

Freeway Travel Time Prediction by Using the GA-Based Hammerstein Recurrent Neural Network

Ru-Kam Lee¹, Yi-Che Yang¹, Jun-Hong Chen¹,
and Yi-Chung Chen^{2(✉)}

¹ Department of Information Engineering and Computer Science,
Feng Chia University, Taichung, Taiwan, R.O.C.
kiam04@hotmail.com, a3l04l23@gmail.com,
ken837l5@gmail.com

² Department of Industrial Engineering and Management,
National Yunlin University of Science and Technology, Yunlin, Taiwan, R.O.C.
mitsukoshi90l@gmail.com

Abstract. Freeway travel time prediction has become a focus of research in recent years. However, we must understand that most conventional methods are very instinctive. They rely on the small amount of real-time data from the day of travel to look for historical data with similar characteristics and then use the similar data to make predictions. This approach is only applicable for a single day and cannot be used to predict the travel time on a day in the future (such as looking up the travel time for the coming Sunday on a Monday). This study therefore developed a Hammerstein recurrent neural network based on genetic algorithms that learns the freeway travel time for different dates. The trained model can then be used to predict freeway travel time for a future date. The experiment results demonstrated the validity of the proposed approach.

Keywords: Freeway travel time prediction · Recurrent neural network · Genetic algorithm

1 Introduction

Travel time prediction enables road users and traffic management to accurately predict travel time. This is needed because the travel time of the same route may vary greatly with what time of the day it is, whether said day is a holiday, and how the weather is. Conventional predictions use the real-time data of the travel day to look for days with similar data in the historical records and then using the data of those days to make predictions. However, this approach is only applicable for a single day and cannot be used to predict the travel time on a day in the future (such as the travel time during a long weekend), so a new method is needed to resolve this issue.

After reading relevant studies on travel time prediction, we found that existing methods rely heavily on historical data, so when the conditions of the historical data differ from those of the travel route in question, the prediction accuracy will be greatly decreased. For instance, suspended tolls and high-occupancy vehicle restrictions are often implemented in Taiwan during special holidays. Suppose that the government does

not implement these measures one year; past conditions will not be the same, and so historical data will not be of use. Furthermore, previous prediction methods based on historical traffic flow generally divided historical data into groups first [11] before comparing them with the current traffic flow. This approach generally fails to make accurate predictions when there are unpredictable traffic issues, such as automobile accidents. This is another issue that warrants the development of new prediction methods.

To overcome the issues of insufficient data for special circumstances and poor response to unexpected situations, we incorporated the genetic algorithm-based Hammerstein recurrent neural network (RNN) [10]. In the event of insufficient historical traffic flow data, the k -nearest neighbor algorithm generally needs several learning cycles before making effective predictions. In contrast, an RNN only needs one learning cycle and thus offers significantly greater efficiency. In existing travel time prediction methods, sudden incidents often make real-time data incomparable with historical data, which then results in inaccurate predictions. An RNN can use the results of the time before sudden incidents to effectively predict the changes in traffic flow after the incidents, thereby solving this issue. Furthermore, the goal of prediction is to obtain the optimal solution of the model without considering the training time of the model. Thus, we adopted a genetic algorithm rather than the conventional back propagation approach to train the target RNN. The simulation results in this study demonstrate the validity of the proposed approach.

2 Related Works

2.1 Works Regarding Freeway Travel Time Prediction

The majority of recent studies regarding freeway travel time prediction employed the k -NN method, which we introduce below. First developed by Benedetti [2], Stone [8], and Tukey [9] based on the concept of nearest neighbors, the k -NN method searches historical data for data with characteristics similar to those of the real-time data. Clark [3] employed this approach to conduct cross-analysis of traffic flow, occupancy rates, and velocity of vehicles, whereas Tsai [11] modified Clark's method, set different detector parameters, and added the dimension of time to develop an even more accurate model:

$$tss = \sum_{i=1}^L \sum_{j=1}^T \left[w_q \left(q_{ij}^r - q_{ij}^h \right)^2 + w_v \left(q_{ij}^r - q_{ij}^h \right)^2 \right], \quad (1)$$

where tss is the sum of the squares; L denotes the total number of detectors; T is the time length; w indicates a weight coefficient; q denotes traffic flow; v is velocity; r and h indicate real-time data and historical data, respectively; i denotes the time point, and j is the detection value.

2.2 Works Regarding Recurrent Neural Networks

The use of RNNs in place of time series modeling is a well-known category in the discipline of system identification [6, 10]. In this discipline, RNNs have been

demonstrated to be one of the most efficient methods to process complex and dynamic problems. However, no RNN algorithm or model has been truly recognized [7]. As numerous RNN structures exist, researchers generally find the most suitable RNN structure for a particular problem by trial-and-error, which is very time-consuming. Thus, researchers have begun developing automated RNNs in recent years to develop algorithms. Such algorithms can automatically complete the process of system identification and generally include effective parameter initialization methods and learning algorithms that can operate steadily with online parameters. In recent years, RNNs with block-oriented (BO) models have been considered the most suitable models to solve dynamic nonlinear problems. For instance, the Hammerstein, Wiener, or Hammerstein-Wiener models all comprise linear dynamic subsystems and nonlinear static subsystems and are widely applied in system identification. Westwick and Kearney [4] used the Hammerstein model to identify stretch reflex dynamic systems. Kalafatis et al. [1] successfully applied the Wiener model to PH processes.

3 Algorithm

This section introduces the proposed algorithm, including (1) the structural design of the eTag traffic data and (2) the RNN.

3.1 Structural Design of eTag Traffic Data

Format of raw data: Table 1 shows the raw data obtained from the traffic database of the Taiwan Area National Freeway Bureau of the Ministry of Transportation and Communications, which include time and date, origin, destination, type of vehicle, travel time, and traffic flow. From this table, we can see that at 12:55 pm on February 11, 2015, 46 vehicles of type 31 (Light vehicles) took 45 min to travel the route from detector 01F0017N to detector 01F0005N.

Table 1. Raw traffic data

Date and time	Origin	Destination	Vehicle type	Travel time (min)	Traffic flow (vehicles)
2017/2/11 12:55	01F0017N	01F0005N	31	45	46
2017/2/11 12:55	01F0017N	01F0005N	32	47	6
2017/2/11 12:55	01F0017N	01F0005N	41	48	2

Format of data for prediction calculation: Table 2 shows the data format converted from the raw data for the prediction calculations in this study, the fields of which include number, origin, destination, total traffic flow, average travel time, date, and time. The main differences were the added numbers to each tuple of data, the separation

of date and time to facilitate grouping later on, and the calculation of total traffic flow and average travel time for further analysis.

Table 2. Traffic data for prediction calculation

No.	Origin	Destination	Total traffic flow (vehicles)	Average travel time (min)	Date	Time
0001	01F0017N	01F0005N	54	46	2017/2/11	12:55
0002	03F0116N	01F0099N	9	36	2017/2/11	23:30
0003	01F0061S	01F0099S	62	152	2017/2/12	18:30

3.2 Hammerstein Recurrent Neural Network

In this next section, we explain the Hammerstein RNN developed in this study in detail, including the model design and the derivation of the model algorithm.

Model Design

In model design, a Hammerstein model [5] refers to a BO model with a static nonlinear subsystem in front and a dynamic linear subsystem in back. Based on this subsystem arrangement, we can design an RNN with the characteristics of a Hammerstein model, as shown in Fig. 1. This model has a four-layer framework, including the input layer, hidden layer, dynamic layer, and output layer. As the name suggests, the input layer is used to receive input data and then transfer the data received to the other layers in the neural network. The neuron in the middle does nothing. The hidden layer of the neural network is responsible for constructing the status nonlinear subsystem. The neurons of this layer have no recurrent items, which means that they are static. The nonlinear function used by the neurons in this layer is responsible for processing the nonlinear part of the static nonlinear subsystem. Next, the dynamic layer and the output layer are in charge of constructing the dynamic linear subsystem. The dynamic layer uses the recurrent items to process the dynamic part, and the output layer processes the linear part. Below, we use equations to introduce the structural details of our model. For the sake of convenience, we considered the most widely used tangent sigmoid function as the target nonlinear function of the neural network. In the neurons of layer j , we $u_i^{(j)}(k)$, $f_i^{(j)}(k)$, and $o_i^{(j)}(k)$ to represent the input of neuron i at time k , the input of the function, and the output.

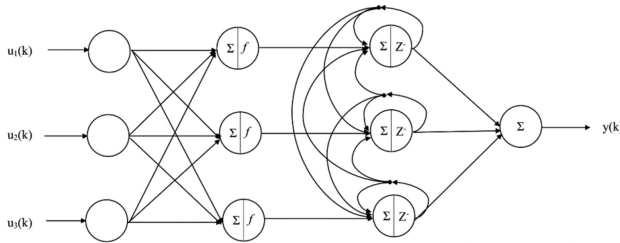


Fig. 1. The structure of Hammerstein recurrent neural network

First is the input layer. As the input layer does not do anything, its output is directly designated as the input:

$$o_i^{(1)}(k) = f_i^{(1)}(k) = u_i^{(1)}(k). \quad (2)$$

Next is the hidden layer, the input of which undergoes the calculations of the tangent sigmoid function. Thus, the equation of this layer can be expressed as

$$f_i^{(2)}(k) = \sum_{j=1}^p w_{ij} o_j^{(1)}(k) + d_i(k), \quad (3)$$

$$o_i^{(2)}(k) = \frac{\exp(f_i^{(2)}(k)) - \exp(-f_i^{(2)}(k))}{\exp(f_i^{(2)}(k)) + \exp(-f_i^{(2)}(k))}, \quad (4)$$

where w denotes the weight value between the input layer and the hidden layer; d is a bias value, and $\exp(\bullet)$ represents an exponential function.

Next in the dynamic layer, the neurons use their previous output and the current output of the hidden layer to calculate their current output. The equation can be written as

$$f_i^{(3)}(k+1) = \sum_{j=1}^q a_{ij} x_j(k) + \sum_{m=1}^p b_{im} o_m^{(2)}(k), \quad (5)$$

$$o_i^{(3)}(k) = x_i(k) = f_i^{(3)}(k). \quad (6)$$

Finally, the output layer integrates the data values from the dynamic layer and sends out the final results. Its equation is thus:

$$o_i^{(4)}(k) = f_i^{(4)}(k) = \sum_{j=1}^r c_{ij} x_j(k). \quad (7)$$

Use of Genetic Algorithm to Train Proposed Recurrent Neural Network

We used a genetic algorithm to train the proposed recurrent neural network. Of particular note, we used a genetic algorithm rather than the conventional back propagation algorithm for training because genetic algorithms can help us find the global optimum of the proposed neural network, whereas back propagation cannot. During our training process, we changed each weight value, including \mathbf{w} , \mathbf{d} , \mathbf{a} , \mathbf{b} , and \mathbf{c} , into 8-bit 01 series. The fitness function of the evolution process can be written as

$$Error(\mathbf{w}, i) = 1/2(y_d(i) - y(i))^2 = 1/2error(i)^2, \quad (8)$$

where $error(i)$ represents the error between the ideal output and the network output. With the settings of these two items and the genetic algorithm, we can complete the training of the target RNN.

4 Simulation Results

We examined three routes in this study: Yuanshan-Donghu, Fengyuan-Houli, and Tainan-Madou, chosen for their different traffic flows distributions. The Yuanshan-Donghu route has evenly distributed traffic flows throughout the day, the Fengyuan-Houli route is generally congested in the mornings and afternoons, and the traffic flow in the Tainan-Madou route is greatest around noon. These different traffic flows distributions

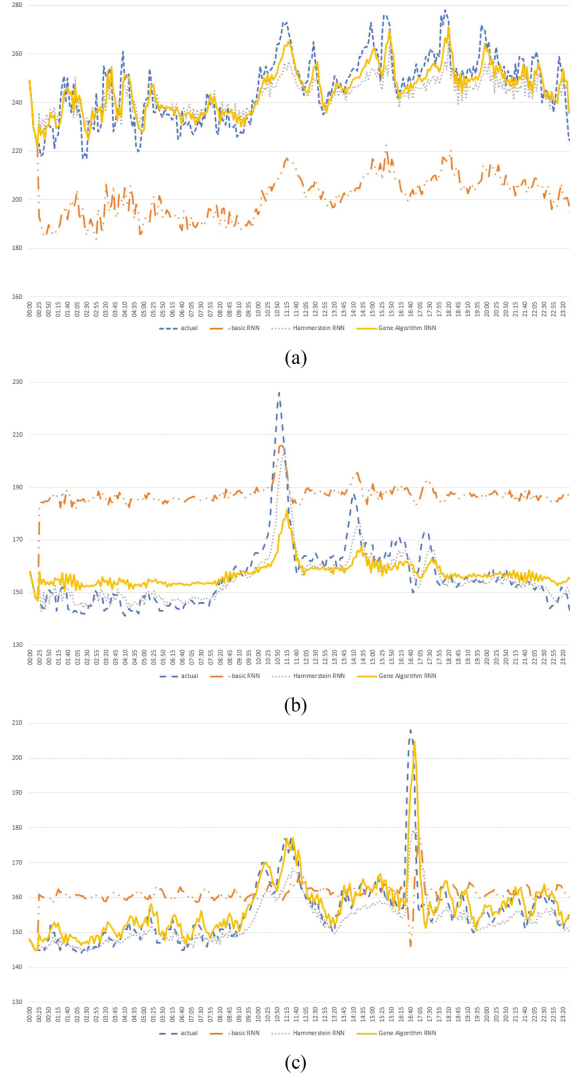


Fig. 2. Comparison of proposed and conventional methods; (a) Yuanshan-Donghu, (b) Fengyuan-Houli, (c) Tainan-Madou

Table 3. Comparison of proposed and conventional methods (MSE)

(MSE)	Basic RNN	Hammerstein RNN + BP	Hammerstein RNN + GA
Yuanshan-Donghu	45.6154	0.0087	0.0066
Fengyuan-Houli	32.2171	0.0087	0.0145
Tainan-Madou	11.0755	0.0178	0.0138

allowed us to verify the performance of the proposed model. We designed a basic RNN and a Hammerstein-based RNN+ back propagation training algorithm to simulate the three different traffic flow patterns and compared the results with those of a Hammerstein-based RNN+ GA training algorithm. The raw data and a comparison of the results of the two methods are presented in Fig. 2. To help the readers compare the two methods, we list the errors of the three methods in Table 3. As can be seen, the Hammerstein model produces significantly better prediction results than the conventional approach.

5 Conclusions

Existing methods used to predict freeway travel time are generally complex and inaccurate and rely heavily on historical data. Thus study therefore proposed a recurrent neural network based on the Hammerstein model using system identification methods for freeway travel time prediction. The proposed approach is significantly more accurate than conventional methods. However, as there are many types of BO models, we will experiment with other models in the future and make further improvements on prediction performance based on the approach proposed in this study.

Acknowledgement. This work was supported in part by the Ministry of Science and Technology of Taiwan, R.O.C., under Contracts MOST 105-2119-M-035-002 and MOST 105-2634-E-035-001. Also, We are grateful to the National Center for High-performance Computing in Taiwan for computer time and facilities.

References

1. Kalafatis, A.D., Wang, L., Cluett, W.R.: Linearizing feedforward-feedback control of PH processes based on the Wiener model. *J. Process Control* **15**, 103–112 (2005)
2. Benedetti, J.: On the nonparametric estimation of regression functions. *J. Roy. Stat. Soc. B* **39**, 248–253 (1977)
3. Clark, S.: Traffic prediction using multivariate nonparametric regression. *J. Transp. Eng.* **129** (2), 161–168 (2003)
4. Westwick, D.T., Kearney, R.E.: Separable least squares identification of nonlinear Hammerstein models: application to stretch reflex dynamics. *Ann. Biomed. Eng.* **29**, 707–718 (2001)
5. Wang, J.S., Hsu, Y.L.: An MDL-based Hammerstein recurrent neural network for control applications. *Neurocomputing* **74**, 315–327 (2010)

6. Wang, J., Zhang, Y., Gao, Y., Xing, C.: PLSM: a highly efficient LSM-tree index supporting real-time big data analysis. In: *Proceedings of International Conference on Computer Software and Applications*, pp. 240–245 (2013)
7. Gori, M., Mozer, M., Tsoi, A.C., Watrous, R.L.: Presenting the special issue on recurrent neural networks for sequence processing. *Neurocomputing* **15**, 181–182 (1997)
8. Stone, C.: Consistent nonparametric regression. *Ann. Stat.* **5**, 595–645 (1977)
9. Tukey, J.: *Exploratory Data Analysis*. Addison Wesley, Reading (1977)
10. Chow, T.W.S., Fang, Y.: A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics. *IEEE Trans. Ind. Electron.* **45**(1), 151–161 (1998)
11. Tsai, C.K.: Freeway travel time prediction by using the k-NN method and comparison of different data classification. Master thesis of Department of Transportation & Logistics Management, National Chiao Tung University (2009)

Genetic and Evolutionary Computing
Proceedings of the Eleventh International Conference
on Genetic and Evolutionary Computing, November 6-8,
2017, Kaohsiung, Taiwan
Lin, J.C.-W.; Pan, J.-S.; Chu, S.-C.; Chen, C.-M. (Eds.)
2018, XIII, 208 p. 80 illus., Softcover
ISBN: 978-981-10-6486-9