsible for communicating with the buses to receive bus status information and GPS data updates of the on-going trajectories. Once an update from a bus b reaches the server, BSM catches the bus status (such as instant speed, current bus coordinate and new time stamp) of b, extracts features associating with the developing trajectory T_b, and stores the information as part of T_b in the historical trajectory repository.

The TTP module predicts the arrival times of buses at bus stops, which can be reduced to a problem of predicting the travel times of buses on their remaining route segments. As mentioned, the TTP module can be invoked to make predictions by (i) a passenger enquiry; or (ii) the real time updates of bus arrival information at stops. The former arrives on demand and the latter usually happens periodically. Without loss of generality, we consider the latter scenario as (i) can be considered as a simplified case of (ii). For simplicity, we focus on predicting the travel time of a bus, given its current location, on remaining segments of its journey on the bus route. Moreover, instead of constantly making predictions, we assume that TTP is invoked every time when BSM receives the updated bus status (including the GPS data of bus location) and passes the required input parameters for prediction to TTP. Our idea behind the TTP module is very simple - find a sample set of historical trajectories similar to the on-going bus journey as a statistical base to estimate the travel time for prediction.

Obviously, TTP relies on the STS module to search for similar trajectories effectively and efficiently. As there could be different ways to identify the sample set of similar trajectories, different notions of similarity could be explored to ensure the effectiveness of TTP. On the other hand, with a massive amount of historical data, it is infeasible to make exhaustive comparison between the trajectory of current bus journey against all the corresponding historical trajectories. To ensure the search efficiency, we create indexes of trajectories and related patterns in the STS module to avoid retrieval of redundant trajectories not useful for our travel time estimation, i.e., only a relatively small set of candidate trajectories are returned to TTP.

5. SIMILAR TRAJECTORY BASED PREDICTION SCHEMES

As discussed earlier, we design HTTP as a general framework to support travel time prediction. Based on this framework, the remaining issue is to devise schemes which first invoke the STS module to retrieve a sample set of similar trajectories for making effective travel time estimation in the TTP module. Based on our data analysis, we observe the *travel time correlation between two segments* and the *travel time patterns corresponding to some temporal features* such as hours and days. Therefore, we follow these observations to introduce two different schemes based on *passed segments* (PS) and *temporal features* (TF). As their names suggest, these two schemes use the passed segments and temporal features of an on-going bus journey, respectively, to identify similar trajectories for prediction.

[4]Analysis on other segments exhibits similar patterns. We have also analyzed the travel time in terms of instant speed but observe no obvious pattern. We do not show those analysis results due to space constraint.

[5]Passengers are able to issue enquiries via mobile web services while many bus stops may be equipped with displays of bus information.

5.1 Passed Segments Scheme

To identify similar trajectories by travel times on passed segments, an idea is to employ conventional similarity measures for time series, e.g., LP-norm, DTW and $LCSS$. However, through our preliminary experimentation, we found that these algorithms are highly sensitive to errors or outliers in the data. As a result, a slightly variation in the collected data might result in dramatically mismatches between the current trajectory and historical data. Moreover, these algorithms tend to emphasize on the overall similarity of the whole trajectory, without considering the similarity of trajectories in individual or subsets of segments (which have different lengths). To address this issue, a solution is to take into account the similarity between two trajectories on individual segments using separate thresholds. If the difference for a segment is less than the threshold specified for the segment, the two trajectories are considered as "similar". As such, it will allow us to extend the conventional notion of trajectory similarity not only for the overall trajectory measure but also individual segment similarity measure.

However, efficiency remains an issue for comparing similar travel times in each segment, especially when the number of historical trajectories is large. To avoid exhaustive similarity comparison of all the trajectories in the repository, we partition the travel times for each segment into groups. Therefore, by matching the travel times in passed segment of an on-going bus journey against the ranges of travel times corresponding to these groups, we are able to retrieve a set of similar trajectories. To partition the segment travel time into groups, we consider two clustering methods, including K-Means [15] and V-Clustering [32]. As the K-Means algorithm is well know, here we discuss the V-Clustering algorithm which partitions the travel times on a segment to multiple groups with minimized variances.

Given the list of all the travel times on a segment L, we first sort L by their values and then recursively perform binary-partition on the sorted L into sub-lists. Basically, in each iteration, we compute the variances of all the data in L to find the "best" split point in accordance with the *minimal weighted average variance (WAV)* as defined below.

$$WAV(i;L) = \frac{|L_A^{(i)}|}{L} V(L_A^{(i)}) + \frac{|L_B^{(i)}|}{L} V(L_B^{(i)}) \qquad (1)$$

where $L_A^{(i)}$ and $L_B^{(i)}$ are two sub-lists of L split at the ith element and V represents the variance. This best split point leads to a maximum decrease of

$$\Delta V^{(i)}(L) = V(L) - WAV(i;L) \qquad (2)$$

The algorithm terminates when $max_i\{\Delta V(i)\}$ is less than a tunable threshold $Vthresh$. As a result, we can obtain a set of split points partitioning the whole list L into several groups $C = c_1, c_2, ..., c_m$ which has minimized variances.

With the groups of similar trajectories created, we are able to efficiently identify similar trajectories using the travel times on passed segments of an on-going bus journey. However, shall we consider any trajectory matching well with one of the passed segments as similar trajectory or only those that match well with all the passed segments? Shall we treat all the segments with equal weight? In HTTP, we adopt an approach, called *segment filtering*, to select similar trajectories.

Based on our data analysis, we observe that for a given segment, its correlations with nearby segments are obviously stronger than that with its faraway segments. Therefore, for a bus currently located at segment S_i, we start with the adjacent/nearest passed segment (i.e., S_{i-1}) to initialize T_r as the set of historical trajectories in the same travel time range group with the bus. Next, we filter out those trajectories not fallen in the same range group with the bus by iterating through the second segment (and then third,... and so on). In practice, the number of trajectories in T_r reduces dramatically through each iteration and thus may possibly cause the result of prediction statistically insignificant. To address this problem, we limit the number of segments, called *window of segment filtering*, in segment filtering to make sure that there are still plenty of trajectories remaining in T_r at the end. Additionally, we introduce *Minimum Number of Trajectories* (MNT) to ensure a minimum number of trajectories remaining in T_r, i.e., Segment filtering stops as soon as the number of trajectories in T_r becomes less than MNT.

Finally, with a sample set of similar trajectories T_r obtained through segment filtering, we describe the travel time estimation approach we developed for the passed segment scheme. For each trajectory $t \in T_r$, we have a travel time for Segment S_i where the bus is located. We map the travel times on S_i of T_r to the travel time range groups for S_i. Finally, we return the average travel time of trajectories in the largest group as the prediction. Additionally, let s be the number of trajectories in the largest group, we compute $\frac{s}{|T_r|}$ as the confidence of our prediction, which is also returned as a supplementary information to our prediction.

5.2 Temporal Features Scheme

In addition to the passed segments scheme, we also propose an alternative prediction method, called *temporal features* (TF) scheme, that uses "hours of a day" and "days of a week" of historical trajectories to make predictions. Without relying on information associated with previous segments, the TF scheme resorts to temporal features related to the current segment, e.g., the time when a bus enters its current segment. As shown in our analysis, it is common that congestions happen during rush hours with a high probability and the weekend and weekdays exhibit different patterns. Therefore, we first cluster the travel times of all trajectories on each segment in terms of the temporal features considered and then use the clusters to facilitate efficient retrieval of trajectories similar to the on-going bus journey .

Given a segment S_i and the collection of segment travel times on S_i, denoted as $ST_i = \{\Delta t_1, ..., \Delta t_M\}$, where M is the number of all historical trajectories passing S_i. As multiple features (i.e., time and day) are associated with the segment travel time, for each $\Delta t_j \in ST_i$, there is a feature vector, V_j, of $\langle time_i, day_i \rangle$. In this paper, using the feature vectors associated with S_i, we adopt the K-Modes algorithm [10] to partition all the trajectories passing S_i into K clusters, $C_i = \{c_0, ..., c_K\}$, where K is the specified number of clusters.[6] In a cluster, all trajectories share the same or similar features.

Finally, to determine the sample set of similar trajectories, we simply use the temporal features of current bus status to find the best matched cluster and return it as the set of sim-

[6]Specifying the number of clusters is an issue with K-Modes algorithm (and other K-Means variants). We empirically test different settings of K in our evaluation.

Parameters	Default
Data Size	12 months
Vthresh	100000
Window Size	2
MNT	200
K' in K-Mode	3

Table 1: Parameters and default values

ilar trajectories T_r. Accordingly, we use the trajectories in T_r to predict the travel time of future segments by computing the average travel time from the matched cluster as the predicted travel time.

5.3 Hybrid Schemes

The PS and TF schemes employ the same design principle but work from different angles. As they both make predictions based on historical trajectories, they can be easily combined into hybrid schemes. Therefore, we propose two different hybrid schemes to make predictions.

The first approach is to find a set of trajectories similar to the current trajectory in terms of travel times of passed segments and then filter the remaining trajectories using the temporal features cluster of the current bus journey. Thus we call this scheme *hybrid passed segments/temporal features (HPT)*. On the other hand, the second approach, called *hybrid temporal features/passed segments (HTP)*, first applies the TF scheme and then the PS scheme in clustering. Notice that the major difference between HPT and HTP lies in the process of filtering. For HPT, as it performs PS first, the segment filtering process remains the same and the matching with temporal feature clusters is executed by post-processing. On the other hand, for HTP, the clustering for both TF and PS can be pre-computed. Thus, the resulted clusters are fine-grained, capturing similar trajectories in both temporal features and passed segments.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of different prediction schemes and test various parameters. We use the data (Route ID 10283) presented in data analysis. The trajectory repository covers the whole year (March 2010 - Feb. 2011) of trajectories on the route. We use 508 trajectories in the first week of March 2011 as the testing data and ground truth to evaluate the accuracy of predictions.

A comprehensive set of experiments were performed. A number of parameters are tested in the experiments. Table 1 summarizes various parameters used in prediction schemes with their default values. Note that Vthresh denotes the threshold used in V-Clustering of the passed segment scheme. Window size (of segment filtering) and MNT denote, respectively, the number of segments filtered and the minimum number of trajectories to maintain in the segment filtering process of the passed segments scheme. K' is used for K-Mode clustering in the temporal features scheme.

We use the *prediction error* of travel times on future segments as the performance metric in our experiments. However, since the lengths of segments are different from each other, comparing the absolute values of errors in travel time is not reasonable. As we expect that the longer a segment is, the larger an error in travel time on this segment may likely be, we adopt the *normalized error* in a segment by dividing the travel time error in the segment by the distance of the segment. When we perform experiments to tune the different parameters, we use the average normalized errors in all segments as the measurement to evaluate the overall performance. On the other hand, in order to illustrate more detailed result, in some of the experiments (e.g., the comparison of prediction schemes), we show the normalized errors in terms of how many segments away a predicted segment is from the current segment (where the bus is located).

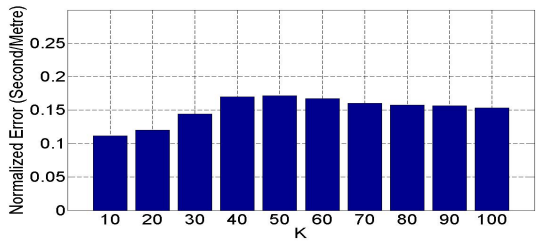
In the evaluation, we aim to evaluate the accuracy of the proposed travel time prediction schemes in HTTP, including (1) passed segments (PS) scheme; (2) temporal features (TF) scheme; (3) hybrid passed segments/temporal features (HPT) scheme, and (4) hybrid temporal features/passed segments (HTP) scheme. Additionally, we use *random prediction (RP)*, i.e., randomly select a trajectory to predict the travel time on the given segment, and *average prediction (AV)*, i.e., use average of all travel times to predict the travel time on the given segment, as the baselines for comparison. Nevertheless, as the PS and TF schemes both employ clustering algorithms to partition trajectories into groups of similar trajectories in order to facilitate efficient retrieval, we first fine tune the parameters of these two schemes. Additionally, given the sample set of similar trajectories retrieved, there are multiple methods to make a prediction based on the sample set. Thus, we next evaluate these different methods. Finally, we make comparison of all the prediction schemes in HTTP.

6.1 Tuning the Passed Segments Scheme

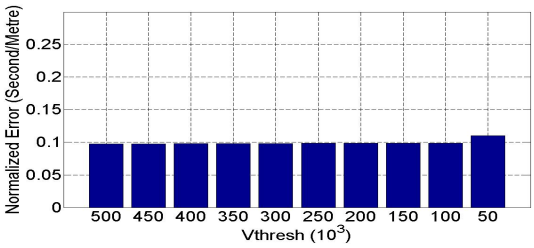
Here we first tune the PS scheme by empirically determining the suitable algorithm for trajectory partitioning and appropriate values for window size and MNT of segment filtering. Finally, we evaluate our proposed approach for travel time estimation against two alternative approaches.

Selection of Partitioning Algorithms. We would like to decide which of K-Means and V-Clustering is a more suitable partitioning algorithm for the PS scheme. Thus, we first use K-Means to cluster the trajectories based on travel time of segments. We vary K from 10 to 100 in step of 10 (with other parameters set in default) to observe the performance. Figure 4(a) shows the average normalized error of the K from 10 to 100. As for V-Clustering, the number of clusters is not explicitly determined like in K-Means but controlled by the setting of Vthresh. The larger Vthresh is, the smaller number of clusters are produced. We test V-Clustering with 10 thresholds, varying from 50000 to 500000 with step of 50000. The experiments were on all segments of the route and the number of clusters resulted for each segment are different. As shown in Figure 4(b), the performance is the worst when Vthresh is 50000 and becomes better as Vthresh increases. From Figure 4, we can tell that the PS scheme using V-Clustering obviously results in better performance than using K-Means. Therefore, we adopt 100000 as the default value for Vthresh and use V-Clustering as the default partitioning algorithm for the PS scheme.

Segment Filtering. In the PS scheme, segment filtering is used to select a set of trajectories similar to the current bus journey. In order to ensure that there is a reasonable number of similar trajectories returned for prediction, two mechanisms, namely, *window of segment filtering* (or *window* for short) and *minimal number of trajectories (MNT)*, are adopted in the PS scheme. In the following, we first



(a) Kmeans

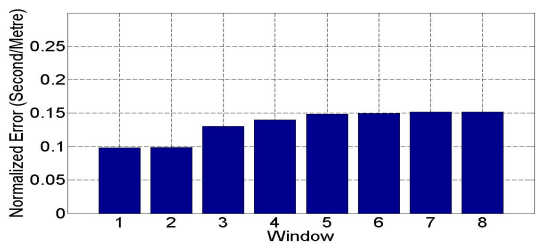


(b) VClustering

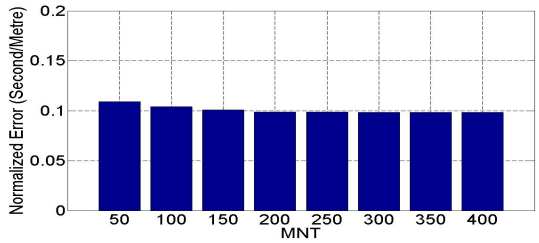
Figure 4: Partitioning Algorithms.

evaluate the impact of different window size on the performance. Based on our data analysis in Section 3, a segment correlated strongly only to its nearby segments. Therefore, instead of extending the experiment up to the maximal window size of 62 (i.e., there are 63 segments in total), we vary the window size from 1 to 8 with step of 1.

Figure 5(a) shows that the predicting performance obviously worsens when the window size of segment filtering increases to 3, i.e., the correlation between segments reduces dramatically when the number is larger than 2. As there would be less trajectories left when window size is 2 rather than 1, we take 2 as the default window size.



(a) Window size



(b) MNT

Figure 5: Segment Filtering.

Next, we study the impact of MNT which is used to ensure that there is a good number of trajectories left after segment filtering. We vary MNTs from 50 to 400 with step of 50. As illustrated in Figure 5(b), the predicting performance becomes better along with the increasing of MNT and remains pretty much that same when MNT equals to 200 or greater. Thus, we set the default for MNT as 200.

Travel Time Estimation. In the PS scheme, after retrieving a sample set of similar trajectories, we adopt a statistical approach to estimate the travel time for prediction. To validate our approach, here we evaluate it against two alternative approaches: *random prediction (RP)* and *survival analysis (SA)* [6]. As its name suggests, RA randomly selects a trajectory from the sample set T_r (generated from segment filtering) to predict the travel time on the given segment. On the other hand, SA first creates a normal distribution random generator based on T_r and then uses it to generate a travel time as the prediction.

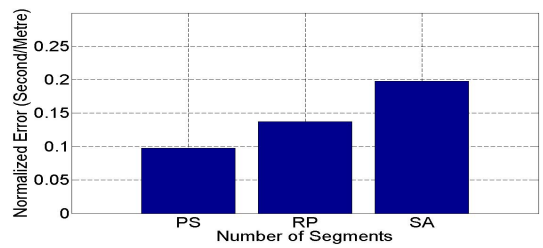


Figure 6: Travel Time Estimation

Figure 6 compares the performance of PS using RP, SA, and our statistical approach (denoted as PS). As shown, PS performs the best as both RP and SA only select a single trajectory to make prediction each time. As a result, RP and SA cannot guarantee the prediction made by such a trajectory is statistically significant.

6.2 Tuning the Temporal Feature Scheme

As mentioned in Section 5, two features are adopted in the temporal features (TF) scheme, i.e., the time when a bus is located in a certain segment, *time*, and the day of the bus journey, *day*. In TF, as those features are categorical, we clusters the trajectories by K-Modes algorithm. We perform the experiments by varying K' (which is used to distinguish from the K of K-Means algorithm) from 2 to 20 with step of 1. Figure 7 plots the performance result. As shown, the predicting error decreases slightly (at $K' = 3$) and then increases as K' increases from 2 to 20. Therefore, we choose 3 as the default number of K.

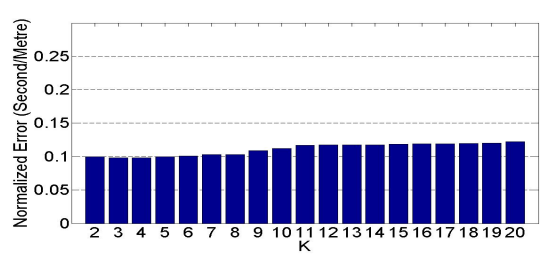


Figure 7: Tuning K-Modes

6.3 Comparison of Prediction Schemes

After tuning the PS and TF schemes, respectively, we now evaluate the performance of proposed prediction schemes along with the average prediction (AP) and TransDB, the state-of-the-art technique for travel time prediction using historical trajectories. In this experiment, we compare the schemes under evaluation by considering segment distance, i.e., the number of segments between the current segment of bus and the predicted segment. For example, for segment distance equals 1, we predict the travel time of next segment from the bus position in every bus status report and compute their average normalized error; for segment distance equals 2, we predict the travel time of segment located two segment away from the bus position in every bus status report and compute their average normalized error; and so on. Figure 8 plots the experimental result. As shown, for all prediction schemes evaluated, the prediction error increases as the segment distance increases because it's more difficult to predict segments far away from the current bus location. AP and TransDB perform obviously worse than the four methods we proposed in HTTP. Since only ONE trajectory (i.e., the NNT) is fetched to make prediction, TransDB can not guarantee the accuracy of its prediction all the time. On the other hand, while AP uses the average value of all historical data to make prediction, the result is not satisfactory as it accommodates too many different situations and consequently compromises its prediction accuracy.

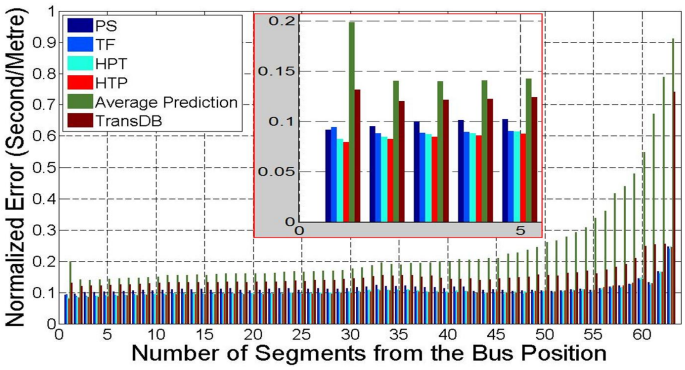


Figure 8: Comparison of Prediction Schemes

Unfortunately the differences between the four proposed schemes cannot be visualized clearly in the general plot of Figure 8. Thus, we zoom in to observe the performance in segment distance 1-5 (see the box within Figure 8; the result in other segment distances are consistent with the observation here). As shown, the two hybrid schemes, HPT and HTP, are better than PS and TP. Between PS and TP, TP is generally better than PS as it results in clusters in finer granularity (as explained earlier in Section 5.3). Between the two hybrid schemes, the HTP scheme is slightly better than HPT because each time a new travel time is received, TF is performed before PS. Therefore, the final set of trajectories that are used to estimate the travel time are similar to the current bus journey in terms of passed segments not only by travel times but also by features. On the other hand, for HPT, TF is applied to the set of trajectories obtained from PS. Therefore the returned trajectories are similar to the current bus journey in terms of features corresponding to the current segment. Somehow HTP filtering is more strict than HPT. That's why the HTP outperforms HPT.

7. CONCLUSION

In this paper, we study the problem of predicting bus arrival/travel time using historical trajectories. We propose a novel prediction framework, namely, HTTP, and develop two basic travel time prediction schemes, passed segment (PS) and temporal features (TF). The PS scheme is based on the correlation between travel times of nearby segments in the same trajectory. It identifies similar trajectories by using travel times on passed segments as the statistical basis of prediction. The TF scheme is based on the temporal patterns of bus travel times in route segments. It uses two temporal features "hours of a day" and "days of a week" to identify similar trajectories in terms of these features to make travel time prediction. As HTTP is a general framework, these two basic prediction schemes are combined to provide two hybrid prediction schemes, namely, hybrid passed segment/temporal features (HPT) and hybrid temporal features/passed segments (HTF).

We conduct a set of comprehensive experiments using a real dataset collected from buses in Taipei City, Taiwan, to evaluate the effectiveness of the proposed prediction schemes with the average prediction (AP) scheme (a baseline) and the TransDB scheme (a state-of-the-art technique for bus travel time prediction). The result shows that all the prediction schemes proposed under HTTP significantly outperforms AP and TransDB. Meanwhile, the HTP scheme performs the best, which demonstrates that a better matching of similar trajectories through a strict filtering produce excellent prediction results.

As for the future work, we plan to explore more features and patterns from the historical trajectory data. Currently, we only explore temporal features. It will be interesting to discover the correlation of segment travel time with other possible features (e.g., weather). Finally, we aim to further enhance the prediction accuracy by considering historical trajectories not only in one bus route but also those in nearby routes.

8. ACKNOWLEDGEMENT

This work was supported in part by National Science Council, Taiwan under grant no. NSC 101-2219-E-001-002.

9. REFERENCES

[1] M. D. Abkowitz. An analysis of the commuter departure time decision. *Transportation*, 10:283–297, 1981.

[2] I. Assent, M. Wichterich, R. Krieger, H. Kremer, and T. Seidl. Anticipatory DTW for efficient similarity search in time series databases. *Proceedings of the VLDB Endowment*, 2(1):826–837, 2009.

[3] F. Batool and S. Khan. Traffic estimation and real time prediction using ad hoc networks. In *Proceedings of the IEEE Symposium on Emerging Technologies*, pages 264–269, September 2005.

[4] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI Workshop on Knowledge Discovery in Databases*, pages 359–370, 1994.

[5] G. Chen, X. Yang, D. Zhang, and J. Teng. Historical travel time based bus-arrival-time prediction model. In

Proceedings of the International Conference of Chinese Transportation Professionals, pages 1493–1504, 2011.

[6] R. Elandt-Johnson and N. Johnson. *Survival Models and Data Analysis*. Wiley, 1980.

[7] H. Fashandi and A. Moghaddam. A new rotation invariant similarity measure for trajectories. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 631–634, june 2005.

[8] D. Ghosh and C. Knapp. Estimation of traffic variables using a linear model of traffic flow. *Transportation Research*, 12(6):395–402, 1978.

[9] C. Hermes, C. Wohler, K. Schenk, and F. Kummert. Long-term vehicle motion prediction. In *IEEE Intelligent Vehicles Symposium*, pages 652–657, 2009.

[10] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998.

[11] C. S. Jensen and D. Tiešytė. TransDB: GPS data management with applications in collective transport. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 1–6, 2008.

[12] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[13] R. Li and G. Rose. Incorporating uncertainty into short-term travel time predictions. *Transportation Research Part C: Emerging Technologies*, 19(6):1006–1018, 2011.

[14] W.-H. Lin and J. Zeng. An experimental study on real time bus arrival time prediction with GPS data. Technical report, Center for Transportation Research and Dept. of Civil and Environmental Engineering, Virginia Polytechnic Institute & State Univ., 2001.

[15] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[16] N. L. Nihan and K. O. Holmesland. Use of the box and jenkins time series technique in traffic forecasting. *Transportation*, 9:125–143, 1980.

[17] T. Oda. An algorithm for prediction of travel time using vehicle sensor data. In *International Conference on Road Traffic Control*, pages 40–44, may 1990.

[18] A. Polus. Modeling and measurements of bus service reliability. *Transportation Research*, 12(4):253–256, 1978.

[19] A. Polus. A study of travel time and reliability on arterial routes. *Transportation*, 8:141–151, 1979.

[20] A. Shalaby. Prediction model of bus arrival and departure times using avl and apc data. *Journal of Pub. Transp*, pages 41–61, 2004.

[21] T. Sumi, Y. Matsumoto, and Y. Miyaki. Departure time and route choice of commuters on mass transit systems. *Transportation Research Part B: Methodological*, 24(4):247–262, 1990.

[22] D. Tiesyte and C. S. Jensen. Similarity-based prediction of travel times for vehicles traveling on known routes. In *Proceedings of the international conference on Advances in geographic information systems*, pages 14:1–14:10, 2008.

[23] D. Tiesyte and C. S. Jensen. Assessing the predictability of scheduled-vehicle travel times. In *Proceedings of the International Conference on Advances in Geographic Information Systems*, pages 416–419, 2009.

[24] J. W. C. van Lint. Reliable real-time framework for short-term freeway travel time prediction. *Journal of Transportation Engineering*, 132(12):921–932, 2006.

[25] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multidimensional time-series. *The Very Large Data Base Journal*, 15(1):1–20, 2006.

[26] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings. the International Conference on Data Engineering*, pages 673–684, 2002.

[27] C.-H. Wu, J.-M. Ho, and D. Lee. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):276–281, December 2004.

[28] C.-H. Wu, D.-C. Su, J. Chang, C.-C. Wei, J.-M. Ho, K.-J. Lin, and D. Lee. An advanced traveler information system with emerging network technologies. In *Proceedings of the Asia-Pacific Intelligent Transportation Systems Forum*, 2003.

[29] T. Xu, O. Tomeh, and L. Sun. Urban expressway real-time traffic state estimation and travel time prediction within EKF framework. In *Proceedings of International Symposium on Transportation and Development Innovative Best Practices*, pages 192–197, 2008.

[30] B.-K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 385–394, 2000.

[31] B. Yu, Z. Yang, and Z. Yao. Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems*, 10(4):151–158, 2006.

[32] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of the International Conference on Advances in Geographic Information Systems*, pages 99–108, 2010.

[33] H. Zhu, Y. Zhu, M. Li, and L. M. Ni. Seer: Metropolitan-scale traffic perception based on lossy sensory data. In *INFOCOM*, pages 217–225, 2009.

[34] T. Zhu, F. Ma, T. Ma, and C. Li. The prediction of bus arrival time using global positioning system data and dynamic traffic information. In *IFIP Wireless and Mobile Networking Conference*, pages 1–5, 2011.

[35] Z. Zhu and W. Wang. A travel time estimation model for route guidance systems. In *International Conference on Transportation and Traffic Studies*, pages 547–552, 2000.

[36] N. Zou, J. Wang, and G.-L. Chang. A reliable hybrid prediction model for real-time travel time prediction with widely spaced detectors. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pages 91–96, October 2008.