# A Comparative Analysis of Short-Range Travel Time Prediction Methods

Giovanni Huisken[1,2] *

phone: +31-53-489 2543, fax: + 31-53-489 4040, email: g.huisken@ctw.utwente.nl

and

Eric C. van Berkum[1,3]

phone: +31-53-489 4886, fax: + 31-53-489 4040, email: e.c.vanberkum@ctw.utwente.nl

[1] Centre for Transport Studies, Department of Civil Engineering, University of Twente
P.O. Box 217, 7500 AE Enschede, the Netherlands

[2] ARCADIS Spatial Development and the Environment
P.O. Box 264, 6800 AG Arnhem, the Netherlands

[3] Goudappel Coffeng
P.O. Box 161, 7400 AD Deventer, the Netherlands

Paper to be presented at the 82nd TRB Conference.

* Corresponding author

Word count: 4,891 + 10 figures = 7,391

Re-submission date: November 14th, 2002

## ABSTRACT

Increasing car mobility has lead to an increasing demand for traffic information. This contribution deals with information about travel times. When car drivers are provided with this type of information, the travel times should ideally be the times that they *will* encounter. As a result travel times must be predicted, often on a short-term basis. Available data for such a prediction are spot measurements of speed and flow from dual induction loop detectors. In this contribution a prediction method that uses a neural network is described. The performance of the neural network approach is compared with two naïve methods that are currently in operation, using data from a short-range motorway site: the A13 motorway from The Hague to Rotterdam.

In order to be able to assess the performance of these methods it is imperative to use data on travel times. Since this data is not readily available, an estimation algorithm was selected where travel time is determined using speed and flow data from loop detectors. Five algorithms to estimate travel times were assessed using a data set with actually measured travel times through license plate recognition.

Results of the assessment of short-range travel time predictions show that the Artificial Neural Network (ANN) method significantly outperforms the Dynamic Travel Time Estimation (DTTE) method, which in turn outperforms the Static Travel Time Estimation (STTE) method.

## INTRODUCTION

The growth in car mobility has lead to more uncertainty in travel times. As a result car drivers have an increasing demand for information on these travel times (*1*). In this contribution we will concentrate on a VMS sign that provides car drivers with travel time information. In order to operate such a system, travel times must ideally be predicted. This prediction is most likely based on point measurements of speed and flow.

Travel times could be measured using automated vehicle identification (AVI) techniques, e.g. floating car data or automated license plate recognition. Yet these techniques are rarely used since they require large investments in roadside equipment. Currently only data from dual inductive loop detectors are available. Since loop detectors yield spot measurements of flow and speed, travel times can only be estimated, e.g. by means of time series (*2*,*3*), traffic flow theory using data of speed traps (*4*), using cross correlation (*5*), or using data from single loops (*6*), and not actually be measured.

When car drivers are provided with information on travel times, these travel times should ideally be the times that they *will* encounter. Therefore, there is a need to predict travel times, e.g. based on historic databases in combination with regression techniques and GIS (*7*) or scenario based statistical modelling (*8*).

This contribution is about developing a method to accurately predict travel times within the short-term range. These predictions are based on spot measurements of flow and speed. We will assess 3 prediction methods, i.e. 2 rather naïve methods that are currently in operation (the Static Travel Time Estimation – STTE – method and the Dynamic Travel Time Estimation – DTTE – method, see (*9*,*10*)) and a new Artificial Neural Network – ANN – method. A prediction method in general has two main characteristics: the prediction is based on history, i.e. how did travel time develop under several different circumstances, and on functional relationships, i.e. what are variables that can explain the development of travel times. Both characteristics are hard to quantify. This means that it is hard to define exactly which variables can explain the development of travel times and also the nature of these dependencies is not trivial, so the learning mechanism is not fully defined. Therefore we did use a methodology, ANN, that can cope with these difficulties. Advantages are that the input for prediction must not be fully predefined, and also, learning from the past does not require to predefine the necessary functional relationships. However this also has a prize, since a ANN is more or less a black box that does not allow for inspection of the functional relationships that were deduced.

In order to be able to assess the performances of these methods, actual travel times are needed. And since these cannot in general be measured a method to estimate actual travel times from loop detector data must be used. In this respect 5 alternative algorithms were compared, and the best alternative was selected. So, the travel time estimation algorithm was used to build a database of travel times that were used to assess the 3 prediction methods.

## DATA COLLECTION

This section is about the data collection site and the data itself. An overview of the geographical site is given along with quantitative descriptions of the data sets.

### The geographical site

The chosen geographical location (figure 1, left side) is the motorway A13 from The Hague to Rotterdam – one direction only. The site has a length of 11.4 km, exists of 3 lanes (some sections have an added weaving lane), has 5 on- and off-ramps, a speed limit of 100 km/h, and is equipped with one gas station (approximately halfway). Further there are 21 dual induction loop detectors (see figure 1, right side - schematic).

**FIGURE 1 Left-hand side: The motorway section from The Hague (up-left) to Rotterdam (down-right). Right-hand side: Schematic overview of the site.**

### Data set description

As mentioned before this research consists of two parts. In the first part an algorithm is selected that yields the best travel time estimate using inductive loop data. The second deals with the assessment of three methods to predict travel times.

For the first part of the research two data sets have been collected: one license plate recognition set (Delft-Noord/Nootdorp [km 7.3 post] – Rotterdam-Overschie [km 17.55 post]) and one through the dual induction loop detectors with the MARI [*More Applicatie Routekeuze Informatie*] system. The license plate recognition set was collected by time and license plate registration of passing red vehicles at the starting and re-identification point and subsequent subtraction produced travel times. The collection took place on October 8th and 9th 1996: 07:00 – 09:30 and 15:30 – 18:30 and on October 11th 1996: 15:00 – 18:45. During this period, also inductive loop data for the 21 locations were collected, containing flow and speed on a one-minute basis.

For the second part of the research, i.e. the comparison of the three prediction methods, loop data was collected from November 11th 1996 – February 23rd 1997. The set also contains 1-minute aggregated data from loop detectors containing information on flow and speed.

## METHODOLOGY

This section describes the algorithms that were used to estimate travel times and the methods that were used to predict travel times. First an analysis of several algorithms will be given to determine the one that produces the most accurate travel times. Then the static travel time estimation method will be discussed followed by an overview of the dynamic travel time estimation method and some information regarding the artificial neural network (ANN) method used in this research.

### Description of travel time estimation algorithms

The performance of five algorithms that estimate travel times using data from induction loops is assessed, see also (*9*, *10*). Definitions are graphically supported by figure 2. Units used are time [*seconds*], distance [*meters*], and intensity (or flow) [*vehicles/hour*].

**FIGURE 2 Time distance diagram of link *c* that is located between loops *a* and *b*.**

*Link method - Speed*

Suppose link *c* is located between dual induction loops *a* and *b* (with *a* upstream of *b*). Speeds reported during period *p* by loop *a*, i.e. $v_{c,a}(p)$ is then assigned to the first half of the link and the speeds reported during period *p* by loop *b*, i.e. $v_{c,b}(p)$ to the second half. Travel time of link *c* is estimated as:

$$TT_c^{speed}(p)=\frac{L_c}{2\cdot v_{c,a}(p)}+\frac{L_c}{2\cdot v_{c,b}(p)} \qquad (1)$$

*Link method - Mass balance*

The mass balance method compares the volume reported by *a* during period (*p*-1), i.e. $I_{c,a}(p-1)$ with volume reported by *b* during period *p*, i.e. $I_{c,b}(p)$ The duration of period *p* should approximate the free flow time, i.e. $t_c^{ff}$. Travel time of link *c* is now estimated as:

$$TT_c^{mass}(p)=t_c^{ff}+\frac{N_c(p)}{C_{eff,c}(p)} \qquad (2a)$$

with

$$N_c(p)=N_c(p-1)+\left(I_{c,a}\left(p-t_c^{ff}\right)-I_{c,b}(p)\right)\cdot\frac{p}{3600} \qquad (2b)$$

and

$$C_{eff,c}(p)=\frac{1}{3}\left(I_{c,b}(p)+I_{c,b}(p-1)+I_{c,b}(p-2)\right)\cdot\frac{p}{3600} \qquad (2c)$$

where $N_c(p)$ is the number of vehicles on link $c$ during period $p$, and $C_{eff,c}(p)$ is the effective outflow out of link $c$ during period $p$.

*Route method - Static*

The static route method uses last-known link travel times and sums them. Links (L) are defined as the distance between two consecutive dual induction loops. Suppose a vehicle enters a specific route (number of links = $m$, number of loops = $m$+1) at time $t = T_0$. The most recent recorded loop speeds $v_c(T_0)$, ($c$ = 1..$m$+1), are assigned to half of the link upstream and downstream of the specific loop location. The Static Travel Time Estimation (STTE) then yields:

$$STTE_{T_0} = \frac{L_1}{2 \cdot v_1(T_0)} + \sum_{c=2}^{m} \left( \frac{L_{c-1} + L_c}{2 \cdot v_c(T_0)} \right) + \frac{L_m}{2 \cdot v_{m+1}(T_0)} \qquad (3)$$

So, link travel times are assumed fixed as of the moment that the vehicle enters the route (on t = $T_0$).

*Route method - Dynamic*

The Dynamic Travel Time Estimation (DTTE) at time $T_0$ can only be estimated by historical reconstruction and is done iteratively. Suppose vehicle 1 enters the route and vehicle 2 leaves the route at $T_0$. The DTTE of vehicle 2 can now be obtained by looking at the speed reported at time $T_0$ by the last loop $v_{m+1}(T_0)$. Tracking vehicle 2 back to halfway between loops $m+1$ and $m$ takes $\tau_1$ = ½·$L_m$ / $v_{m+1}(T_0)$. The time spent on the section halfway between loops $m+1$ and $m$ to loop $m$ is now estimated by $t_2$ = ½·$L_m$ / $v_m(T_0 - \tau_1)$. This iterative process goes on until the starting point of the route is reached with consequently DTTE = $\tau_1 + \tau_2 + \tau_3 + \ldots + \tau_{2m}$. This now means that vehicle 1 is presented with a travel time of $\tau_1 + \tau_2 + \tau_3 + \ldots + \tau_{2m}$ on time $T_0$ that actually vehicle 2 should have gotten when it entered the route on $T_0 - (\tau_1 + \tau_2 + \tau_3 + \ldots + \tau_{2m})$, so there is a prediction time lag of $\tau_1 + \tau_2 + \tau_3 + \ldots + \tau_{2m}$.

*Smoothing method - Input*

Smoothing of the input (speed and volume) is done when the values reported by loop $c$ lie outside the domain of [80% - 120%] of the values of loops ($c$-1) and ($c$+1).

*Smoothing method - Output*

Smoothing of the output (link travel time) is done when the values estimated for link $c$ at period $p$ lie outside the domain of [80% - 120%] of the values estimated for loop $c$ at period $p$-1. The performance of five algorithms that estimate travel times using data from induction loops is assessed. An overview of these algorithms using is given in table 1.

**TABLE 1 Travel Time Estimation Algorithms**

Algorithms RT0, RT1, and RT2 use the speed method as well as the mass balance method to determine the travel time on links and the respective weights of both methods when used simultaneous are determined by speed and are given in table 2.

**TABLE 2 Weights of the Speed and the Mass Balance Method**

The travel time estimates of the five different algorithms were compared with the actually measured travel times RT_M_AVG (figure 3). It appeared that the outcome of the algorithms had to be shifted in time. The optimal shift, however, was different for each algorithm. Best fit was achieved using RT4 after shifting it by 6 minutes. This algorithm yielded an average error of 23.0 seconds on a mean travel time of 8 minutes and 46 seconds.

**FIGURE 3 Estimated travel times by algorithm.**

### Description of the three prediction methods

Now that we have selected the best algorithm to estimate travel times from loop data, i.e. RT4, three methods to actually predict travel times are assessed. The first two methods, the static and dynamic estimation (STTE and DTTE) that were described in the previous sub-sections, are currently in use and are rather naïve by nature. The third method, that uses an artificial neural network, was developed specifically for short-range travel time prediction.

*Artificial Neural Networks*

Artificial neural networks (ANN) are based upon biological neural networks **-** like the human brain **-** by mimicking their architectural structure and information processing in a simplified manner. They both consist of building blocks or processing elements called neurons that are highly interconnected making the networks parallel information processing systems. Although the artificial neural networks are a rudimentary imitation of biological ones, they are to some extend capable of tasks such as pattern recognition, perception and motor control which are considered poorly performed and highly processor time inefficient by conventional linear processing whereas they seem to be done with ease by e.g. the human brain. These parallel systems are also known to be robust and to have the capability to capture highly non-linear mappings between input and output. We now give a short overview of a widely used basic artificial neural network called a Multi Layer Feedforward (MLF) neural network.

MLF neural networks generally exist of one layer of input neurons, one or more layers of hidden neurons and a layer of output neurons whereas the subsequent layers are fully connected (figure 4a). The neurons are fundamental to the operation of any artificial neural network and like biological neurons they can be identified by three basic elements (figure 4b):

- *input*: a set of signals, each of which is characterised by its synaptic weight or strength; they can be either positive (excitatory) or negative (inhibitory).
- *adding*: all incoming signals are added at the summing junction; this is a linear combiner.
- *output*: obtained through squashing the added input signals after subtraction of the threshold in the activation function.

**FIGURE 4 a) MLF topography. b) Nonlinear model of artificial neuron *k*.**

In mathematical terms, a neuron *k* can be described by the following equation:

$$y_k = \varphi(u_k - \theta_k) \quad with \quad u_k = \sum_{j=1}^{q} w_{kj} \cdot x_j \qquad (4)$$

where $x_1$, $x_2$, …, $x_p$ are the input signals, $w_{k1}$, $w_{k2}$, …, $w_{kq}$ are the synaptic weights of neuron *k*, $u_k$ is the linear combiner output, $\theta_k$ is the threshold that can be looked upon as an external parameter, $\varphi(\cdot)$ is the activation function (a sigmoid function such as the hyperbolic tangent function), and $y_k$ is the output signal of the neuron.

The MLF neural network is a supervised learning network meaning that during the training phase all inputs are mapped on desired outputs. The error, i.e. the difference between the actual and the desired output, is a criterion that is used to adjust the weights of the neurons iteratively so that the total error of all input-output pairs is minimised. The algorithm responsible is called a learning rule and the most commonly used one is the back-propagation algorithm. The instantaneous sum of the network error signal generated at iteration *n* is defined as the sum of all squared output layer neuron errors:

$$E(n) = \frac{1}{2}\sum_{k \in C} e_k(n)^2 \quad with \quad e_k(n) = d_k(n) - y_k(n) \qquad (5)$$

where *e* defines the error signal of neuron *k* with *d* being the target signal and *y* defined by formula (4). The gradient to minimise *E* with respect to the free parameters of the network (the weights) is given by:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial e_k} \cdot \frac{\partial e_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_{kj}} = e_k \cdot -1 \cdot \varphi'(u_k) \cdot x_j \qquad (6)$$

The delta rule is defined by this gradient multiplied by the rate of learning $\eta$:

$$\Delta w_{kj}(n) = \eta \cdot \delta_k(n) \cdot x_j(n) \quad with \quad \delta_k = -e_k \cdot \varphi'(u_k) \tag{7}$$

The back-propagation algorithm uses the delta rule to adjust the weights in the network, however, the above described change in weights holds only for neurons belonging to the output layer. Weights belonging to hidden layers are adjusted backwards according to:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot x_j(n) \quad with \quad \delta_j = \varphi'(u_k) \cdot \sum_k \delta_k \cdot w_{kj} \tag{8}$$

More comprehensive information on neural networks, both MLF and other ANNs can be found in e.g. (*11*, *12*), whereas (*13*, *14*, *15*, *16*) show ANN-applications to transportation.

## MODEL DEVELOPMENT

The previous sections gave brief descriptions of the prediction methods. In this section the actual implementation of the methods is discussed.

STTE and DTTE modelling was done straightforward. The STTE model was set up with the link lengths parameters. After this the model was ready to process input data (speed values collected by the dual induction loops) into estimated travel times. The same holds for the DTTE model: it also was set up with the link length parameters and transforms the speed values iteratively into travel times.

ANN modelling was done by using algorithm RT4 and shifting it 6 minutes backwards in time to estimate the most accurate travel time ($tt_{\text{acc}}$) of a vehicle *leaving* the site. Subsequently the thus established $tt_{\text{acc}}$ on time (t = $T_1$) was linked as target value to the values produced by the induction loops at the time the vehicle *entered* the site (t = $T_1$ - $tt_{\text{acc}}$). In this manner the values produced by the induction loops are linked to the travel times that will be encountered.

For the ANN approach the number of input variables is: 21 (induction loops) * 2 (quantities: speed and intensity) + 1 (time of day) = 43. Because of the high number of inputs and slow convergence this approach resulted in time consuming training phases. Pre-processing was used to cut down the number of input neurons.

### Pre-processing

The choice which variables the input layer must contain is not trivial. A trade off between completeness and size must be made. An input layer with data from all measurements, i.e. all detectors and all time intervals, may yield the best predictions. However, this layer will be so large that the network is almost impossible to operate. Therefore the data in this complete layer will be preprocessed, such that it only contains data that contribute significantly to the prediction of travel times.

In this respect it is hard to use any sound statistical technique as factor analysis since we are not specifically interested in the order in which a preprocessed variable explains the travel time, but more how this variable contributes to a better prediction of travel times. Since this is hard to determine numerically, Human Intelligence was used. First a selection of variables that are likely to influence travel times was made. These variables can be set out in 2-dimensional scatter plots and by projecting the accompanying travel times (colour coded) one can identify clusters of travel times in case of high variable influence or a blur in case of low variable influence. The variables that have the largest explaining value on travel time are:

- time of day
- mean speed
- maximum speed
- number of vehicles
- most recent travel time
- mean speed in bottlenecks
- mean speed in between congested areas (if present)
- maximum speed in between congested areas (if present)
- length of congested area

- variation of length of congested area
- speed reported by loop 19 (upstream of bottleneck)
- speed reported by loop 20 (bottleneck)

The variables were normalised and accompanied by the travel times that acted as targets. The number of input neurons now reduced to 12 (the afore-mentioned ones) instead of 43 (speed + flow of 21 loops + time of day) or even more if historical data of previous periods is also included (every 1-minute period *p* produces speed + flow data per loop resulting in (*p* periods)*(21 loops)*(2 variables: speed and flow)+(1 variable – time of day) input variables. Since the number of input neurons dramatically influences the processor time in the learning phase this reduction of the number of input neurons not only speeded up the training phase, it also proved to produce better results coming from smaller neural networks.

### Establishing the optimum ANN architecture

The data set was divided into 3 equally sized subsets (for cross-validation purposes) where one subset was used to find the optimum number of hidden neurons and epochs (training cycles). Figure 5 shows the MLF generated Mean Square Error (MSE) for the variable neurons in the hidden layer indicating that the optimum MLF architecture is a 12-5-1 neurons architecture and the optimum training epochs is 150 training epochs (after which over-fitting occurred).

**FIGURE 5 a) MSE of test set vs. hidden nodes. b) MSE of test set vs. epochs.**

## RESULTS

The predictions of the three methods were compared with the travel times determined by RT4. Travel time prediction becomes interesting when free flow conditions no longer hold. Therefore prediction was only executed when travel times exceeded 500 seconds (i.e. average speed under 82 km/h). To compare the method's performances several error measurements were determined (see formula 9a-e), i.e. Mean Relative Error (MRE) [*%*], the Mean Absolute Relative Error (MARE) [*%*], the Mean Time Error (MTE) [*seconds*], the Mean Absolute Time Error (MATE) [*seconds*], and correlation coefficient $r^2$.

$$MRE = \sum_s \frac{\hat{t}_s - t_s}{s \cdot \hat{t}_s} \qquad (9a) \qquad\qquad MARE = \sum_s \left| \frac{\hat{t}_s - t_s}{s \cdot \hat{t}_s} \right| \qquad (9b)$$

$$MTE = \sum_s \frac{\hat{t}_s - t_s}{s} \qquad (9c) \qquad\qquad MATE = \sum_s \left| \frac{\hat{t}_s - t_s}{s} \right| \qquad (9d)$$

$$r^2 = \frac{\left( s\left( \sum_s t_s \cdot \hat{t}_s \right) - \left( \sum_s t_s \right)\left( \sum_s \hat{t}_s \right) \right)^2}{\left( s\sum_s {t_s}^2 - \left( \sum_s t_s \right)^2 \right) \cdot \left( s\sum_s {\hat{t}_s}^2 - \left( \sum_s \hat{t}_s \right)^2 \right)} \qquad (9e)$$

where *s* is the number of cases (7,075 = the total number of cases, being every minute of the second data set - as mentioned in the data collection section - where the travel time exceeded 500 seconds), $\hat{t}_s$ is the travel time to be predicted (target value), and $t_s$ is the travel time generated by the model.

The results are given in table 3 that shows that ANN significantly outperforms DTTE, which in turn significantly outperforms STTE. The MRE values are also classified into 5%-error intervals and from this can be concludes that roughly two-thirds, a half, and one-third of the prediction cases fall in the [-5%, 5%] error domain (between the 2 vertical lines) for ANN, DTTE, and STTE, respectively (figure 6).

**TABLE 3 MRE, MARE, MTE, MATE and $r^2$ Results**

**FIGURE 6 Categorised relative error percentage per method.**

Figure 7, finally, shows 2 alternatively obtained travel times during 900 minutes of peak hour time. The first displayed travel time graph (- - -) was obtained by back-tracing vehicles and shifting the travel time, i.e. by dynamically – a posteriori – estimation. The second displayed graph (– - –) was obtained by feeding online data coming from induction loops into the MLF neural network that thus gave the MLF predicted travel times.

**FIGURE 7 Dynamic travel time (a posteriori estimated) and MLF travel time (predicted).**

## CONCLUSIONS

Five algorithms were used to estimate travel times using data from induction loops. The results of these five algorithms were compared with actually measured travel times, using license plate recognition. It appeared that RT4, a dynamic speed based algorithm, with a 6-minute shift, produced the best results.

The ultimate goal of this research is to select a method that can provide travellers with accurate travel time they will encounter. Therefore a short-term prediction must be performed. Three different methods were compared. Next to two naïve methods STTE and DTTE (Static and Dynamic Travel Time Estimation) that are currently used, a more sophisticated method that uses an artificial neural network was developed.

During the process it turned out that the pre-processing phase is not to be neglected due to its possible high impact on the duration of the training phase and the performance of an ANN. This research showed that selection of variables not only speeded up the training phase because of the significant decrease of input variables resulting in a much smaller ANN architecture but also that the performance of the neural network increased.

For a period of 13 days loop data was collected, and algorithm RT4 was executed in order to obtain a dataset of travel times. A subset of this set, where average speed was smaller than 82 km/h, was used for the assessment of the three methods. The results show that the ANN approach is able to produce very good predictions of short-range travel times ($r^2 = 0,96$) and clearly outperforms the more naïve approaches STTE and DTTE. However, we observed that the differences of performance occur mainly due to better predictions of the ANN when conditions are changing, i.e. during building up of congestion and regeneration of traffic flow.

For practical reasons further research amongst drivers should reveal to what extend the extra effort of training and testing the new ANN method for each trajectory to come to more accurate travel time predictions is worthwhile. This especially applies to relative short stretches such as the site used in this research keeping in mind that the free flow travel time is just beneath 7 minutes and the mean deviation of the naïve methods is 80 seconds and 55 seconds for the STTE method and DTTE method, respectively.

Another direction for further research might be to create a database out of a large – costly – license plate recognition data set and a loop detector data set that should be collected simultaneous. This database then has target travel time values that can directly connected onto loop values. This procedure would eliminate the reconstruction phase of choosing and implementing a 'best algorithm' described in this research and therefore an ANN trained with this database would probably predict even more accurate travel times.

## REFERENCES

1. Huisken, G., and E.C. Van Berkum. DAB in the Netherlands? In: *Proceedings of the 8th Meeting of the Euro Working Group Transportation EWGT*, September 2000, Rome, Italy.
2. Ahmed M.S., and A.R. Cook. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. *Transportation Research Record 722*, TRB, National Research Council, Washington, D.C., 1977, pp. 1-9.
3. Van Arem B., M.J.M. Van Der Vlist, M. Muste, and S.A. Smulders. Travel time estimation in the GERDIEN project. *International Journal of Forecasting*, Vol. 13, No. 1, 1997, pp. 73-85.
4. Petty K.F., P. Bickel, M. Ostland, J. Rice, F. Schoenberg, J. Jiang, and Y. Ritov. Accurate estimation of travel times from single-loop detectors. *Transportation Research*, Vol. 32A, No. 1, 1998, pp. 1-17.
5. Coifman, B. Estimating travel times and vehicle trajectories on freeways using dual loop detectors, *Transportation Research*, Vol. 36A, No. 4, 2002, pp. 351-364.
6. Dailey, D.J. 1992. Travel-time estimation using cross-correlation techniques. *Transportation Research*, Vol. 27B, No. 2, 1992, pp. 97-107.
7. You J., and T.J. Kim. Development and evaluation of a hybrid travel time forecasting model. *Transportation Research*, Vol. 8C, No. 1-6, 2000, pp. 231-256.
8. Hounsell N.B., and S. Ishtiaq. Journey time forecasting for dynamic route guidance systems in incident conditions. *International Journal of Forecasting*, Vol. 13, No. 1, 1997, pp. 33-42.
9. Zee J.C. *Oude reistijden actueel* (in Dutch). M.Sc. report, University of Twente, Enschede, The Netherlands, 2001.
10. Bovy P.H.L., and R. Thijs. *Estimators of travel time for road networks, new developments, evaluation results, and applications*. Delft University Press, Delft, The Netherlands, 2000.
11. Bishop C.M. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1995.
12. Haykin S. *Neural Networks: a comprehensive foundation*. Prentice Hall, Upper Saddle River, NJ, USA, 1994.
13. Dougherty M.S. A review of neural networks applied to transport. *Transportation Research*, Vol. 3C, No. 4, 1995, pp. 247-260.
14. Huisken G. *Literature review: neural network applications in traffic and transport*. Research report: 98W – 012 / V&V-003, Enschede, University of Twente, The Netherlands, 1998.
15. Huisken G. Short-term forecasting of traffic flow on freeways. In: *Proceedings of the 9th World Conference on Transport Research*, July 2001, Seoul, Korea.
16. Huisken G., and M.F.A.M. van Maarseveen. Congestion prediction on motorways: a comparative analysis. In: *Proceedings of the 7th World Congress on ITS* (CD-ROM), November 2000, Turin, Italy.

**LIST OF FIGURES AND TABLES**

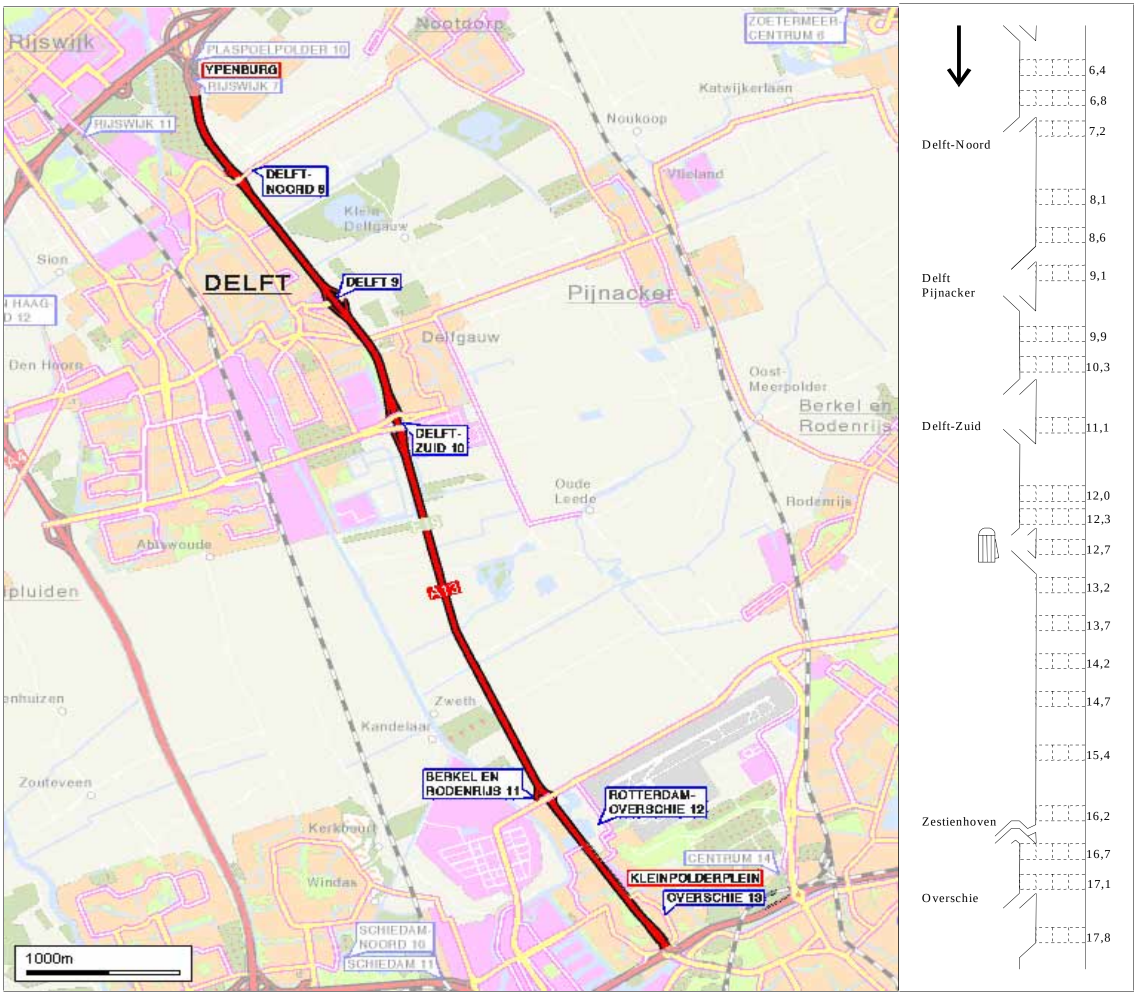

**FIGURE 1 Left-hand side: The motorway section from The Hague (up-left) to Rotterdam (down-right). Right-hand side: Schematic overview of the site.**

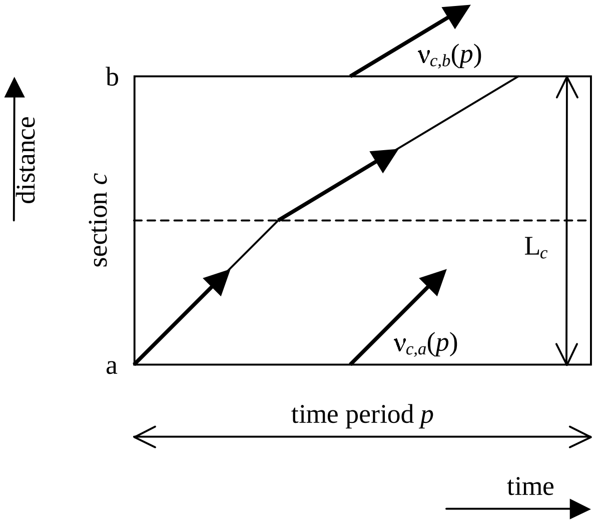

**FIGURE 2 Time distance diagram of link *c* that is located between loops *a* and *b*.**

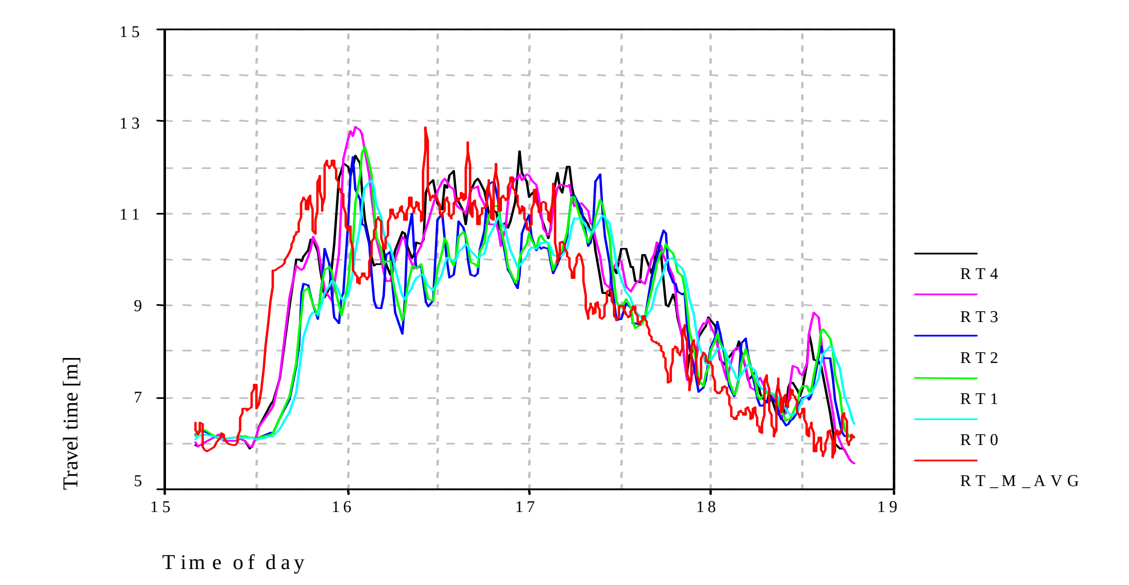

**FIGURE 3 Estimated travel times by algorithm.**

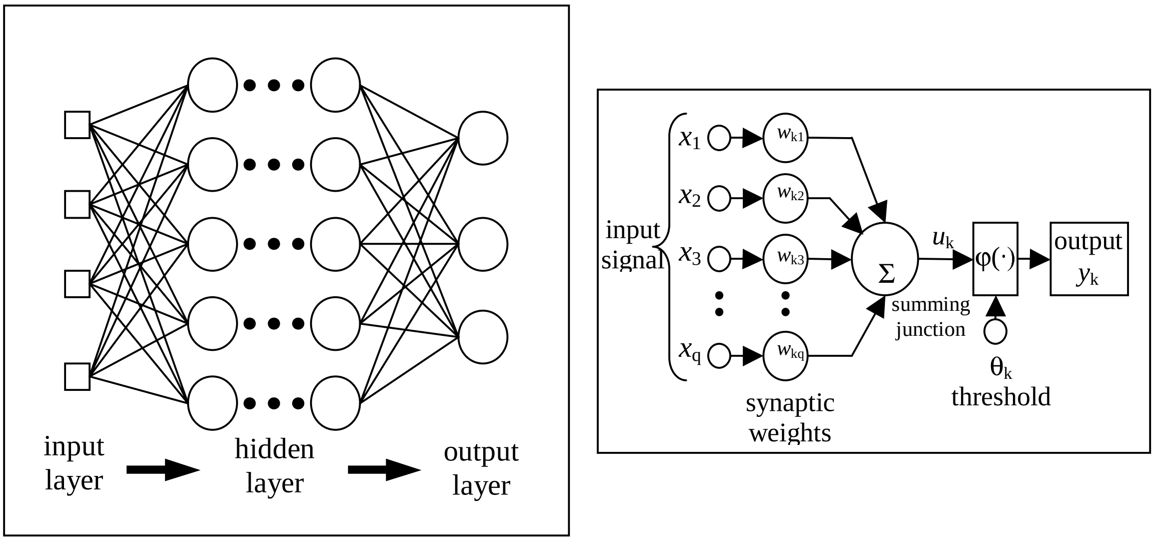

**FIGURE 4 a) MLF topography.** **b) Nonlinear model of artificial neuron *k*.**

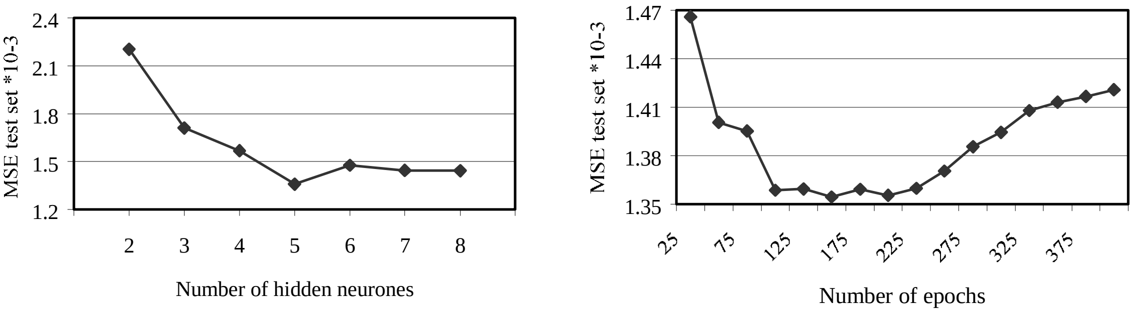

**FIGURE 5 a) MSE of test set vs. hidden nodes. b) MSE of test set vs. epochs.**

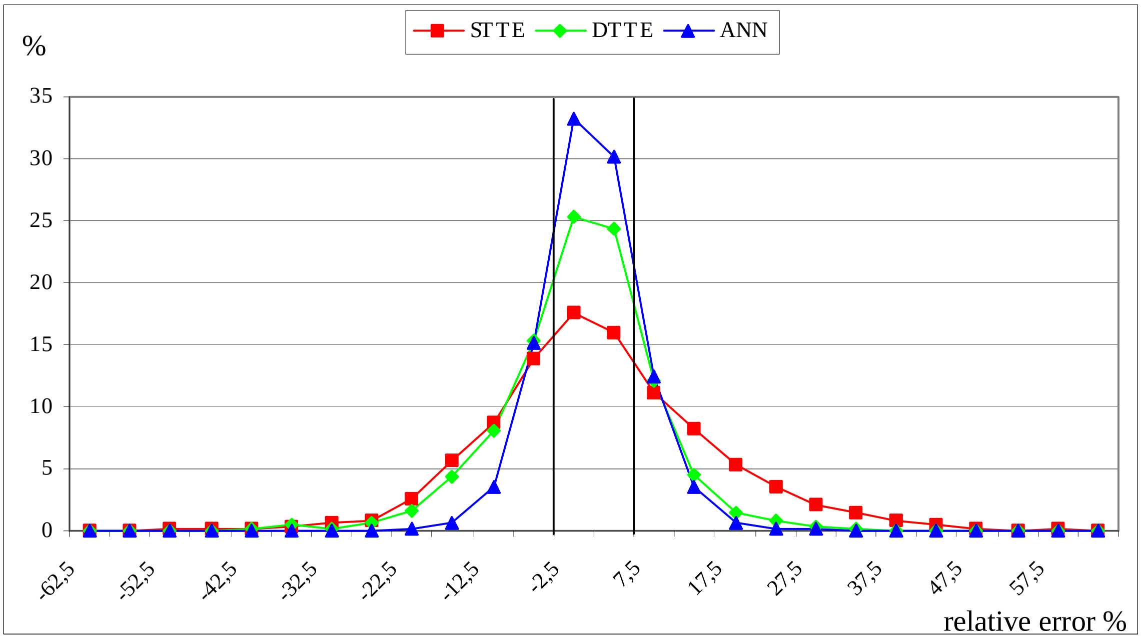

**FIGURE 6 Categorised relative error percentage per method.**

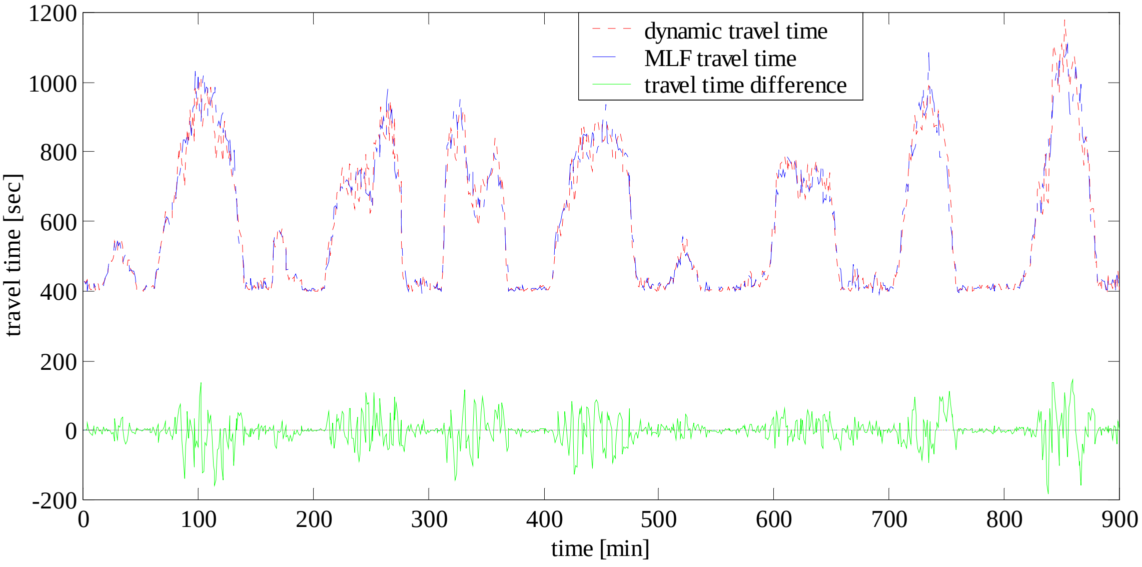

**FIGURE 7 Dynamic travel time (a posteriori estimated) and MLF travel time (predicted).**

**TABLE 1 Travel Time Estimation Algorithms**

| | Link method | | Route method | | Smoothing | |
|---|---|---|---|---|---|---|
| | Speed | Mass balance | Static | Dynamic | Input | Output |
| RT0 | X | X | X | | X | X |
| RT1 | X | X | X | | X | |
| RT2 | X | X | X | | | |
| RT3 | X | | | X | X | |
| RT4 | X | | | X | | |

**TABLE 2 Weights of the Speed and the Mass Balance Method**

| Minimum speed a & b | Weights speed method | Weights mass balance method |
|---|---|---|
| ≥ 85 km/hour | 0,40 | 0,60 |
| ≥ 35 km/hour & < 85 km/uur | 0,27 | 0,73 |
| ≥ 20 km/hour & < 35 km/hour | 0,13 | 0,87 |
| < 20 km/hour | 0,00 | 1,00 |

**TABLE 3 MRE, MARE, MTE, MATE and $r^2$ Results**

| Error measurement | STTE | DTTE | ANN |
|---|---|---|---|
| MRE [%] | 1.06 | -1.71 | -0.249 |
| MARE [%] | 10.7 | 6.91 | 4.61 |
| MTE [*sec*] | -2.30 | -7.75 | -0.107 |
| MATE [*sec*] | 79.5 | 55.0 | 35.1 |
| $r^2$ | 0.816 | 0.874 | 0.957 |