Dynamic Bus Arrival Time Prediction: A temporal difference learning approach

Vignesh L K P*, Avinash Achar* and Gokul Karthik
Research and Innovation, Tata Consultancy Services, Chennai, India.
{*vignesh.lkp, achar.avinash, gokul.karthik*}@*tcs.com*

***Abstract*—Public transport buses suffer travel time uncertainties owing to diverse factors such as dwell times at bus stops, signals, seasonal variations and fluctuating travel demands etc. Traffic in the developing world in particular is afflicted by additional factors like lack of lane discipline, diverse modes of transport and excess vehicles. The bus travel time prediction problem on account of these factors continues to remain a demanding problem especially in developing countries. The current work proposes a method to address bus travel time prediction in real-time. The central idea of our method is to recast the dynamic prediction problem as a value-function prediction problem under a suitably constructed Markov reward process (MRP). Once recast as an MRP, we explore a family of value-function predictors using temporal-difference (TD) learning for bus prediction. Existing approaches build supervised models either by (a)training based on travel time targets only between successive bus-stops while keeping the no. of models linear in the number of bus-stops OR (b)training a single model which predicts between any two bus-stops while ignoring the huge variation in the travel-time targets during training. Our TD-based approach attempts to strike an optimal balance between the above two class of approaches by training with travel-time targets between any two bus-stops while keeping the number of models (approximately) linear in the number of bus-stops. It also keeps a check on the variation in the travel-time targets. Our extensive experimental results vindicate the efficacy of the proposed method. The method exhibits comparable or superior prediction performance on mid-length and long-length routes compared to the state-of-the art.**

***Index Terms*—Travel Time Prediction; Markov reward process; temporal-difference learning; Non-linear function approximation;**

I. Introduction

Given the steady migration of people from rural pockets into urban areas (especially in developing countries), city populations are exploding leading to increased traffic volumes across the globe. Improving the quality of public transport is one sustainable approach to mitigate this ubiquitous problem. This can potentially not only reduce traffic volumes and congestion in turn, but also curtail pollution. To make public transport an attractive option, it needs to be reliable. This will require maintaining bus schedules to the extent possible while providing accurate arrival time predictions. Accurate arrival time estimates can help commuters waiting or planning to board on time at a specific bus stop. Accurate Bus Arrival time prediction (BATP) can also help travellers plan or decide to take a bus headed to a desired destination. BATP can also aid transport authorities to administer necessary corrective actions in real-time in order to maintain the overall quality of the transport system. Hence providing quality BATPs is a significant aspect of reliable public transport.

Research on BATP is about two decades old in the developed world. It is arguably a relatively well solved problem in developed countries. However, in the context of developing countries, BATP is far from solved and there is substantial scope for improvement. The primary reasons for this include (1)non-adherence to lane discipline (2) in-homogeneity of traffic (i.e. transport modes can range from bicycles, two wheelers to heavy vehicles like buses and trucks) with no dedicated lanes for specific modes of transport. We compactly refer to this as *mixed traffic* conditions. The real data considered in this paper is collected on a bus route from an Indian metropolitan city[1] where such mixed traffic conditions exist. Currently, public transport based bus travel time predictions in real time via Google maps are mostly absent in India. Any travel time query on google maps in most Indian cities returns a mostly static travel time value irrespective of the time or date of query. The travel times returned are almost fixed and seem to be based on some pre-fixed schedules. Given the heterogeneous and complex nature of traffic in developing economies like India, bus schedules are currently very hard to implement. Also, to the best of our knowledge there do not exist any mobile app-based systems which can give real-time predictions of the currently active buses in the Indian city considered in this paper. Given all these various factors, BATP is still an active research problem in general [1] and in particular under mixed traffic conditions [2].

Gaps and Contributions: Over the years, researchers have proposed a variety of approaches for BATP. Among these, the data driven class of methods has been very prominent. Most of these methods segment the entire route into smaller sections either uniformly [3] OR based on segments connecting successive bus-stops. The data input in each of these various methods can range from different quantities like speed, flow, travel time and so on depending on the algorithm and the sensing infrastructure in place. There has been a significant class of methods which stick to travel time experienced at each

*Both authors contributed equally to this work.

[1]Due to confidentiality issues, we are not in a position to reveal the name of the city in this paper.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

of these sections/segments as the primary data input to the prediction algorithm. In this paper, we stick to this approach. The observed section travel times, include both the dwell time (at the bus-stops in the section) and the actual running time across the section. This data for instance can be obtained by high-quality GPS sensing.

Existing data-driven methods have adopted a diverse range of techniques which include ARIMA models [4], linear statistical models like Kalman filters [5], nonlinear Kalman filters [6], support vector machines [7], [8], feed-forward ANNs [9], recurrent neural networks [10], CNNs [11] and so on. Most of the existing approaches suffer from a variety of issues like (i)not exploiting the historical data well enough for model calibration [3], [12], [13] OR (ii)fail to capture spatial correlations [8], [14] (iii) do not exploit the temporal correlations([9], [10]) (iv)do not exploit the current real-time information sufficiently([15]). The earliest work (to the best of our knowledge) on exploiting spatio-temporal correlations was in [7], while there has been some very recent work [5], [6], [11] as well in this direction. The current work is a unique attempt to capture the spatio-temporal correlations in a fashion distinct from the above methods. Specifically:

- Unlike other methods, the proposed approach builds models by training on travel-time targets between arbitrary sections while keeping a check on variation in the target travel times and the number of models built (refer to Sec. II-B for details).
- We construct an associated Markov reward process to reformulate the dynamic bus travel time prediction problem as a value function prediction problem under the MRP.
- This reformulation allows us to explore a family of predictors based on temporal difference learning [16] unlike any of the other approaches. In particular, we explore the n-step TD and the TD(λ) class of predictors for training. Towards final prediction, we choose the best of the predictors in these classes.
- The nature of the state in the Markov process is naturally continuous, which warrants use of function approximation (ANN based) for learning. The state definition is made sufficiently rich to account for various features influencing travel-time prediction.
- The effectiveness of the proposed method is demonstrated on real data collected in a mixed traffic condition. Our experiments indicate how the proposed method outperforms a range of existing methods by at least 17% (in the best case) over routes of mid and long length ranges.

II. Literature Review and Related Work

Research on BATP over the years has seen diverse approaches. Researchers have considered diverse data inputs like speed, travel times, flow information, weather, scheduled time tables [17], crowd-sourced data [18] and so on. The computational landscape can be broadly divided into two categories: (i)traffic-theory based (ii)data-driven. Given that the current work is a purely data-driven approach, we elaborate only on the data-driven approaches further.

A. *Data-driven methods*

Data-driven methods typically impose a coarse model (based on measurable entities) that is just enough for the prediction problem at hand instead of modeling based on underlying physics of the traffic. There exist a subclass of methods which employ a data-based model but do not employ a full-fledged learning based on historical data. The remaining approaches typically use past historical data to learn the necessary parameters of a suitable prediction model before employing it to perform real-time predictions.

Without Learning: Amongst the class of methods which don't employ explicit historical-data based learning, [19], arguably one of the earliest Kalman filter based approaches, uses the previous bus travel times and the previous day travel times experienced at the same time as data inputs. The method exploits only temporal correlations while the choice of parameters of the state space model looks adhoc. The later methods which use a linear state-space model capturing travel times, *calibrate* (or fix) the data-based model parameters for the current bus just before prediction, either based on (a) the current data of the previous bus [3], [12] or (b)a suitably optimal travel-time data vector from the historical data-base [20].

Historical-Data based learning: Among the learning based methods, there are methods which do not factor enough the current real-time position OR the travel-time information of the bus in question (current). For instance, approaches based on support vector regression [8] and a feed-forward ANN [14] have been proposed which exploit only the temporal correlations that come from multiple previous bus travel times. [15] considers a SVR based prediction scheme with static inputs like link length and dynamic inputs like speed, rate of road usage. However it doesn't factor the position of the current bus OR exploit the previous bus information. [21] considers an ANN based prediction with arrival time, dwell time and schedule adherence as inputs without using any previous bus information. the previous bus information. [7] consider a dynamic prediction scheme where the travel time of the subsequent section is predicted using an SVR. The inputs include previous bus's travel time at the same section and the current bus's travel time at the previous. This prediction scheme utilizes both temporal and spatial characteristics of the data in a minimal way. [9] learns a one-hidden layer feed-forward ANN based on current time and bus position to predict arrival times across all subsequent stops. Since it learns one model to predict travel times between any two bus stops on the route, dynamic range of the target travel time will be very large in this method. An LSTM based RNN approach was proposed in [10] which basically uses a many to many architecture for training and captures the spatial correlations in the data. A recent approach capturing spatio-temporal correlations based on linear statistical models (non-stationary) was proposed in [5] which uses a kalman filter for prediction across sections. A non-linear extension of this idea was carried out in [6] which employed an extended kalman filter for prediction after an

SVR based learning. A CNN based approach was proposed in [11] recently where the conditional predictive distributions were parameterized using masked-CNNs. The CNN output modelled the travel-time between any two bus-stops in a quantized form.

B. Relevance of the Contribution

Overall one can deduce that existing methods either do not exploit historical data for learning OR do not sufficiently capture the spatial and temporal correlations for prediction. Even if some methods exploit the spatial and temporal correlations in different ways [5]–[7], [11], the models built during training by any of them only use travel-time targets either across a section OR between two consecutive bus-stops only. This aspect of these methods leads to the number of models built by any of them to be just linear in the number of sections. Since travel-time prediction needs to be carried out between any two bus-stops, this strategy of first predicting between consecutive bus-stops (OR individual sections) and then aggregating the predictions can potentially accumulate errors. An approach which partly addresses this issue is by training using travel-time targets between arbitrary bus-stops directly and was considered in [9]. This method also builds a single compact model for prediction between arbitrary bus-stops. However, this method ignores the huge variation in the target travel times during training which can lead to poor performance on routes of large and short lengths.

The proposed method strikes a balance between these two approaches by training with travel time targets between arbitrary sections, keeping the number of trained models (approximately) linear in the number of bus-stops while also not allowing the range in the travel-time targets to be too large. Another strong distinguishing feature of our method includes posing the prediction problem as a value function prediction of a suitably constructed Markov reward process. This recast crucially enables us to explore a spectrum of predictors based on temporal-difference learning for superior prediction as compared to standard supervised learning. This aspect of the current work is significantly novel compared to existing supervised learning approaches for BATP.

III. Proposed Method

Data Input: A bus route can be segmented uniformly into sections/segments and the travel times experienced to traverse each of these sections in a trip constitute the input data. The segment length could be minimum of the distance between any two consecutive bus-stops. The travel time across a section includes running time and dwell times at bus-stops.

A. Markov Reward Process formulation:

Given the N_{sc} number of sections, we need predictions between any two sections i and j. While one can build separate models for each i and j, the number of models needed would grow quadratically with the number of sections. The data may also not be rich enough to build so many such accurate models. On the other hand, there have been methods which build a single model to predict the travel time between any two sections. One can strike a balance between these two extreme approaches and instead build a single model for every destination section j. As described later in Sec. IV-C and Alg. 1, we slightly relax this strategy and consider multiple models (depending on the value of j but not more than 5) for each destination section. The no. of models built is approximately linear in the number of sections.

For the current problem, one can construct a Markov process based on the current position and the trip elapsed time. Based on this, the state has two components: (a)section number the bus is about to enter (k) (b)time elapsed to reach the current position from the start (T_{el}). The state (k, T_{el}) evolves as the bus moves ahead towards the destination j. Specifically, as the bus moves from section k to section $k+1$ (for any $k \leq j$ experiencing a section travel time of T_k, the current state $s = (k, T_{el})$ changes to the next state $s' = (k+1, T'_{el})$, where $T'_{el} = T_{el} + T_k$. Fig. 1 indicates the state transition pictorially. For a fixed destination section j, a state with section number $j+1$ indicates the bus has reached its destination section. Hence, any such state would be a terminal or an absorbing state irrespective of the value of the second field, trip elapsed time. The MRP performs exactly j state transitions before entering the terminal state $(j+1, T_{el})$ and from then on it continues to remain in this state with zero terminal reward as is the case in standard episodic MRPs [22] or the equivalent stochastic shortest path MRPs [23].

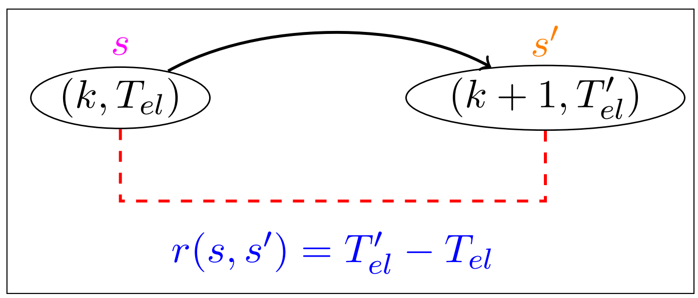


Fig. 1. State Transition Structure

The first component transitions are deterministic as the section number increments by one for every transition. The second component transition is stochastic as the section traversal times have uncertainties involved. We assume the trip elapsed time to reach section $k+1$ given the elapsed time to reach k is probabilistically independent of all past trip elapsed times. This assumption vindicates the Markovian nature of the process. For every such state transition (s, s'), we define the reward to be the difference of the trip elapsed times at these two states (Fig. 1). Hence we now have a MRP.

The value function of a MRP starting at a state s is the expected long-term sum of rewards if the process starts starts at this state s. Based on our reward definition, it is easy to see that the value function at any state is the expected time to reach the destination section j. This is the desired quantity of prediction interest.

$$V(s) = E\left(\sum_{i=0}^{\infty} r(s_i, s_{i+1})/s_0 = s\right) \quad (1)$$

Eqn. 1 even though is an infinite sum for general episodic MRPs, is a finite summation in our setting. The number of non-zero terms starting from any state is at most j.

B. TD based Learning

If the state space is finite and not too large, then the value function prediction is relatively straightforward as one needs to estimate the value function for each state based on data. When the state space is finite but very large OR infinite and continuous, employing function approximation is a typical work-around to mitigate this curse of dimensionality. Essentially the tabular approach of storing the value function for each state value breaks down. Under a linear function approximation idea, each state s is mapped to real valued feature vector $\Phi(s)$ and the value function $V(s)$ is hypothesized to be a linear combination of the features. Its a linear approximation since the value function is a linear function of the unknown weights.

$$V(s) = \mathbf{w}^T\Phi(s), \text{where } \Phi(s) = [\phi_1(s), \dots \phi_N(s)] \quad (2)$$

Instead of learning tabular entries (value function values for each state) directly, function approximation methods learn a weight vector which indirectly store the entire value function possibly for a infinite or continuous state space. For each state, the value is not prestored but is computed on the fly when presented with the state and its associated feature vector. In the current application, the second component of the state is the elapsed time to reach the current position. It seems most appropriate to leave the continuous nature of this entity as it is than trying to quantize this. Also, we don't have a system or environment here to interact with which can generate arbitrary amount of data for a fixed policy. We essentially have a fixed policy based on which we have finite limited data. Both these issues *in addition to limited data* compel us to use function approximation. Function approximation also enables generalization of the value function across states (which assume a continuum of values here) based on finite data.

Monte-Carlo based and one-step ahead temporal-difference learning methods are two extreme approaches to learn the value function. Both approaches essentially minimize the same training error and employ a stochastic gradient weight update as follows.

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\left[U_t - \widehat{V}(s_t, \mathbf{w}_t)\right]\nabla\widehat{V}(s_t, \mathbf{w}_t) \quad (3)$$

where $\mathbf{w}$ can be the weights of Eqn. 2 if a linear function approximation is employed OR can be the weights of Figure 2 for a nonlinear approximation.

However, both methods differ in the manner in which the targets U_t are defined. In Monte Carlo learning, U_t is the sum of rewards till the end of the episode. It is also referred to as return and denoted as G_t.

$$G_t = r(s_t, s_{t+1}) + \cdots + r(s_{T-1}, s_T) \quad (4)$$

In one-step TD, the target is a bootstrap estimate computed as

$$U_t = r(s_t, s_{t+1}) + \widehat{V}(s_{t+1}, w_t). \quad (5)$$

The estimate is based on the immediate reward and the bootstrapped estimate of the value function of the next state, which is motivated directly from the one-step Bellman equation [23].

The n-step TD methods provides a discrete range of learning methods where one-step TD and Monte-Carlo methods are at two extremes. The n-step return ($G_{t:t+n}$) is defined as follows.

$$G_{t:t+n} = r(s_t, s_{t+1}) + \cdots + r(s_{t+n-1}, s_{t+n}) + \widehat{V}(s_{t+n}, \mathbf{w}_t) \quad (6)$$

The associated target U_t in Eqn. 3 is set to $G_{t:t+n}$. This is the first class in which we search for new predictors superior to one-step TD and Monte-Carlo.

One can further generalize by considering the average of n-step returns as the target. One such weighted average of all n-step returns leads to the TD(λ) bag of predictors [16]. Specifically, each n-step return is weighted proportional to λ^{n-1}, where $\lambda \in [0, 1]$. This variation in λ is what leads to a continuous range of predictors. The associated return is defined as follows.

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_{t:t+n} \quad (7)$$

For an episodic MRP the above equation can be simplified as follows.

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_{t:t+n} + \lambda^{T-t-1}G_t \quad (8)$$

From the above equation, it is easy to observe that when $\lambda = 1$, $G_t^{\lambda} = G_t$, namely the Monte-Carlo return is given by Eqn. 4. While $\lambda = 0$, $G_t^{\lambda} = G_{t:t+1}$, the one-step TD return. Hence Monte-Carlo and one-step TD are at the two extremes of the λ-return based predictors.

The above λ-version based predictors need to be implemented offline only at the end of an episode. In our case, since we are learning from historical data the above offline version can also be used. In this paper, we explore its online version namely the TD(λ) algorithm.

It involves the following steps at each iteration.

$$\mathbf{z}_{-1} = \mathbf{0} \quad (9)$$

$$\mathbf{z}_t = \lambda\mathbf{z}_{t-1} + \nabla\widehat{V}(s_t, \mathbf{w}_t), 0 \le t \le T. \quad (10)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha\delta_t\mathbf{z}_t. \quad (11)$$

where δ_t is the one-step TD-error at s_t, i.e. $(U_t - \widehat{V}(s_t, \mathbf{w}_t))$, where U_t comes from Eqn. 5.

In general, λ OR n control the bias-variance trade off and an intermediate value of n OR λ are known to work best [24] in terms of prediction performance (mean square error for instance).

C. Improving predictions

Towards improving the predictive model, we enlarge the state definition by adding additional features. In particular, the complete set of input features are indicated in Figure 2. The extra features include (i)T_{pv} - the time to reach destination section from the current position based on currently plying most recent previous buses which traversed the subsequent sections. The point to note is that the most recent bus might not have crossed the destination section in real-time. In which case, we resort to the next previous bus's section travel time information to compute the destination time based on real-time information of the previous buses. The other time feature (ii)T_{pw}, we have used is based on a (closest) historical trip that happened in the previous week, same weekday. The connotation of 'closest' here is with respect to the trip start times. The current time OR the time of day is also an input. However, its encoded as a categorical feature of up to 4-5 levels which capture periods like morning peak, morning off-peak, evening peak and evening off-peak hours. The last additional feature includes the week of day which distinguishes between Monday, Friday, Saturday and the rest of the week days (Tue-Thu).

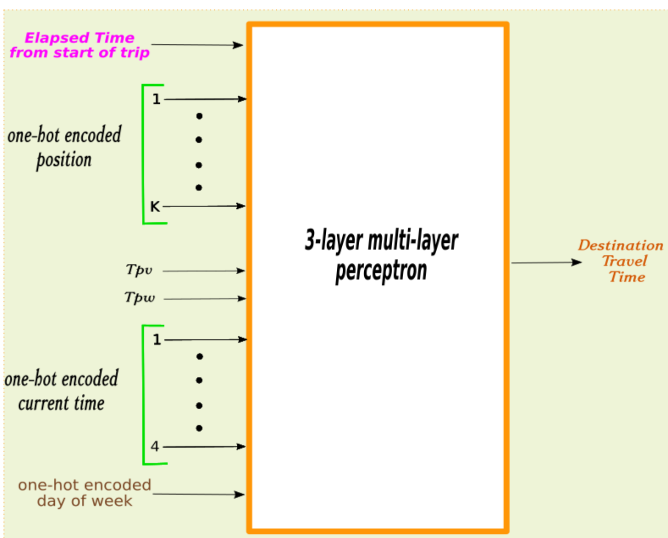


Fig. 2. Value Function Approximation

D. Feature Construction

In our search for the right n or the optimal λ, we need to perform a grid search over the step-size α for each n and λ. Performing this search with full non-linear function approximation can lead to large learning times and possibly unstable learning. To address this issue, we propose a heuristic to fix or choose the non-linear features. The chosen non-linear features can now be used with linear function approximation while searching for the optimal n or λ. TD with linear function approximation (in-spite of complicated features) will be faster as the gradient computation is simple and hence doesn't need back-propagation and is generally more stable. *To choose these*

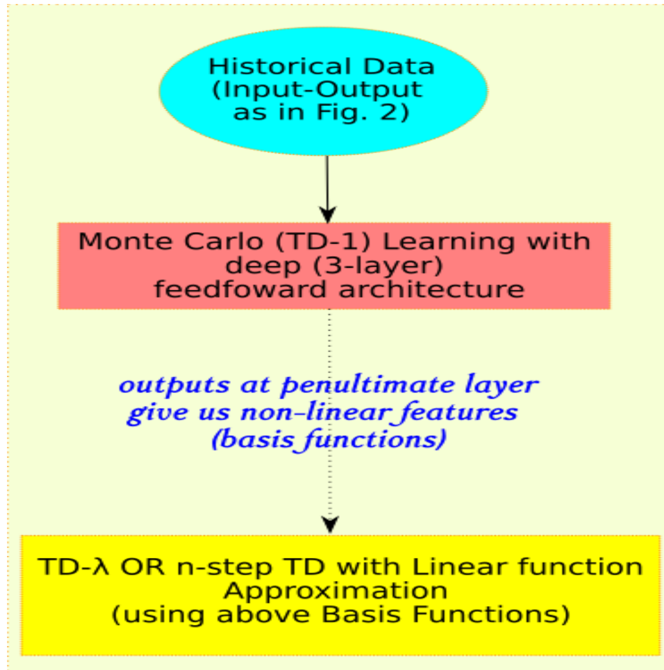


Fig. 3. Feature Extraction

Algorithm 1: Overall Method

I/P : Travel times across all sections (N_{sec} of them), across all trips and days, K (chosen as 15 here)
O/P: $\mathcal{M}$ - set of all predictive models (For each destination section j, at most N_{sec}/K models. Together one can predict from any section preceding j.)

Initialize $\mathcal{M} = \phi$;
for *every destination section* $j = 1, 2, \ldots, N_{sec}$ **do**
    if $j \leq K$ **then**
        Form one group $\mathcal{G}$ of j sections;
    else
        Divide the j sections into partitions of $K = 15$ consecutive sections (starting from the first section); ▷ `(When` $j \pmod K \neq 0$`, then the last group will have less than` K `sections)`
        Denote by $\mathcal{G}$, the class of $\lceil j/k \rceil$ such partitions;
    for *every partition* p *in* $\mathcal{G}$ **do**
        Build a *single Markov reward process* model with j as destination section and with bus starting positions at the sections constituting p;
        Extract *non-linear features using Monte-Carlo learning* as described in Fig. 3;
        Build predictive models using n-step TD (for each $n = 1$ to $min(K, j)$) with above features (non-linear) using linear approximation (Eqn. 2);
        Choose the best n-step model (MAPE sense);
        On above similar lines, build best TD(λ) model;
        Choose best among the optimal n-step and TD(λ) models and add to $\mathcal{M}$;

non-linear features, we propose to use TD-1 OR Monte-Carlo learning with non-linear function approximation. We choose Monte-Carlo learning towards this as it is essentially standard ANN-based supervised learning for the prediction problem. We use a 3-hidden layer feed-forward structure towards this during learning. The outputs of this learned structure at the penultimate layer denote the final non-linear features of interest. We essentially freeze the weights learnt in all layers except the last layer. Figure 3 describes the feature construction idea.

Algorithm 1 explains the overall approach for prediction.

IV. Results

Data: The bus route we tested on was from an Indian city and of length 28 kms. We segment this route uniformly into sections of length 400m, which is slightly below the average distance between two consecutive bus stops. This means the total number of sections on the route, $N_{sec} = 70$ in total. We test our algorithms on one data set collected over 12 consecutive weeks. The prediction was carried out on the last week's data (test period) while the remaining data was used for training. The prediction accuracy was evaluated in terms of Mean Absolute Percentage Error (MAPE), which is a scale independent metric. *Percentage error is the Absolute Error divided by the true prediction expressed in percentage*. As the actual travel time across any section is always strictly positive, this metric is always well-defined in this application.

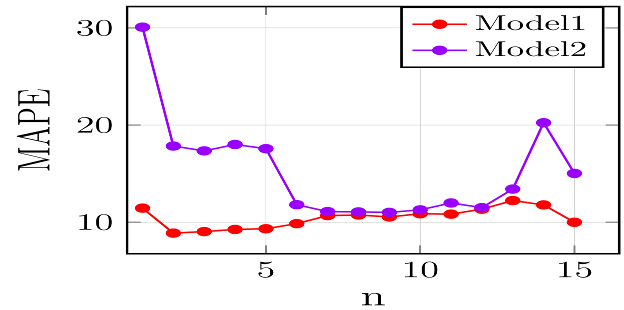
In this paper, we benchmark the proposed TD method against 4 other methods: (1)Historical average (HA) of the training data which serves as a simple baseline, (2)LSTM [10], which captures mainly the spatial correlations in the data, (3)ST [6], a recent spatio-temporal approach for the same problem, (4)ANN [9], which builds a single model to predict between any two sections using a feed-forward structure. This choice of methods enables a computationally diverse comparison.

A. Plain features vs Nonlinear features

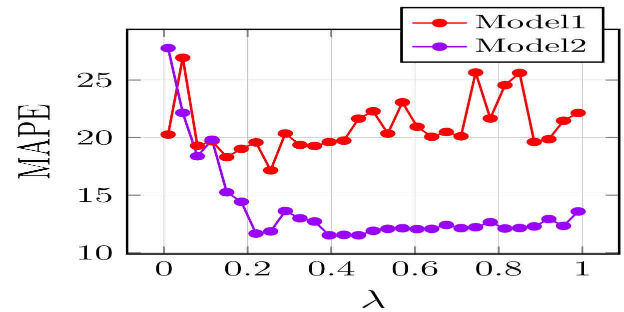
While employing a λ OR n based exploration, for every model built using linear function approximation based TD learning, choice of features can happen in multiple ways. A naive choice would be to use the raw features (described in Sec. III-C) directly. A more sophisticated choice would be to employ the non-linear features as described earlier in Sec. III-D. While depending on the nature of data, there could exist situations where the naive choice of features could lead to superior models, for the purpose of experiments in this paper, we use the non-linear features obtained via our novel feature construction approach to build all models.

B. Improvements due to search over n and λ

As explained earlier, in a bid to improve predictions over Monte-Carlo learning, we search over $n = 1, 2 \ldots$ and $\lambda \in [0, 1]$ for various step sizes α. Fig. 4(a) gives the MAPE obtained for varying n for two random MRP models. The MAPE indicated for each n is obtained by searching for an optimal step size, which could be different for different n.



(a) Variation in n



(b) Variation in λ

Fig. 4. MAPE variation over n and λ for two random MRP models.

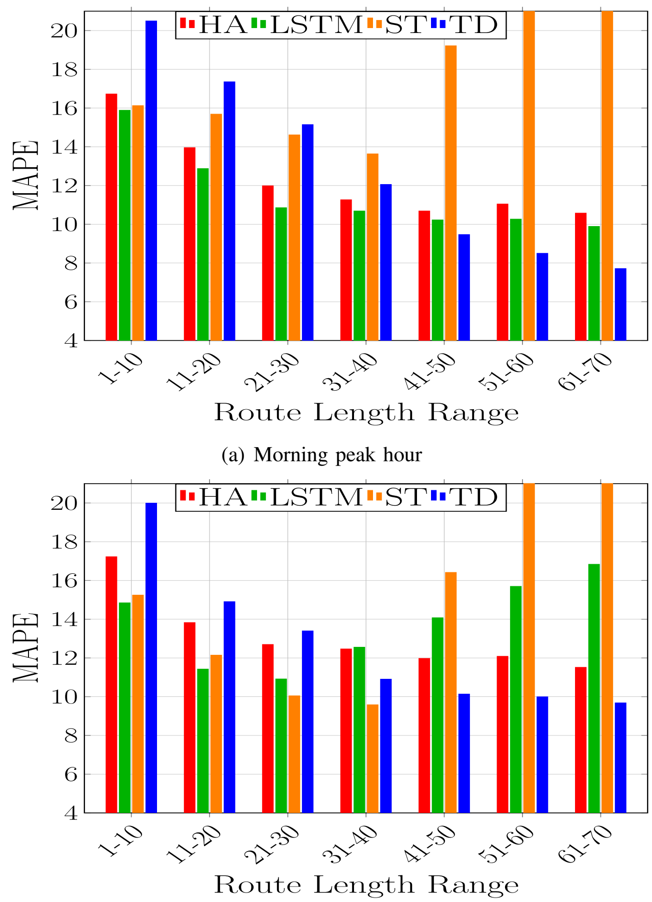
Figure clearly indicates that the best MAPE is achieved at an intermediate n which is greater than 1 and something lesser than $T = 15$ (which corresponds to Monte-Carlo learning), the no. of steps (or sections) in the model.

On similar lines, Fig. 4(b) gives MAPE obtained for a range of λ values varying from 0 (1-step TD learning) to 1 (Monte-Carlo learning). An intermediate λ seems to do better than the extreme cases. As before, the MAPE indicated for each λ is by searching over step-sizes, the best of which could be different for different λ.

Given the elaborate search over n and λ that needs to be carried out for each model, we cut down on this search partly by sticking to the λ search alone in this paper. Hence in all subsequent experiments, the acronym TD refers to TD(λ) only.

C. Average Comparison across Route Lengths

Models for each destination section j: For a given destination section j, depending on its distance from the start of the route, we allow the possibility of building multiple Markov Reward process (MRP) models *to keep the travel-time target variability per MRP model within limits.* A strategy to construct these models would be to bunch together K consecutive sections (chosen to be 15 here) from the start section and build one MRP model (and the associated value function learning) for each bunch of K start positions (as described in Algorithm 1). Each finite horizon MRP model essentially corresponds to a bunch of potential current positions of the bus at a group of consecutive sections. For the extreme case where $j = N_{sec} = 70$, our choice of $K = 15$ leads to 5 models, where the last model covers the last 10 sections. For a given destination section j, an alternate (non-uniform) strategy could be to equally divide the j sections into 2 groups and build a single model based on the first $j/2$ sections. The remaining $j/2$ sections are again split into 2 halves and the process continues. This way the first partition would include sections 1 to $j/2$, the second would constitute $j/2$ to $3j/4$ and so on, leading to $log_2(j)$ models. In this paper, we stick to the first strategy of uniformly bunching the sections together.

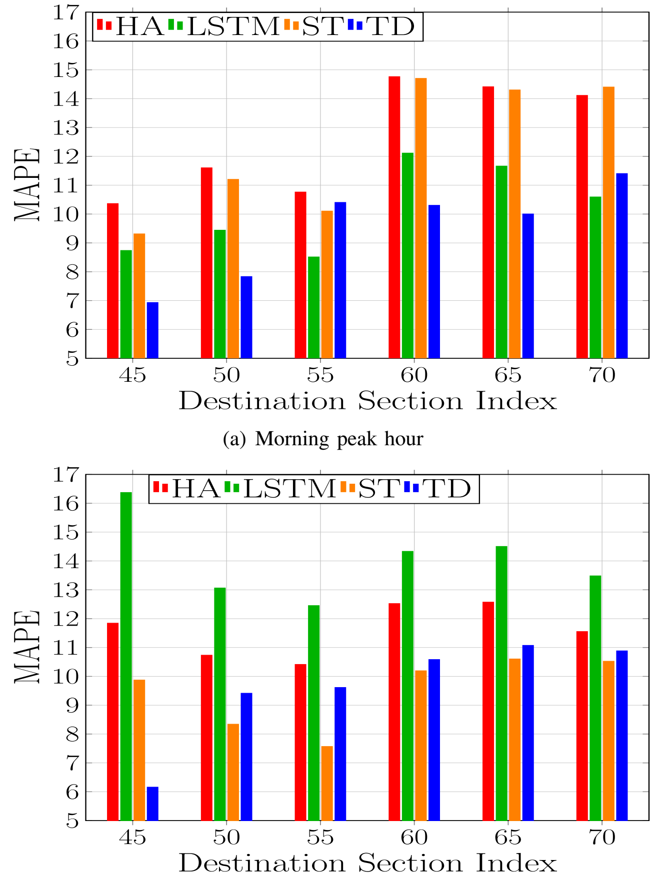


(a) Morning peak hour

(b) Evening peak hour

Fig. 5. MAPE across the entire range of route lengths

We consider testing our proposed approach on routes of diverse lengths. For testing purposes, we build complete models for 6 different destination sections, where the destinations chosen are uniformly placed on the second half of the route. Specifically, we start with $j = 45$ and move till section 70 in steps of 5. We separately look at the MAPE comparisons during morning and evening peak hour traffic. Its during peak hour traffic that BATP needs to be particularly accurate. Fig. 5(a) and 5(a) gives the APE averaged (MAPE) across all morning and evening peak hour test trips respectively, based on the length of the routes (distance of the current position (i) of the bus from the destination j). *We consider 7 ranges (each range spanning* 10 *sections OR* 4 *kms) which cover routes of all possible lengths.* Fig. 5 indicates the MAPE comparisons against the chosen baselines. The figures clearly illustrate that the proposed method makes comparable



(a) Morning peak hour

(b) Evening peak hour

Fig. 6. MAPE across six chosen destinations (averaged across medium and long routes)

OR superior predictions (compared to existing state-of-art) for route lengths greater than 12 kms ($>$ 30 sections). In other words, *it performs better than existing methods on mid and long length sub-routes.* Within this range, it outperforms existing methods with an advantage up to 3%, 7.2%, 30% (some of the long length ST MAPEs have been clipped to 21% for inferrable plots) and 74% over HA, LSTM, ST and ANN approaches respectively. ANN [9] MAPE during both morning and evening peak hours is at least 28% across all 7 route length ranges. Hence we haven't indicated it in Fig. 5.

We further compare the performance in the mid and long length route ranges by now projecting (or averaging) over all routes ending at a particular destination section. Fig 6 shows the comparative MAPEs for morning and evening peak hours separately across the six destination sections. Overall, the TD method is either comparable or outperforming the existing methods. In particular, the approach outperforms existing methods with an advantage of up to 4.5%, 10.2% and 4.4% over HA, LSTM and ST approaches respectively. ANN [9] MAPE in this case, during both morning and evening peak hours is very poor as before and is at least 26% across all 7 route-length ranges. Hence we haven't indicated it in Fig. 6.

D. Best Case improvements

We next explore best case improvements of the proposed method in comparison to each of the baseline methods for different sub-route length ranges. We restrict ourselves to mid and long route-length ranges over which the proposed TD approach has been shown to perform the best in an average MAPE sense. Table I indicates the top-5 best case improvements in MAPE. The top-5 best case improvements are at least 17% across the 4 methods against which we benchmarked our method in detail.

TABLE I
BEST-CASE MAPE IMPROVEMENTS (TOP-5) OF PROPOSED TD SCHEME

	HA	LSTM	ST	ANN
31-40	96,94,94,90,90	34,32,32,31,31	92,89,89,89,87	>100
41-50	82,81,81,80,79	27,27,26,25,24	73,64,59,59,58	>149
51-60	80,79,79,79,78	29,28,28,27,27	61,60,59,57,57	>171
61-70	32,32,31,30,29	21,21,19,17,17	55,54,54,54,51	>187

V. CONCLUSIONS

We proposed a temporal-difference learning based data-driven approach for BATP. The method intelligently reposes the BATP problem as a value function estimation under a suitable Markov reward process (MRP). We strengthened the state definition by including relevant features to the MRP towards enhanced prediction. We used Monte-Carlo learning in this setting to construct non-linear features from the raw (plain) features. These non-linear features were subsequently used for a linear function approximation of the value function. The novel MRP recast and (non-linear) feature construction then enabled us an extensive search over the family of n-step and TD(λ) bag of predictors in an efficient fashion. Recall the gradient computation under linear function approximation is readily feasible in closed form (which makes learning substantially faster) while the weight learning is also relatively stable. The exhaustive n and λ search (we perform) is hence made possible by this novel feature construction and subsequent linear function approximation. Our experimental results indicate the superior performance of our method over a spectrum of existing methods on mid and long length routes. We demonstrated the efficacy of our approach on real field data from complex Indian traffic conditions. As future work, we intend to explore more recent TD approaches like True online TD(λ) [25] for potential performance enhancements.

ACKNOWLEDGEMENT

The authors acknowledge CoEUT (Centre of Excellence in Urban Transport), IIT Madras for the BATP problem introduction and some valuable related discussions.

REFERENCES

[1] P. Ranjitkar, L. S. Tey, E. Chakravorty, and K. L. Hurley, "Bus arrival time modeling based on auckland data," *Transportation Research Record*, vol. 2673, no. 6, pp. 1–9, 2019.

[2] B. D. Bharathi, B. A. Kumar, A. Achar, and L. Vanajakshi, "Bus travel time prediction: a log-normal auto-regressive (ar) modelling approach," *Transportmetrica A: Transport Science*, vol. 16, no. 3, pp. 807–839, 2020.

[3] L. Vanajakshi, S. C. Subramanian, and R. Sivanandan, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses," *IET Intelligent Transporation Systems*, vol. 3(1), pp. 1–9, 2009.

[4] R. Jairam, B. A. Kumar, S. S. Arkatkar, and L. Vanajakshi, "Performance comparison of bus travel time prediction models across indian cities," *Transportation Research Record*, vol. 2672, no. 31, pp. 87–98, 2018.

[5] A. Achar, D. Bharathi, B. A. Kumar, and L. Vanajakshi, "Bus arrival time prediction: A spatial kalman filter approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1298–1307, 2020.

[6] A. Achar, R. Regikumar, and B. A. Kumar, "Dynamic bus arrival time prediction exploiting non-linear correlations," in *2019 International Joint Conference on Neural Networks (IJCNN)*, July 2019, pp. 1–8.

[7] Y. Bin, Y. Zhinzhen, and Y. Baozhen, "Bus arrival time prediction using support vector machines," *Journal of Intelligent Transportation Systems*, vol. 10(4), pp. 151–158, 2007.

[8] K. K. Reddy, B. A. Kumar, and L. Vanajakshi, "Bus travel time prediction under high variability conditions," *Current Science*, vol. 111, no. 4, pp. 700–711, 2016.

[9] W. Fan and Z. Gurmu, "Dynamic travel time prediction models for buses using only gps data," *International Journal of Transportation Science and Technology*, vol. 4, no. 4, pp. 353 – 366, 2015.

[10] Y. Duan, Y. L.V., and F. Wang, "Travel time prediction with lstm neural network," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1053–1058.

[11] C. Paliwal and P. Biyani, "To each route its own eta: A generative modeling framework for eta prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 3076–3081.

[12] B. A. Kumar, S. Mothukuri, L. Vanajakshi, and S. C. Subramanian, "Analytical approach to identify the optimum inputs for a bus travel time prediction method," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2535, pp. 25–34, 2015.

[13] B. A. Kumar, L. Vanajakshi, and S. C. Subramanian, "Bus travel time prediction using a time-space discretization approach," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 308–332, 2017.

[14] B. A. Kumar, V. Kumar, L. Vanajakshi, and S. Subramanian, "Performance comparison of data driven and less data demanding techniques for bus travel time prediction," *European Transport*, vol. 65(9), 2017.

[15] M. Yang, C. C. ., L. Wangz., X. Yanx., and L. Zhou, "Bus arrival time prediction using support vector machine with genetic algorithm," *Neural Network World*, vol. 3, pp. 205–217, 2016.

[16] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, p. 9–44, Aug. 1988.

[17] W. H. Lin and J. Zeng, "Experimental study on real-time bus arrival time prediction with GPS data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1666, pp. 101–109, 1999.

[18] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. New York, NY, USA: ACM, 2012, pp. 379–392.

[19] A. Shalaby and A. Farhan, "Bus travel time prediction for dynamic operations control and passenger information systems," in *82nd Annual Meeting of the Transportation Research Board*. Washington D.C., USA: National Research Council, 2003.

[20] S. V. Kumar and L. Vanajakshi, "Pattern identification based bus arrival time prediction," *Proceedings of the Institution of Civil Engineers-Transport*, vol. 167(3), pp. 194–203, 2012.

[21] R. Jeong and R. Rilett, "Bus arrival time prediction using artificial neural network model," in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, 2004, pp. 988–993.

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming.* Belmont, MA: Athena Scientific, 1996.

[24] S. Singh and P. Dayan, "Analytical mean squared error curves for temporal differencelearning," *Mach. Learn.*, vol. 32, no. 1, p. 5–40, Jul. 1998.

[25] H. Seijen and R. Sutton, "True online td(lambda)," in *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. PMLR, Jun 2014, pp. 692–700.