Neural Network Based Traffic Flow Model for Urban Arterial Travel Time Prediction

Hao Liu
Faculty of Civil Engineering and Geosciences,
Delft University of Technology,
P.O.Box 5048
NL-2600 GA Delft, the Netherlands
Tel: 0031-15-2787690, Fax: 0031-15-2783179
Email: h.liu@tudelft.nl

and
National ITS Center of Engineering and Technology,
No.8 Xi Tu Cheng Road, Haidian Dist, Beijing, 100088, P.R.China
Tel: 0086-10-62079526, Fax: 0086-10-62045674
E-mail: lhao@itsc.com.cn

Hans van Lint
Faculty of Civil Engineering and Geosciences,
Delft University of Technology,
P.O.Box 5048
NL-2600 GA Delft, the Netherlands
Tel: 0031-15-2785061, Fax: 0031-15-2783179
Email: h.vanlint@tudelft.nl

Henk van Zuylen
Faculty of Civil Engineering and Geosciences,
Delft University of Technology,
P.O.Box 5048
NL-2600 GA Delft, the Netherlands
Tel: 0031-15-2782761, Fax: 0031-15-2783179
Email: h.j.vanzuylen@tudelft.nl

Total: (Words:5286 ; Figures:7 ; Tables:1) = 7283
Submitted for presentation at the 86th Annual Meeting of the Transportation Research Board January 21-25, 2007,
Washington D.C. and for publication in Transportation Research Record
Submission Date: 2006 Nov 14
ABSTRACT

Many research efforts on travel time prediction focus predominantly on freeways, while limited work has been done on urban arterials. Among the latter, data driven models, particularly neural networks, have demonstrated promising performance. In most cases, the inputs from spatially separated sources (volumes/speeds collected by loop detectors from different locations) are combined in one single input vector. As a consequence, as the length of the route of interest increases, more and more input variables are inserted into the input vector. This results in a significant increase of parameters of the neural network. A larger parameter space yields more difficulty in training neural networks to not only fit the particular data used for training, but also generalize well to ‘unseen’ data.

This paper provides a neural network based traffic flow model to address the problem of urban arterial travel time prediction. A single segment model based on the Recurrent Neural Network is used for modeling traffic flow on one single signalized link. To model a longer arterial, several separate segment models can be assembled together. This reduces significantly the amount of parameters of the neural network, which make it simpler and easier to be implemented in practice. An urban arterial in the Netherlands has been selected as test bed. Both empirical data and simulation data have been used to evaluate this model. The results indicate that this proposed model is capable of dealing with complex nonlinear urban arterial travel time predictions with satisfying accuracy and effectiveness.
INTRODUCTION

Accurate and timely prediction of travel time is a critical component for many advanced traveler information and advanced traffic management systems. Many research efforts on travel time prediction focus predominantly on freeways, while limited work has been done on urban streets. Since the characteristics and resulting traffic behavior on freeways and urban streets are quite different, approaches that address freeway travel time prediction are not easily translated to urban environments. Artificial neural networks have been shown to be a promising solution to predict traffic variables on urban arterials.

Since delays play an important role in travel time on urban arterials, several promising neural network models have been proposed on queue prediction (delays can be derived from queues), for example (1,2). However, training these neural network models require direct measurements of queue lengths, which are not easily obtained from sensors in practice. This is why these models were tested with simulation data. Thus, these models still need to be adjusted or improved if only conventional loop detector data are available.

This paper presents a neural network based traffic flow model to address the problem of travel time prediction on urban arterials. An urban arterial is subdivided into several segments/links. A general segment neural network model is proposed to predict travel time on a basic segment of urban arterials. Modeling traffic along a signalized urban arterial can be conducted by assembling basic segment models for each segment along the route. To train this scalable neural network, two types of data are needed: volumes (input) and travel times (output). This data requirement is applicable for real-life systems.

Two data sets, respectively data set A and B, have been used to assess this proposed approach. Data set A is synthetic data set obtained from a traffic micro-simulation tool, VISSIM (3). Synthetic data are suitable, from conceptual point of view, for evaluating performance of this proposed model with 100% correct and reliable simulation data. Data set B is empirical data set drawn from a real urban arterial, Kruithuisweg, in the Netherlands. Results obtained from test on empirical data demonstrate the applicability in practice.

The rest of this paper is organized as follows: The next section summarizes related literature regarding travel time prediction on urban arterials. In the section thereafter, a discussion of why we choose data-driven approach instead of model-based approach is carefully elaborated. Next, the design of proposed model based on neural network for a segment of urban arterial is addressed, including a discussion how to extend the basic general model to whole route of urban arterial. The last sections evaluate the performance of the proposed models and offer conclusions and directions for future research.

THE STATE OF THE ART

From literature two main approaches to travel time prediction can be identified: model-based approaches and data-driven approaches. Note that most approaches used for urban environments are data-driven approaches. Here we give a short review of these data-driven models.

A comprehensive literature review has been done to summarize previous research efforts aiming at the development of models for estimating travel time from detector output under various traffic and road conditions (4). The authors found the following limitations of previous research: a) most existing models are link-specific and site-dependent so that these limit the applicability and transferability of the models; b) none of the existing models account for the differences in travel times due to movement type; c) field validation is generally missing, although simulation results showed promising.

Although non-parameterized methods, such as the k-Nearest Neighbor (k-NN) method (5-7) show the capability to estimate travel time. The basic concept of k-NN is to match the
present measurement (e.g. flow, speed, occupancy) with a similar historical pattern. It implicitly acts as an instantaneous model, which assumes traffic condition remaining stationary in the future. Moreover, the choice of temporal window and spatial scope for feature vector require a priori knowledge. For the key parameter k the value of k might be specific for different traffic situations. It is an intensive task to find out appropriate value of k.

Lin proposed a Markov Chain model, splitting the delay experienced by drivers at each intersection along arterial roads into two distinctive states, a state of zero-delay and a state of nominal delay (8). This is coupled with a one-step transition matrix that relates the delay of a through vehicle at an intersection to its delay status at the adjacent upstream intersection. However, the effort needed for a detailed calibration of the transition matrix is extremely time-consuming and tedious. Also, it is quite difficult to use empirical data (e.g. loop detector data) to calibrate this model.

In (9) Liu reviews previous neural network models by categorizing them into two main branches: enhancement of input layers and the use of a new structure of neural networks. All of these models are regarded as data-driven approaches, which are different from model-based approaches, without physical meaning of transportation systems.

To build up neural network models that are able to describe traffic flow, neural network arterial models in (1, 2) have been developed. However, training these neural network models require direct measurements of queue lengths, which are difficult to be measured in real world. Thus, these models still need to be adjusted or improved if no innovative measurement equipment can collect queues in real-life, although they showed promising performance on simulation data.

**MOTIVATION OF RESEARCH APPROACHES**

Insight into the limitations of model-based approaches is our motivation to choose a data-driven approach. The classic Lighthill, Whitham and Richards macroscopic traffic model (10) and analytical delay models are possible model-based approaches dealing with specific urban traffic problems. However, they are not suitable for our purpose. The following explains more details about the reasons.

The classic Lighthill, Whitham and Richards macroscopic traffic model is capable of describing traffic flow analogously to fluid behavior. Built upon the model of Lighthill and Whitham, several models have been proposed as modifications or extensions in an attempt to overcome inherent limitations in this pioneer model (11-13). However, these models aim at dealing with traffic flow on freeways. The main assumption in both 1st and 2nd order macroscopic models is that on segments (typically in the order of 100-500 meters) traffic flows are assumed to be homogeneous. Since urban traffic is interrupted by (controlled or uncontrolled) intersections, deceleration and acceleration behavior may seriously violate those assumptions in both free-flow and congested conditions.

Secondly, several analytical delay models have been developed. These calculate travel time as the sum of link free-flow travel time and the sum of delays encountered at intersections. For instance, the models in 2000 HCM, the 1995 Canadian Capacity Guide, and the 1981 Australian Capacity Guide have been widely accepted to derive delays. However, these models implicitly assume that the mean flow rate is constant for the whole analysis period of, to say, 15 minutes or even longer. This assumption is less applicable for real-time delay prediction with dynamic traffic flow. As Viti and van Zuylen showed the variation in flow rates can give highly significant variations in queue length and delays (14). The queuing models for controlled intersections in literature are not suited for traffic flows that vary between cycles, so the alternatives are to use dynamic models to predict travel times or to develop a heuristic travel time model.
METHODOLOGY

In general, to model traffic flow on road stretches, particularly at network level, one basic element of these stretches is defined as segment/ link. Modeling traffic along a signalized urban arterial can be conducted by assembling each basic segment model together.

Appropriate Structure of Neural networks

Travel times are the results of complex nonlinear and spatiotemporal dynamics of traffic flow. To capture the dynamic processes, two main concerns for neural networks are: input settings and the structure of neural network.

For standard feed-forward neural networks (FNN) the input vector consists of spatially separated inputs at the same time instant, and the output vector corresponds as the same time instant as the input vector. The FNNs do not take into account of temporal relationship between inputs and outputs. To solve the limitation, the so-called time-delayed neural networks (TDNN) are presented to account for input time series with fixed time lags. However, the input selection of temporal dimension (time lag) results in a trade off between model complexity and model generality. In practice, it is quite time-consuming to determine the suitable length of time lags for different spatial inputs.

To avoid involving in the selection of input settings, recurrent neural networks (RNN) offer a structure, which allows the analyst to present inputs of consecutive time instants sequentially, while a feedback (memory) mechanism in the model itself takes into account the temporal dynamics. A context layer is connected with hidden layer to store the internal states. The advantage of this context layer is that the selection of input time window (range) is not required.

However, if all the spatially separated inputs are augmented in a single input vector, this increases the number of parameters and, hence, may also lead to overly complex models. For instance, let assume a route with 3 segments, each segment has 3 input variables, and each segment requires 5 hidden neurons. Thus, the total number of weight parameters of a RNN model with one single input vector is 375(3×3×3×5+3×5×3×5+3×5×1), while that for a RNN model with 3 basic segment input vectors is 135(3×3×5+3×5×5+3×5×1).

Also, the site-specific configuration of neural networks with one single input vector makes transferability to other routes cumbersome. Therefore, this paper will present a generic model, which can describe traffic dynamics within a basic spatial segment, which can be concatenated to predict travel time along a route. The ensuing section presents the basic concept of this paper: modeling traffic dynamics in a basic segment of urban arterials and extending it to an urban street by incorporating with flow prediction.

Basic Segment of Urban Streets

Along urban streets, vehicles decelerate while approaching stop-lines, stop when traffic lights are in red phase, and accelerate when traffic lights turn green. Due to the heterogeneous and non-stationary nature of urban traffic, local speed measurements (e.g. from inductive loops) provide limited information. For example, Pueboobpaphan shows that homogeneity in most cases doesn’t hold for an urban street with a length in the order of 200m, or even shorter (15). This implies that in an urban environment, local speed measurements are not appropriate to represent section traffic states.

The basic segment of urban streets is defined as a link in figure 1(a). Figure 1(b) depicts a typical intersection with four branches, which can be assembled with eight basic segments. Note that no consideration has been taken into the traffic behavior when vehicles travel the inner space of the intersection.
Modeling Traffic on a Single Signalized Segment Using Recurrent Neural Network

As stated above, there are two key issues of modeling traffic using neural networks: a) selection of a suitable type of neural networks; b) determination of input and output pairs.

For the first issue, we propose to use recurrent neural networks (RNN). Elman gives insight in how the RNN manages to represent spatiotemporal patterns in a very efficient distributed manner through its weights (16). The basic idea is to add a context layer as short-term memory, which stores the previous internal states, in order to learn complex spatiotemporal patterns. Such a RNN mathematically resembles a non-linear discrete state-space model, which in essence operates in the same manner as for example macroscopic traffic models mentioned above. Like these a RNN has a common structure in which current states are functions of previous states and the inputs in the previous time period. Van Lint showed that a particular form of RNN, a state space neural network (SSNN), is suitable to capture the complex spatiotemporal traffic dynamics, and applicable for freeway travel time prediction (17).

![Figure 1](image)

(a) Basic segment for an urban arterial
(b) An intersection assembled with basic segments

For input and output data we make the following choices. The output $Y(k)$ is a scalar depicting the segment travel time. For inputs $X(k)$ three variables are used. First of all the available signal control parameters, green time $g(k)$ and cycle length $C(k)$, are used. Secondly, a (rough) proxy for the number of vehicles on the segment is used as input:

$$N(k+1) = \max(N(k) + A(k) - D(k), 0)$$

(1)

Note that the number of vehicles $N(k)$ is derived from the arrival $A(k)$ and departure $D(k)$ volumes. In fact, this would not improve predictive performance compared with using $A(k)$ and $D(k)$ directly. But, since less input variables results in less weight parameters, this strategy reduces the number of weighted links connecting inputs and hidden neuron significantly, which decreases training time.

The resulting neural network based travel time prediction model for a single signalized segment is shown in figure 2.
We will now briefly summarize the mathematical workings of this model. The hidden layer vector $S(k)$ is calculated from the input vector $X(k)$ and previous context vector $S(k-1)$. A weighted sum of inputs, previous states and bias (here the value is fixed at 1) is calculated, and the results are transformed by a transfer function (see equation 2).

\[
S(k) = F(X(k), S(k-1), \psi)
\]

\[
S(k) = \begin{bmatrix}
  s_1(k) \\
  s_2(k) \\
  \vdots \\
  s_m(k)
\end{bmatrix}
\begin{bmatrix}
  h(\sum_{i=1}^{n} w_{i,1}^{ij} x_i(k) + \sum_{j=1}^{m} w_{ij}^{ik} s_j(k-1) + v_{i}^{hj} b_j) \\
  h(\sum_{i=1}^{n} w_{i,2}^{ij} x_i(k) + \sum_{j=1}^{m} w_{ij}^{ik} s_j(k-1) + v_{i}^{hj} b_j) \\
  \vdots \\
  h(\sum_{i=1}^{n} w_{i,m}^{ij} x_i(k) + \sum_{j=1}^{m} w_{ij}^{ik} s_j(k-1) + v_{i}^{hj} b_j)
\end{bmatrix}
\]

\[
X(k) = \begin{bmatrix}
  x_1(k) \\
  x_2(k) \\
  x_3(k)
\end{bmatrix}
\begin{bmatrix}
  g(k) \\
  C(k) \\
  \max(0, N(k) + A(k) - D(k))
\end{bmatrix}
\]

where:
- $s_m$ denotes the value of the $m^{th}$ hidden neuron
- $w_{i,m}^{ij}$ denotes the weight connecting the $i^{th}$ input neuron and the $m^{th}$ hidden neuron
- $w_{e,m}^{ij}$ denotes the weight connecting the $e^{th}$ hidden neuron and the $m^{th}$ context neuron
- $v_{i,m}^{hj}$ denotes the weight of bias associated with the $m^{th}$ hidden neuron
- $b_m$ denotes a bias with fixed value 1 for the $m^{th}$ hidden neuron
- $g$ denotes green time
- $C$ denotes cycle length
- $A$ denotes inflow volume
- $D$ denotes outflow volume
- $h(.)$ is the transfer function
The well-known nonlinear sigmoid transfer function is used. This sigmoid transfer function takes the value from the summation results, and turns them into values between zero and one.

\[ h(z) = \frac{1}{1 + e^{-z}} \]  

(3)

Similarly, the output layer vector \( Y(k) \) is calculated as follows.

\[ Y(k) = h(\sum_{i=1}^{m} w_{io}^i s_i(k) + v_{io}^i b^i) \]

(4)

where \( w_{io}^i \) denotes the weight connecting the \( i^{th} \) hidden neuron and the output neuron; \( v_{io}^i \) denotes the weight of bias associated with the output neuron; \( b^i \) denotes a bias with fixed value 1 for the output neuron.

**Training Neural Networks**

Training neural networks refers to finding the set parameters (weights and bias) which minimize an objective cost function. Two main training paradigms have emerged: batch learning, in which optimization is carried out with respect to the entire training set simultaneously, and online learning, where network parameters are updated after the presentation of each training example (which may be sampled with or without repetition). The two training paradigms can be implemented in the different phases of travel time prediction procedure (see figure 3).

**Figure 3 Recurrent neural network training framework**

**Batch Training**

Normally, batch training is applicable for offline parameter optimization. For batch training, the parameters are only updated after all of the inputs and outputs from historical database are presented. The parameter set which minimize the objective cost function is stored as default optimal parameters, which are used as basis for online application. Many of the conventional approaches can be found in (18). In this paper, we propose to use Levenberg-Marquardt algorithm, which is designed specifically for minimizing a sum of squares error function in the form
\[ E = \frac{1}{2} \sum_{i=1}^{N_p} (\varepsilon_i)^2 \]  \hspace{1cm} (5)

in which \( \varepsilon_i = d_i - y_i \) denotes the model error (desired value minus model prediction), and \( N_p \) is the number of input and output pairs, \( \psi \) is the parameter set.

In short, Levenberg-Marquardt algorithm is to minimize the error function while at the same time trying to keep the step size small so as to ensure that the linear approximation remains valid. The new parameter set is updated according to the rule

\[ \psi_{\text{new}} = \psi_{\text{old}} - (H + \lambda I)^{-1} J^T \varepsilon(\psi_{\text{old}}) \]
\[ J = \frac{\partial Y}{\partial \psi} \]
\[ H = \frac{\partial^2 Y}{\partial \psi^2} = J^T J \]  \hspace{1cm} (6)

\( J \) and \( H \) are the first and second derivative of the output with respect to parameters respectively. For implementation details see for example (19).

**Online Training**

Since the parameter set obtained from offline training represents average conditions over the period represented in the data, it is not sensitive to the variability of prevailing traffic conditions. For instance, the change of weather or surface conditions may result in variations in the parameters. Therefore, the objective of online training is to introduce a systematic procedure that will use the available data to steer the model parameters to values closer to the ones that are most applicable for the present situation. The wealth of information included in the offline values can be incorporated into this process by using them as initial estimates.

The global EKF training algorithm was introduced to online training neural networks (20). A neural network’s behavior can be formulated by the following nonlinear discrete-time system:

\[ \psi(k) = \psi(k-1) + \delta(k-1) \]
\[ Y(k) = G(X(k), \psi(k)) + \xi(k-1) \]  \hspace{1cm} (7)

where \( G(.\) refers to neural network model, \( \psi \) are the parameters of neural network which are assumed to correspond to a stationary process (random walk), \( \xi \) and \( \delta \) are process noise and measurement noise respectively.

The algorithm to solve this nonlinear discrete-time system is listed as follows

1) Initialize the estimate parameter \( \psi \) and the error covariance \( P_0 \) with
\[ \hat{\psi}_0 = E[\psi] \]
\[ P_0 = E[(\psi - \hat{\psi}_0)(\psi - \hat{\psi}_0)^T] \]  \hspace{1cm} (8)

2) Time update for \( k=1, 2, \ldots \)
\[ \hat{\psi}_{k|k-1} = \hat{\psi}_{k-1} \]
\[ P_{k|k-1} = P_{k-1} + E[\delta_{k-1}\delta_{k-1}^T] \]  \hspace{1cm} (9)
3) Measurement update for $k=1,2,...$

\[ \mathbf{e}_k = d_k - G \left( X_k, \mathbf{y}_{k|k-1} \right) \]  

\[ K_k = \frac{P_{k|k-1} J_k^T}{P_{k|k-1} J_k^T P_{k|k-1} + E(\mathbf{e}_k \mathbf{e}_k^T)} \]  

\[ \hat{y}_k = \mathbf{y}_{k|k-1} + K_k \mathbf{e}_k \]  

\[ P_k = P_{k|k-1} - K_k J_k P_{k|k-1} \]

**Modeling Traffic on a Signalized Arterial Using a Neural Network**

We now have the basic ingredients for a (route) travel time prediction model (see figure 4(a)). For offline neural network training each segment performs separately as described above. Inputs are obtained from loop detectors and traffic lights. However, since measured route travel times (RTT) are more common than measured segment travel times (STT) (it is commercially infeasible to install two cameras at each segment), it is necessary to split RTT to several STT.

**Route Travel Time Split for offline Training**

More specifically, an example which only includes four segments is given to demonstrate how to divide RTT into STT (see figure 4(b)). Let assume travel time $t_t(k)$ is RTT at time instant $k$. Now, we need to split it into two parts, one is travel time $t_{t_{i-3}}$ along segment $i-3$, the other is $t_t(k + t_{t_{i-3}}(k))$. It is reasonable to assume that the segment travel time is proportional with respect to degree of congestion $N/gs$ and the length of segment $L$. RTT is subdivided into two STTs as follows

\[ t_t(k) = t_{t_{i-3}}(k) + t_t(k + t_{t_{i-3}}(k)) \]

\[ t_{t_{i-3}}(k) = \frac{(N_{i-3}(k) / g_{i-3} s_{i-3}) \times L_{i-3}}{(N_{i-3}(k) / g_{i-3} s_{i-3}) \times L_{i-3} + (N_{i}(k + t_{t_{i-3}}(k)) / g_{i} s_{i}) \times L_{i}} \times t_t(k) \]  

\[ t_t(k + t_{t_{i-3}}(k)) = \frac{(N_{i}(k + t_{t_{i-3}}(k)) / g_{i} s_{i}) \times L_{i}}{(N_{i}(k + t_{t_{i-3}}(k)) / g_{i} s_{i}) \times L_{i} + (N_{i}(k + t_{t_{i-3}}(k)) / g_{i} s_{i}) \times L_{i}} \times t_t(k) \]

where $g$ is green time, $s$ is saturation flow, and $L$ is the length of segment.

After travel time split, each neural network of individual segment can be trained by available inputs (volume) and outputs (segment travel times).

**Travel Time Prediction**

Predicting route travel time along signalized arterials consists in building up links between upstream and downstream segments. Since inputs of neural network model are volumes, thus travel time prediction requires prediction of volumes. Here we still use figure 4(b) as an example to illustrate how to do this work.

Let assume that all the volumes of each loop detector are available up to time $k$. Then, first predict travel time on segment $i-1$, $i-2$, and $i-3$ as follows

\[ t_{t_{i-1}}(k) = G \left( N_{i-1}(k), A_{i-1}(k), D_{i-1}(k), g_{i-1}, C_{i-1}, \psi_{i-1} \right) \]

\[ t_{t_{i-2}}(k) = G \left( N_{i-2}(k), A_{i-2}(k), D_{i-2}(k), g_{i-2}, C_{i-2}, \psi_{i-2} \right) \]  

\[ t_{t_{i-3}}(k) = G \left( N_{i-3}(k), A_{i-3}(k), D_{i-3}(k), g_{i-3}, C_{i-3}, \psi_{i-3} \right) \]

The outflow volume of segment $i-1$, $i-2$, and $i-3$ are

\[ D_{i-1}(k + t_{t_{i-1}}(k)) = N_{i-1}(k) + A_{i-1}(k) \]

\[ D_{i-2}(k + t_{t_{i-2}}(k)) = N_{i-2}(k) + A_{i-2}(k) \]  

\[ D_{i-3}(k + t_{t_{i-3}}(k)) = N_{i-3}(k) + A_{i-3}(k) \]
To predict travel time along segment $i$, traffic flow prediction of $A_i(k + t_{i-3}(k))$ and $D_i(k + t_{i-3}(k))$ are needed. Inflow of segment $i$ is the sum of outflow of upstream segments as follows:

$$A_i(k + t_{i-3}(k)) = \beta_i^{-1}D_{i-1}(k + t_{i-3}(k)) + \beta_i^{-2}D_{i-2}(k + t_{i-3}(k)) + \beta_i^{-3}D_{i-3}(k + t_{i-3}(k))$$

where $\beta_i^{-1}, \beta_i^{-2}, \beta_i^{-3}$ are right, left and throughput turning ratio.

Note that since travel time on the three segments $i-1$, $i-2$, and $i-3$ might be different, the following conditions should be taken into account.

If $t_{i-3}(k) < both t_{i-1}(k)$ and $t_{i-2}(k)$, then $D_{i-1}(k + t_{i-3}(k))$ and $D_{i-2}(k + t_{i-3}(k))$ can be calculated by interpolation from equation 17.

If $t_{i-2}(k) < t_{i-3}(k) < t_{i-1}(k)$, then $D_{i-1}(k + t_{i-3}(k))$ can be calculated by interpolation from equation 17.

$D_{i-2}$ is only available up to time $k + t_{i-2}(k)$, thus prediction of $D_{i-2}$ from time $k + t_{i-2}(k)$ to $k + t_{i-3}(k)$ is required. Since $D_{i-2}$ can be derived from $A_{i-2}$, prediction of $A_{i-2}$ from time $k + t_{i-2}(k)$ to $k + t_{i-3}(k)$ is required. The methods of prediction boundary variables, $A_{i-2}$, is beyond the scope of this paper. The simple and realistic way is to take the historical average value or multiple a ratio of $A_{i-2}(k + t_{i-2}(k))/A_{i-2(hist)}(k + t_{i-2}(k))$ ($A_{i-2(hist)}(k + t_{i-2}(k))$ is historical average value).
If \( t_{i-1}(k) < t_{i-3}(k) < t_{i-2}(k) \) or both \( t_{i-2}(k) \) and \( t_{i-1}(k) < t_{i-3}(k) \), the calculation is same as above.

**EVALUATION SETUP**

**Models for Comparison**

Three existing models have been used to compare the performance with this proposed model. One is a typical SSNN model, which augments all the differently spatial inputs into the single input vector, trained by offline algorithms. The second is also a typical SSNN model trained by online algorithm. The other is a revised naïve approach, which simply uses arrival travel time \( T_a \) (travel time with arrival time instance) by adding the expected value \( E(\eta(T_a - T_d)) \) of the deviation between arrival travel time \( T_a \) and departure travel time \( T_d \) (travel time with departure time instance). Note that \( E(\eta(T_a - T_d)) \) is negative during congestion set up and positive during congestion dissolve. Predicted travel time \( T_p \) is calculated as the following equation:

\[
T_p(k) = T_a(k) + E(\eta(T_a - T_d))
\]

For convenience,

- **MODEL 1** refers to the proposed model trained by offline algorithm
- **MODEL 2** refers to the proposed model trained by online algorithm
- **MODEL 3** refers to the typical SSNN model trained by offline algorithm
- **MODEL 4** refers to the typical SSNN model trained by online algorithm
- **MODEL 5** refers to the revised naïve model

**Qualitative Criteria for Evaluation**

Two criteria, predictive performance and model complexity, are used.

First, to compare the predictive performance of models three measures are defined as follows.

- **MRE** Mean Relative Error
  \[
  \text{MRE} = \frac{100}{N} \sum_{n=1}^{N} \frac{y_n - d_n}{d_n}
  \]

- **SRE** Standard Deviation of Relative Error
  \[
  \text{SRE} = 100 \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} \left( \frac{y_n - d_n}{d_n} - \frac{\text{MRE}}{100} \right)^2}
  \]

- **MARE** Mean Absolute Relative Error
  \[
  \text{MARE} = \frac{100}{N} \sum_{n=1}^{N} \left| \frac{y_n - d_n}{d_n} \right|
  \]

  where \( N \) denotes the total number of observations, \( y_n \) denotes the \( n^{th} \) predicted value for the \( n^{th} \) input, and \( d_n \) denotes the actual measured value.

Second, the required number of weight parameters with respect of desired accuracy (here refers to MARE) is an indicator for the comparison of model complexity. Also, the training time shows the performance in terms of computational expense on a Pentium IV 2.1 GHz machine. MODEL 5 does not need any parameter due to the use of direct measurements. Online training models are not involved in training time because they update weight at each computation step. Therefore, here only MODEL 1 and 3 are compared.
Simulation Data

The micro-simulation tool VISSIM 3.70 provides full control over all the quantities (inputs and outputs) in highly detailed resolution. Also, it allows simulation of various scenarios of traffic conditions (free-flow, congested, intermediate). The geometrics for the simulation are drawn the same as for the practical test bed (see in figure 5). Since the synthetic data are used to evaluate the performance of this proposed approach, we have not calibrated the simulation model with empirical data.

No sophisticated OD estimators were used, since our primary interest was to collect detailed data (flows from detectors and observed travel times) to train and test our models, rather than have the simulation results fit the real time data. A time-varying traffic demand pattern was input to VISSIM starting with low flows increasing to higher flows and then decreasing back to lower flows to emulate the growth and decay of demand. This resulted in a total of 84 simulation runs (21 traffic demand patterns and each with 4 different random seeds) conducted for data generation and collection. 64 of 84 simulation runs were used to train a neural network, and the rest were used to test the performance of this calibrated model.

Figure 5 The layout of equipments installed along Kruithuisweg

Empirical Data

The empirical data used to evaluate this model are drawn from the RegioLab Delft (21), which is a traffic monitoring system for a region around Delft in The Netherlands. RegioLab Delft collects and integrates detailed traffic data from various traffic data collection systems installed on a wide range of different roads (both urban and freeway) within the region of Delft. The test bed for this study is selected on a provincial road connecting two motorways, A4 and A13. Three license plate cameras with sequence number 256, 257 (left and right) are installed along the 2.05 km Kruithuisweg, in order to measure volumes and travel times. In addition, inductive loops installed at each intersection are used to measure traffic flows within the individual lanes. Figure 5 depicts the layout of measurement equipments installed on this test urban arterial.

Data set B consists of two subset: training set and validation set. Due to corrupted and missing data, only 82 separate days in 2004 have available data. 62 of 82 days were used to construct the training set. 20 days in September and October of 2004 were chosen to provide the validation data. For our purpose, morning peaks (in between 6:00 and 11:00) data are used to evaluated this proposed model.

The loop detectors which are installed along this arterial provide aggregated traffic flows. From the different time stamps of the same set of license plate characters collected at different locations, the individual travel time between these locations can be obtained. This arterial produces travel times of approximate 180 up to 1100 seconds in free flow and seriously congested conditions respectively.
RESULT ANALYSIS AND COMPARISON

Predictive Performance

Table 1 shows the comparison of the results produced by the five models on two validation data sets. It is obvious that all of these models perform better on simulation data than on empirical data. This is because simulation data are 100% accurate, while empirical data always contains corrupted and unreliable data.

Among the four models, the models using offline training algorithm (MODEL 1 and 3) outperform the models using online training algorithm (MODEL 2 and 4). Online training algorithm utilizes the latest measurements (arrival travel time) to adjust weight parameters. In the form of online training algorithm, the algorithm puts no constraints on the model parameters, which may lead to large parameters and hence a sensitive model prone to overfit the data (arrival travel time). This implies a tradeoff between training time and accuracy, which should be considered by model designers.

MODEL 1 and MODEL 2 slightly outperform than MODEL 3 and MODEL 4 respectively. Due to application-specific circumstances we argue that ranking MODEL 1, 2 higher than MODEL 3, 4 is not very meaningful. However, a reasonable conclusion is that all the four models are generally considered accurate with performance (MRE, SRE and MARE) in the range below 20%. An interesting finding is that MODEL 5 is not significantly worse than MODEL 2 and 4 on empirical data. This gives us a hint that simple models might be also feasible for practical implementation when empirical data are unreliable.

<table>
<thead>
<tr>
<th></th>
<th>MODEL 1</th>
<th>MODEL 2</th>
<th>MODEL 3</th>
<th>MODEL 4</th>
<th>MODEL 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Date set A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRE (%)</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>SRE (%)</td>
<td>6</td>
<td>17</td>
<td>7</td>
<td>19</td>
<td>22</td>
</tr>
<tr>
<td>MARE (%)</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td><strong>Date set B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRE (%)</td>
<td>9</td>
<td>14</td>
<td>10</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>SRE (%)</td>
<td>16</td>
<td>21</td>
<td>17</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>MARE (%)</td>
<td>13</td>
<td>18</td>
<td>13</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

Figure 6 Predicted travel times and observations on Sep 30, 2004
More specifically, the results obtained from data of Sep 30, 2004 are depicted in figure 7. It illustrates the performance of these five models on the data of a single day. It is in line with the above discussion. Note that MODEL 3, 4 and 5 underestimates travel times when congestion builds up, and overestimates travel times when congestion dissolves.

Model Complexity

Figure 6 shows that as the desired accuracy reduces (the MARE increases), both the required number of weight parameters and training time for the two models all decrease significantly. Apparently, MODEL 3 needs more weight parameters and training time than MODEL 1, especially in terms of high desired accuracy. This is due to MODEL 3 augments all of spatial different variables into one single input vector, which increases the number of weight parameters.

![Figure 7 Results of comparing model complexity](image)

CONCLUSIONS AND FURTHER RESEARCH

Accuracy, transferability and generalization (fast training speed and low-complexity) are the key characteristics for any travel time prediction model that is to be applied in a real-time environment. The features of this proposed model are summarized as follows:

1. A neural network based traffic flow model is proposed to address the problem of urban arterial travel time prediction. The generic segment model can be assembled together by incorporating flow predictions in order to model traffic propagation along a signalized urban arterial. This generic model is easily to be transferred to any urban street.

2. Compared with previous neural network models, this proposed model provides a generic structure of modeling signalized links, which reduces significantly the required number of weight parameters of neural network without loss of predictive performance. As a result, the training task for neural network model can be conducted easily and quickly.

3. Using offline training algorithms yields more accurate predictions than by online training algorithms.
The approach presented here demonstrates how neural network models can (and should) be used in the design of data driven models, yielding smaller and less-complex models. Given the promising results of this study, further research topics should include:

1. In the case of the evaluation with empirical data, the predictive performance is worse than for simulation data, especially for the case of the online training algorithm. Thus, methodologies for data screening and checking, like online outlier detection should be investigated.
2. Online training algorithms are needed to be improved with some strategies, such as weight constraint, etc.
3. More different types of measurements except volumes, signal timing and travel times are necessary to capture the whole transportation systems. Clearly, the value of the proposed model is limited without consideration of several factors like vehicle composition, bus priority strategies, etc. Since travel times are the results caused by many factors, the further research will extend the proposed model by incorporating more influence factors.

ACKNOWLEDGEMENTS

This research is supported by the advanced traffic monitoring (ATMO) project, which is part of the Dutch national traffic and transport research program TRANSUMO, and partially supported by the Chinese National Intelligent Transport Systems Center of Engineering and Technology. All statements of this paper are under the sole responsibility of the authors and do not necessarily reflect the sponsors’ views.

REFERENCE


