DeepTRANS: A Deep Learning System for Public Bus Travel Time Estimation using Traffic Forecasting

Luan Tran
University of Southern California
Los Angeles, CA
luantran@usc.edu

Min Y. Mun*
Samsung Advanced Institute of Technology
Samsung Electronics, South Korea
minyoung.mun@samsung.com

Matthew Lim
University of Southern California
Los Angeles, CA
lim643@usc.edu

Jonah Yamato
University of Southern California
Los Angeles, CA
jyamato@usc.edu

Nathan Huh
University of Southern California
Los Angeles, CA
nhuh@usc.edu

Cyrus Shahabi
University of Southern California
Los Angeles, CA
shahabi@usc.edu

ABSTRACT

In the public transportation domain, accurate estimation of travel times helps to manage rider expectations as well as to provide a powerful tool for transportation agencies to coordinate the public transport vehicles. Although many statistical and machine learning methods have been proposed to estimate travel times, none of the methods consider utilizing predicted traffic information. Forecasting how congestion is going to evolve is critical for accurate travel time estimations. In this paper, we present DeepTRANS, which incorporates traffic forecasting information to our prior Deep Learning-based Bus Estimated Time of Arrival (ETA) model, increasing its accuracy by 21% in estimating bus travel time.

PVLDB Reference Format:
Luan Tran, Min Y. Mun, Matthew Lim, Jonah Yamato, Nathan Huh, Cyrus Shahabi. DeepTRANS: A Deep Learning System for Public Bus Travel Time Estimation using Traffic Forecasting. *PVLDB*, 13(12): 2957-2960, 2020.
DOI: https://doi.org/10.14778/3415478.3415518

1. INTRODUCTION

According to the 2019 Urban Mobility Report,[1] 494 metropolitan areas in the U.S experienced 8.8 billion vehicle-hours of delay in 2017, which is equivalent to 3.3 billion gallons in wasted fuel and $179 billion in lost productivity. Therefore, increasing ridership of public transportation and hence reducing traffic congestion has been one of the primary objectives for transportation agencies and policymakers. In the public transportation domain, accurate estimation of travel times helps to manage rider expectations (e.g., will the bus be on time in the next 30 minutes?). It also provides a powerful tool for transportation agencies to coordinate public transport vehicles. Many statistical and machine learning methods have been proposed to estimate travel times, e.g., Support Vector Machine [5] and Recurrent Neural Network [6, 2]. While most recent methods strive to improve their estimation by incorporating current traffic situation [5], none have utilized the predicted traffic information.

In this paper, we propose DeepTRANS, the first data-driven, deep learning system incorporating traffic flow forecasting for bus arrival time estimation. Specifically, we plan to extend a current state-of-the-art model in bus ETA computation, DeepTTE [5], with our prior work in traffic forecasting [4] to improve its accuracy. The DeepTTE model employs a Geo-Convolution (Geo-Conv) layer to embed longitude and latitude of locations and Long Short-Term Memory (LSTM) layers to capture the temporal dependency between them. Our prior work models traffic flow as a diffusion process on a directed graph and introduces Diffusion Convolutional Recurrent Neural Network (DCRNN), a deep learning framework for traffic forecasting that considers both spatial and temporal dependency in the traffic flow. We concatenate the traffic flow predictions produced by our DCRNN model with the output vectors of the Geo-Conv layer in DeepTTE. Using the real-world traffic sensor datasets archived in our data warehouse [1], we show that our proposed Bus ETA model is more accurate than the existing method without traffic predictions by 21% in estimating bus travel times. Finally, we deployed our Bus ETA model as a web application so that users can check bus arrival times between a starting and destination bus station.

We demonstrate DeepTRANS in two scenarios: (1) We allow the conference attendees to select a bus route in Los Angeles. Then for the selected bus route, we ask them to choose starting and destination stations with a starting time. The starting time can be a current or future time for which the ground truth information is not available yet. Subsequently, we show the ETA of DeepTRANS with those of the other approaches as well as the bus schedule. (2) With the selected bus route, we ask the attendees to select a starting time in the past. By comparing the predicted measurements with ground truth information, the attendee

*The work was conducted and completed while the author was a research staff at USC's Integrated Media Systems Center (IMSC).

[1]https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-report-2019.pdf

Proceedings of the VLDB Endowment, Vol. 13, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3415478.3415518

can check how accurately our proposed method estimated the bus arrival times.

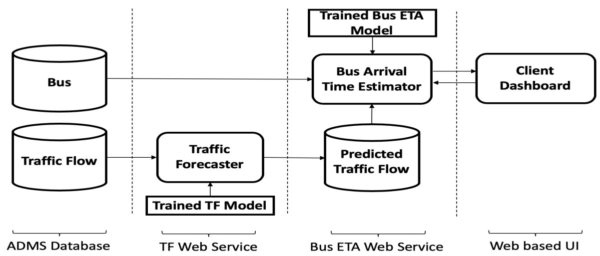

Figure 1: DeepTRANS System Architecture

2. SYSTEM OVERVIEW

DeepTRANS is a complex system optimized for public transportation ETA computation. It takes a trajectory with a starting time as input and returns the ETA for each stop on the trajectory. Given a spatial domain $\sum$, each trajectory t consists of a sequence of locations ($_i$), where $_i \in \sum$ for $i = 1, 2, \ldots$ The spatial domain is a set of longitude-latitude coordinates. We divide the times in a day into bins of five minutes. The input starting time is the bin corresponding to the time observed at the first location $_1$ of the trajectory. Figure 1 shows the system architecture of DeepTRANS which consists of four modules - 1) ADMS Database, 2) Traffic Forecasting (TF) Web Service, 3) Bus ETA Web Service, and 4) Web-based User Interface (UI).

2.1 ADMS Database

With our partnership with the Los Angeles Metropolitan Transportation Authority, we developed a large transportation data warehouse called Archived Traffic Data Management System (ADMS) [1]. The ADMS Database stores the bus and traffic flow information. This dataset includes both sensor metadata and real-time data for freeway and arterial traffic sensors (approximately 16,000 loop-detectors) covering 4,300 miles and 2,000 bus and train automatic vehicle locations with an update rate as high as 30 seconds. We have been continuously collecting and archiving the datasets mentioned above for the past 8 years. These datasets are used to train and test our traffic forecasting and Bus ETA models.

2.2 Traffic Forecasting Web Service

We implemented our Diffusion Convolutional Recurrent Neural Network (DCRNN) [4] to forecast the traffic flow one hour in advance. DCRNN takes the graph of traffic sensors with speed information in the last hour as the input and outputs traffic flow in the next one hour.

Large Scale Model Training. Training DCRNN with an extensive road network poses two main challenges. First, the training data size for thousands of locations is often too large to fit in the main memory of a single machine. Second, the time required for training the DCRNN on a large dataset can be prohibitive, rendering the method ineffective for large road networks. To overcome these challenges, we split the traffic sensor set into 14 non-overlapping subsets. Each subset contains spatially close traffic sensors. Consequently, we train 14 DCRNN models separately.

Tra c Flow Estimation for Trajectories. The DCRNN model outputs the predicted speed information at the locations of traffic sensors. However, there are many locations in a bus trajectory where sensor data is not available. Therefore, we estimate the speed at each location in a trajectory using the speed values of its neighboring sensors. More specifically, for each location, we select k nearest traffic sensors and compute the weighted average of the predicted speed values at the locations of those sensors. The weights are computed by taking the reciprocal of the distance values from the location to its neighboring traffic sensors, thus, closer sensors have higher weights than farther sensors. Let o be the location that we want to estimate speed from its k neighboring traffic sensors $s_1, s_2, \ldots, s_k$ with their corresponding distances $d_1, d_2, \ldots, d_k$. The speed at o is calculated as follows:

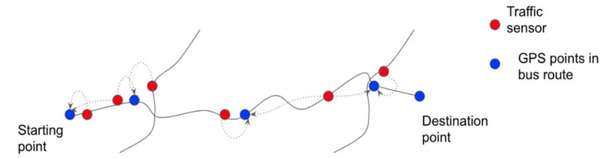

Figure 2: Estimating speed at locations using two neighboring tra c sensors

$$speed(o) = \frac{\frac{s_1}{d_1} + \frac{s_2}{d_2} + \cdots + \frac{s_k}{d_k}}{\frac{1}{d_1} + \frac{1}{d_2} + \cdots + \frac{1}{d_k}}$$

Figure 2 illustrates the speed estimation when k is 2.

Finally, the traffic flow estimator is invoked by the Traffic Forecasting Web Service every five minutes to update predicted traffic flow measurements and the results are stored to be used by the Bus ETA Web Service. Continuously storing prediction results allows the Bus ETA Web Service to quickly retrieve the predicted traffic flow information on all the bus routes in the next hour.

2.3 Bus ETA Web Service

Previous Bus ETA approaches only considered historical data to estimate future travel times. Traffic forecasting information is essential for more accurately estimating when a bus will arrive at the next stops as it takes into account future traffic congestion. We incorporate traffic predictions into the state-of-the-art Bus ETA model [8].

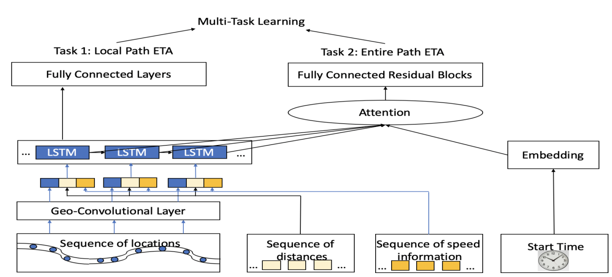

Figure 3: Bus ETA Architecture

Incorporating Tra c Flow to Bus ETA Model. The authors in [8] use a Geo-Convolution layer (Geo-Conv) to convert a sequence of location data into vectors and a Long Short-Term Memory (LSTM) layer to capture the temporal dependencies of the locations. More specifically, the Geo-Conv layer converts a pair of longitude and latitude of a location into a point in the 16-dimensional space.

Table 1: MAPE Comparison to the Ground Truth

DeepTRANS	LR	AVG	SVR	GBDT	DeepTTE
26%	79%	51%	39%	36%	33%

For each location p_i in a trajectory, it is a non-linear mapping: $loc_i = \tanh(W_{loc} \cdot [p_i.lat \circ p_i.lon])$ where $\circ$ is the concatenate operation and W_{loc} is a learnable matrix. They apply a convolutional operation on the sequence loc along with a 1-dimensional sliding window: $loc_i^{conv} = \sigma_{cnn}(W_{conv} * loc_{i:i+k-1} + b)$, where $*$ is the convolution operation, σ_{cnn} is an activation function, $loc_{i:i+k-1}$ is a subsequence of loc, and b is a bias term. This method is known to be more useful for capturing spatial dependencies of GPS points than traditional CNN [8]. We studied many different approaches to incorporate the speed values, e.g., concatenating them with the input vector of the Geo-Conv layer or the output vector of the LSTM layer. We obtained the best performance when we added the speed values along with the distances from the starting location to the output of the Geo-Conv layer. This can be formulated as follows: $lsd_i = loc_i^{conv} \circ d_i \circ s_i$, where d_i is the distance from the starting location and s_i is the speed information of a location in a trajectory. After this step, each location in the trajectory is represented by a vector, including embedded spatial information, the distance from the starting location, and the predicted speed information. Subsequently, we used the vector lsd as the input to the LSTM layer. Finally, the output of the LSTM layer and the embedding vector of the start time are passed through fully connected residual blocks with an attention mechanism. The architecture of our Bus ETA model is illustated in Figure 3.

Multi-task training. We use a multi-task learning approach, similarly to [8], to combine the local and entire path predictions. Let L_1 be the loss of the local path predictions and L_2 be the loss of the entire path prediction. The loss that our model strives to minimize is: $L = \beta L_1 + (1 - \beta) L_2$, where $\beta \in [0, 1]$ is the combination coefficient that balances the trade-off between L_1 and L_2. In our experiments, the parameter β was set to 10%.

Finally, we deployed our Bus ETA model as a web service that communicates with the Web-based UI. The Bus ETA Web Service receives requests from the Web-based UI with the input, including a trajectory and starting time. Next, it loads the trained Bus ETA model, runs predictions on the estimated travel time to each bus stop in a trajectory, and returns the prediction measurements to the Web-based UI.

2.4 Web-based User Interface (UI)

We developed a Web-based User Interface that presents the information of bus routes and bus stops, and allows users to plan a trip in the future with the information of predicted bus arrival times to every bus stop on a selected bus route. The requests will be sent to the Bus ETA Web Service to retrieve ETA information.

3. EXPERIMENTS

We selected 62,000 bus trajectories from our data repository (ADMS) collected for two months (April 2017, September 2017) in Los Angeles. This data contains location (longitude and latitude), timestamp, and vehicle ID information. It allows us to compute ground truth speed values of each GPS point and travel times of bus stops for evaluation. Out of 62,000 trajectories, 50,000 were used for training our Bus ETA model, and the rest were used for evaluation. The mean of each trajectory travel time is 1,344sec. The mean distance is 21km; most of the trajectories in the test data are shorter than 20km.

We compared our proposed approach, DeepTRANS, with several baselines: 1) Linear Regression (LR) [7], 2) AVG which uses the average speed in the past during the same period, 3) Support Vector Regression Machine (SVR) [3], 4) Gradient Boosting Decision Tree (GBDT), and 5) DeepTTE, a state-of-the-art Deep learning-based Bus ETA model [8]. Table 1 shows the comparison of different approaches for estimating travel times. These methods were evaluated based on a commonly used metric in bus ETA computation, Mean Absolute Percentage Error (MAPE), compared to the ground truth. We had the following observations. (1) LSTM-based methods such as DeepTRANS and DeepTTE generally outperformed other baselines, which emphasizes the importance of modeling the temporal dependency. (2) DeepTRANS improved DeepTTE 21% in MAPE, which proves the effectiveness of utilizing the traffic forecasting results.

4. DEMONSTRATION

We used real-time traffic flow and bus data in Los Angeles, including 136 bus routes with 14,003 bus stops. The data have been collected from 03/20/2018 to 04/01/2020 and is continuously updated. The traffic flow information is also continuously collected from approximately 16,000 traffic sensors in Los Angeles.

4.1 User Interface Overview

Figure 4 presents the user interface of our system. Figure 4(a) shows the front page with the overview information of all bus routes in Los Angeles. The left sidebar contains a table displaying every bus route's name, and the right sidebar shows the statistical information of all bus routes. The map has multiple lines representing all of the bus routes. It allows the user to zoom in or out the map view to explore different levels of details. When a bus route from the left sidebar is selected, the map updates to show only the selected route. The left-hand table updates to display the list of the bus stops in the route, and the right sidebar shows the statistics of the selected bus route. The user can select the start and destination bus stations and the start time of travel, as in Figure 4(b). Then by clicking the "Estimate" button, the dashboard sends requests to the Bus ETA Web Service and displays the estimated arrival times for each bus stop in the truncated route on the left-hand table. The center map reflects the table and only shows the route from the start to the end bus station. When the user clicks each bus stop in the left-hand table, the right sidebar shows the statistical information specific to the bus stop.

4.2 Demonstration Scenarios

DeepTRANS can be used for everyday travel. For example, a user may be planning their commute from their home to their work office. The user can select a bus route, followed by the starting and ending bus stops. Then they can then select the time to leave or keep the leaving time with the default option "Leave Now", which takes the nearest bus schedule of the chosen starting bus stop as the leaving time. Consequently, the user will obtain the ETA at each bus stop along the route. Thus, the user can plan their trip ahead of time.

Alternatively, a transportation agency can verify the performance of DeepTRANS by checking the algorithms'