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

TRAIL Research School, Delft, November 2006

Authors
Hao Liu MSc
Dr. ir. Hans van Lint
Prof. dr. ir. Henk van Zuylen

Faculty of Civil Engineering and Geosciences,
Delft University of Technology,

© 2006 by H. Liu, H. van Lint, H. van Zuylen and TRAIL Research School
## Contents

Abstract

1 Introduction ....................................................................................................... 5

2 The State of the Art ........................................................................................... 5

3 Motivation of Research Approaches ........................................................................ 6

4 Methodology ...................................................................................................... 6

4.1 Modeling Traffic on a Single Signalized Segment ............................................. 6

4.2 Training Neural Networks ............................................................................... 8

4.2.1 Batch Training ............................................................................................... 8

4.2.2 Online Training ............................................................................................. 9

4.3 Modeling Traffic on a Signalized Arterial ............................................................ 10

4.3.1 Route Travel Time Split for offline Training ................................................. 10

4.3.2 Travel Time Prediction ................................................................................. 10

5 Evaluation Setup ............................................................................................. 11

5.1 Models for Comparison ............................................................................... 11

5.2 Qualitative Criteria for Evaluation ................................................................. 12

5.3 Simulation Data and Empirical Data ............................................................... 12

6 Result Analysis and Comparison .................................................................. 13

6.1 Predictive Performance ............................................................................... 13

6.2 Model Complexity ....................................................................................... 14

7 Conclusions and Further Research ............................................................... 15

Reference ............................................................................................................. 16
Abstract

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.

Keywords

Neural networks, urban arterials, travel time prediction
1 Introduction

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. Because of controlled conflicts, priorities, pedestrian, etc., the delay highly effects the travel time over a signalized arterial. Artificial neural networks have been shown to be a promising solution to predict traffic variables on urban arterials. Several promising neural network models have been proposed on queue prediction (delays can be derived from queues), for example (Chang, 1995; Ledous, 1997). 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.

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 future research.

2 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 (Sisiopiku et al, 1997). The authors found the following limitations of previous research: a) most existing models are link-specific and site-dependent; 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 (Robinson, 2004) 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.
Lin (2004) 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. However, the effort needed for a detailed calibration of the transition matrix is extremely time-consuming and tedious. Liu (2006) 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 (Chang, 1995; Ledous, 1997) have been developed. However, training these neural network models require direct measurements of queue lengths, which are difficult to be measured in real world.

### 3 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 (Lighthill, 1955) 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. The main assumption in both 1st and 2nd order macroscopic models is that 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 model in 2000 HCM has been widely accepted to derive delays. However, this model implicitly assumes 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.

### 4 Methodology

#### 4.1 Modeling Traffic on a Single Signalized Segment

The basic segment of urban streets is defined as a link in figure 1(a). A typical intersection with four branches, which can be assembled with eight basic segments (shown in figure 1(b)). Before we model the traffic on a basic segment, 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, 1990). The RNN has a common structure in which current states are functions of previous states and the inputs in the previous time period. 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
Neural Network Based Traffic Flow Model for Urban Arterial Travel Time Prediction

$g(k)$ 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)

Figure 1: (a) Basic segment for an urban arterial  
(b) An intersection assembled with basic segments

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.

Figure 2: Recurrent neural network topology for basic urban segment

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...
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).

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

\[
S(k) = \begin{bmatrix}
  s_1(k) \\
  s_2(k) \\
  \vdots \\
  s_m(k)
\end{bmatrix} = \begin{bmatrix}
  h(\sum_{i=1}^{2} w_{i1}^{m} x_i(k) + \sum_{i=1}^{2} w_{i2}^{m} x_i(k-1) + \nu_1^{m} b_m) \\
  h(\sum_{i=1}^{2} w_{i1}^{m} x_i(k) + \sum_{i=1}^{2} w_{i2}^{m} x_i(k-1) + \nu_2^{m} b_m) \\
  \vdots \\
  h(\sum_{i=1}^{2} w_{i1}^{m} x_i(k) + \sum_{i=1}^{2} w_{i2}^{m} x_i(k-1) + \nu_1^{m} b_m)
\end{bmatrix}
\]

(2)

where: \(s_m\) denotes the value of the \(m^{th}\) hidden neuron, \(w_{im}^{m}\) denotes the weight connecting the \(i^{th}\) input neuron and the \(m^{th}\) hidden neuron, \(w_{em}^{m}\) denotes the weight connecting the \(e^{th}\) hidden neuron and the \(m^{th}\) context neuron, \(\nu_m^{m}\) 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, \(h(\cdot)\) 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.

\[
Y(k) = h(\sum_{i=1}^{2} w_{i1}^{o} s_i(k) + \nu_1^{o} b_1)
\]

(4)

where \(w_{i1}^{o}\) denotes the weight connecting the \(i^{th}\) hidden neuron and the output neuron; \(\nu_1^{o}\) denotes the weight of bias associated with the output neuron; \(b_1\) denotes a bias with fixed value 1 for the output neuron.

### 4.2 Training Neural Networks

Two main training paradigms have emerged: batch learning and online learning.

#### 4.2.1 Batch Training

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 (Bishop, 1995). 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

\[
E = \frac{1}{2} \sum_{i=1}^{N} (e_i)^2
\]

(5)

in which, \(e_i = d_i - y_i\) 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. The new parameter set is updated according to the rule

$$
\psi_{\text{new}} = \psi_{\text{old}} - (H + \bar{A}I)^{-1}J^T \epsilon(\psi_{\text{old}})
$$

$$
J = \frac{\partial Y}{\partial \psi}
$$

$$
H = \frac{\partial^2 Y}{\partial \psi^2} = J^T J
$$

(6)

$J$ and $H$ are the first and second derivative of the output with respect to parameters respectively. For implementation details see (Demuth et al 1998).

### 4.2.2 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 (Singhal et al, 1989). A neural network’s behavior can be formulated by the following nonlinear discrete-time system:

$$
\psi(k) = \psi(k-1) + \delta(k-1)
$$

$$
Y(k) = G(X(k), \psi(k)) + \xi(k-1)
$$

(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

$$
\psi_0 = E[\psi]
$$

$$
P_0 = E[(\psi - \psi_0)(\psi - \psi_0)^T]
$$

(8)

2) Time update for $k=1, 2, \ldots$

$$
\tilde{\psi}_{k-1} = \psi_{k-1}
$$

$$
P_{k-1} = P_{k-1} + E[\delta_{k-1}\delta_{k-1}^T]
$$

(9)

3) Measurement update for $k=1,2,\ldots$

$$
e_k = d_k - G(X_k, \psi_{k-1})
$$

$$
K_k = \frac{P_{k-1}J_k^T}{P_{k-1}J_k^T P_{k-1} + E(e_k e_k^T)}
$$

$$
\psi_k = \psi_{k-1} + K_k e_k
$$

$$
P_k = P_{k-1} - K_k J_k P_{k-1}
$$

(10)
4.3 Modeling Traffic on a Signalized Arterial

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.

4.3.1 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 us assume travel time $t(k)$ is RTT at time instant $k$. Now, we need to split it into two parts, one is travel time $t_{i-3}(k)$ along segment $i-3$, the other is $t_{i}(k)$ along segment $i$. It is reasonable to assume that the segment travel time is proportional with respect to degree of congestion $N/g_s$ and the length of segment $L$. RTT is subdivided into two STTs as follows

$$t(k) = t_{i-3}(k) + t_i(k)$$

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).

4.3.2 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

$$t_{i-1}(k) = G(N_{i-1}(k), A_{i-1}(k), D_{i-1}(k), g_{i-1}, C_{i-1}, \psi_{i-1})$$

$$t_{i-2}(k) = G(N_{i-2}(k), A_{i-2}(k), D_{i-2}(k), g_{i-2}, C_{i-2}, \psi_{i-2})$$

$$t_{i-3}(k) = G(N_{i-3}(k), A_{i-3}(k), D_{i-3}(k), g_{i-3}, C_{i-3}, \psi_{i-3})$$

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

$$D_{i-1}(k) = N_{i-1}(k) + A_{i-1}(k)$$

$$D_{i-2}(k) = N_{i-2}(k) + A_{i-2}(k)$$

$$D_{i-3}(k) = N_{i-3}(k) + A_{i-3}(k)$$

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 $\beta_{i-1}^{-1}$, $\beta_{i-2}^{-2}$, $\beta_{i-3}^{-3}$ 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 $t_{t_i,3}(k)< both t_{t_i,1}(k) and t_{t_i,2}(k)$, then
$D_{i,1}(k+t_{t_i,3}(k))$ and $D_{i,2}(k+t_{t_i,3}(k))$ can be calculated by interpolation from equation 17.
If $t_{t_i,2}(k) < t_{t_i,3}(k) < t_{t_i,1}(k)$, then
$D_{i,1}(k+t_{t_i,3}(k))$ can be calculated by interpolation from equation 17.
$D_{i,2}$ is only available up to time $k+t_{t_i,2}(k)$, thus prediction of $D_{i,2}$ from time $k+t_{t_i,2}(k)$ to $k+t_{t_i,3}(k)$ is required. Since $D_{i,2}$ can be derived from $A_{i,2}$, prediction of $A_{i,2}$ from time $k+t_{t_i,2}(k)$ to $k+t_{t_i,3}(k)$ 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 $A_{i,2}(k+t_{t_i,2}(k))/A_{i,2}(hist)(k+t_{t_i,2}(k))$ ($A_{i,2}(hist)(k+t_{t_i,2}(k))$ is historical average value).
If $t_{t_i,1}(k) < t_{t_i,3}(k) < t_{t_i,2}(k)$ or both $t_{t_i,2}(k)$ and $t_{t_i,1}(k)< t_{t_i,3}(k)$, the calculation is same as above.

![Figure 4: (a) A urban route assembled with several basic segments (b) An intersection assembled with several basic segments](image)

### 5 Evaluation Setup

#### 5.1 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). Predicted travel time $T_p$ is calculated as the following equation:

$$T_p(k) = T_a(k) + E(\eta(T_a - T_d))$$  \hspace{1cm} (15)

For convenience,

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

5.2 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
  \[ MRE = 100 \frac{1}{N} \sum_{n=1}^{N} \left| \frac{y_n - d_n}{d_n} \right| \]

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

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

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.

5.3 Simulation Data and Empirical Data

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. 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](image)
The test bed for this study is selected on a provincial road connecting two motorways, A4 and A13 (see in figure 5). 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.

6 Result Analysis and Comparison

6.1 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. The models using offline training algorithm (ML 1 and 3) outperform the models using online training algorithm (ML 2 and 4). 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).

ML 1 and ML 2 slightly outperform than ML 3 and ML 4 respectively. Due to application-specific circumstances we argue that ranking ML 1, 2 higher than ML 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%.

<table>
<thead>
<tr>
<th></th>
<th>ML 1</th>
<th>ML 2</th>
<th>ML 3</th>
<th>ML 4</th>
<th>ML 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date set A</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>Date set B</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>
More specifically, the results obtained from data of Sep 30, 2004 with time interval of 4 minutes are depicted in figure 6. It illustrates the performance of these five models on the data of a single day. It is in line with the above discussion.

6.2 Model Complexity

Figure 7 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, ML 3 needs more weight parameters and training time than ML 1, especially in terms of high desired accuracy.
7 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: a) 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; b) 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; c) using offline training algorithms yields more accurate predictions than by online training algorithms.

Given the promising results of this study, further research topics should include: a) methodologies for data screening and checking, like online outlier detection should be investigated; b) online training algorithms are needed to be improved with some strategies, such as weight constraint, etc.
Reference