Postprint: Short-term Traffic Flow Speed Prediction Model Based on Time Series and BP-ANN

Authors: Tian Ruijie, Zhang Weishi, Zhai Huawei

Date: 2018-08-13T00:00:00+00:00

Abstract

Short-term traffic flow speed prediction is considered a crucial component of intelligent transportation systems, where the accuracy of the prediction model determines, to a certain extent, the performance of real-time traffic control and management. To address the limitations of existing traffic flow speed prediction models that utilize a single dataset and a singular model architecture, a hybrid prediction model integrating time series analysis with artificial neural networks is proposed. The model employs time series methods to separately model and forecast real-time and historical data, while utilizing an artificial neural network to adjust the predicted values from both data sources. Experimental results demonstrate that the proposed prediction model can control the prediction error within 7% and effectively predict short-term traffic flow speed under various input parameters.

Full Text

Preamble

Short-term Traffic Flow Velocity Prediction Model Based on Time Series and BP-ANN

Tian Ruijie, Zhang Weishi, Zhai Huawei†

(School of Information Science & Technology, Dalian Maritime University, Dalian, Liaoning 116026, China)

Abstract: Short-term traffic flow velocity prediction is considered a critical component of intelligent transportation systems, where the accuracy of prediction models directly determines the effectiveness of real-time traffic control and management. Existing traffic flow velocity prediction models suffer from reliance on single datasets and monolithic model architectures. To address these limitations, this paper proposes a hybrid prediction model that integrates time series analysis with artificial neural networks. The model employs time series methods to separately model and predict real-time and historical data, then utilizes an artificial neural network to adjust the predicted values from both sources. Experimental results demonstrate that the proposed model can constrain prediction errors within 7% and effectively predict short-term traffic flow velocity under varying input parameters.

Keywords: time series; artificial neural network; short-term prediction; traffic flow velocity

1 Related Research

Traffic prediction has become a significant research topic in intelligent transportation systems, playing a vital role in daily life. Traffic road prediction constitutes a core function of widely-used intelligent vehicle navigation devices and electronic maps, particularly short-term traffic road prediction. By enabling real-time predictive analysis of road conditions, these tools help users avoid congested routes and provide more reasonable and diverse travel route options. Macroscopic traffic operation characteristics are measured by flow, density, and speed, while the most critical evaluation metrics for bus lane benefit assessment include travel speed, corridor passenger capacity, and road user travel time savings. Therefore, this paper selects traffic flow velocity as the indicator to reflect bus lane conditions.

Research methods can be broadly categorized into two approaches: parametric and non-parametric methods. Typical parametric methods employ time series models. For instance, Lai et al. utilized time series models and linear regression for high-speed railway short-term passenger flow prediction. Their primary contribution involved combining two models to process different data portions, integrating them through least absolute values to generate final predictions and improve accuracy. However, their work focused on urban traffic flow velocity prediction. For non-parametric approaches, Moreno et al. adopted fuzzy C-means clustering to estimate road traffic flow velocity, clustering instantaneous speed data into classification levels to obtain a membership matrix and cluster center vector, ultimately deriving average speed. This study failed to account for the impact of abnormal instantaneous speed variations when buses enter and exit stations, which significantly affects clustering results and prediction accuracy.

Cao designed a BP neural network-based urban arterial traffic flow prediction model using historical data as training and test sets. This approach only reflected the periodicity and universality of road traffic flow without incorporating real-time road conditions, thus failing to capture the particularity of traffic flow. Data singularity represents a notable deficiency in this model. Furthermore, Zhang noted that neural network models excel at processing large volumes of historical data with nonlinear characteristics but overlook linear relationships within the data. In response, Li et al. developed a short-term traffic flow prediction model combining ARIMA and Radial Basis Function Artificial Neural Network (RBF-ANN). Their model used ARIMA to model the linear components of traffic flow time series and RBF-ANN to capture nonlinear components through ARIMA residuals. However, this research did not investigate peak and off-peak traffic periods, meaning the model' s performance would be severely degraded when traffic conditions changed abruptly during congestion.

In summary, short-term traffic flow velocity prediction exhibits inherent characteristics of strong uncertainty and weak regularity. Relying on a single model for prediction makes it difficult to further improve accuracy. To address this, this paper proposes a Time Series-Artificial Neural Network (TS-ANN) hybrid model that overcomes the limitations of single models by leveraging their complementary strengths to construct a robust short-term traffic flow velocity prediction framework. Predicting traffic flow velocity under the TS-ANN model represents an important step toward developing smart cities.

2 The TS-ANN Model

This section presents the TS-ANN prediction model, with its architecture illustrated in [Figure 1: see original paper]. Real-time data $\bar{V}_{\text{real-time}} = \{\bar{v}_1, \bar{v}_2, \bar{v}_3, \dots, \bar{v}_m\}$ represents real-time average road segment speed (km/h) information, while historical data $\bar{V}_{\text{historical}} = \{\bar{v}_1, \bar{v}_2, \bar{v}_3, \dots, \bar{v}_n\}$ represents historical average road segment speed (km/h) information. Subsets $V_r = \{\bar{v}_i, \bar{v}_{i+1}, \dots, \bar{v}_j\}$ (with $i \geq 1, j \leq m$) and $V_h = \{\bar{v}_i, \bar{v}_{i+1}, \dots, \bar{v}_j\}$ (with $i \geq 1, j \leq n$) are selected from $\bar{V}_{\text{real-time}}$ and $\bar{V}_{\text{historical}}$ respectively as time series data for modeling and analysis to predict the next time period' s average road segment speed. The time interval between adjacent elements in V_r is $t = 1$ minute, while in V_h it is $t = 1$ week.

During time series modeling, the first step involves stationarity testing of the time series data, followed by examination of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) values. The most suitable model is selected from AR(p), MA(q), and ARMA(p,q) based on these observations. The lag orders p and q are then determined using the Akaike Information Criterion (AIC), Schwarz Criterion (SC), and Hannan-Quinn Criterion (HQC). Finally, model fitting is validated by testing whether the residual sequence constitutes Gaussian white noise. In the TS-ANN model, prediction horizon and prediction time are considered factors affecting prediction accuracy (discussed in detail in Section 2.1.3). The prediction horizon unit is minutes, and prediction time is the sum of current time and prediction horizon. Since this paper considers two prediction time states—peak and off-peak periods—prediction time is set as time $\{1, 2\}$, where 1 represents peak periods and 2 represents off-peak periods. The time series predictions $V_{\text{real-time}}$ and $V_{\text{historical}}$, combined with prediction horizon and prediction time, serve as inputs to the artificial neural network. The TS-ANN model employs a BP neural network.

A BP network typically comprises an input layer, one or more hidden layers, and an output layer. Nodes in adjacent layers are interconnected, with each node only receiving inputs from the previous layer' s neurons. Only the input information processed by each layer' s neurons can become the output of the output layer.

Assuming the input layer, hidden layer, and output layer contain m, h, and p neurons respectively, the hidden layer' s input and output are represented by equations (1) and (2), while the output layer's input and output are represented by equations (3) and (4):

$$I_j = \sum_{i=1}^{m} w_{ji} x_i + b_j \quad (j = 1, 2, 3, \dots, h)$$

$$y_j = f_h(I_j) \quad (j = 1, 2, 3, \dots, h)$$

$$I_o = \sum_{j=1}^{h} w_{oj} y_j + b_o \quad (o = 1, 2, 3, \dots, p)$$

$$y_o = f_p(I_o) \quad (o = 1, 2, 3, \dots, p)$$

where w_{ji} represents connection weights between the input and hidden layers, w_{oj} represents connection weights between the hidden and output layers, x_i denotes input values from the input layer, I_j and I_o represent input values for the hidden and output layers respectively, b_j and b_o are thresholds for the hidden and output layers, and f_h and f_p are activation functions for the hidden and output layers. Typically, the hidden layer uses tanh or sigmoid activation functions, while the output layer uses ReLU, with corresponding formulas as follows:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$f(x) = \mathrm{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$f(x) = \mathrm{ReLU}(x) = \max(0, x)$$

Algorithm 1 outlines the TS-ANN model training process. The algorithm first randomly initializes weight values w and thresholds b, with parameter dimensions determined by the number of nodes in the input, hidden, and output layers.

In the first loop, time series models are built separately for real-time and historical data to generate predictions. The results from both models, combined with prediction horizon and prediction time, are fed into the second loop as inputs to the ANN model. The input data format is $\{V_{\text{real-time}}, V_{\text{historical}}, \text{time}, \text{PH}\}$. The ANN model is trained by comparing neural network predictions with real-world traffic data, calculating the loss function, and applying backpropagation to learn parameters. The trained model is returned at the algorithm' s conclusion.

Algorithm 2 describes the speed estimation process under TS-ANN. The time series model generates initial predictions, which are then combined with prediction horizon and prediction time as inputs for the second-stage prediction to produce the final TS-ANN result, with output data format $V_{\text{predicted}}$.

2.1 Time Series Prediction Model

The fundamental model is ARMA(p,q), expressed as:

$$Y_t = \sum_{i=1}^{p} \varphi_i Y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}$$

where Y_t (with $t = 1, 2, 3, ...$) represents the time series, ε is zero-mean white noise, φ denotes AR model parameters, θ denotes MA model parameters, p is the autoregressive order, q is the moving average order, and t is the time index. ARMA(p,q) assumes a stationary random process. However, real-world time series data often exhibit systematic trends or periodic fluctuations, indicating non-stationary random processes that cannot be directly modeled using ARMA(p,q). For non-stationary time series, differencing transformations are first applied to achieve stationarity, followed by ARMA(p,q) modeling of the stationary process. This approach constitutes the Autoregressive Integrated Moving Average model, denoted as ARIMA(p,d,q):

$$\phi(B)(1-B)^d Y_t = \theta(B)\varepsilon_t$$

where B is the backshift operator ($BY_t = Y_{t-1}$) and d is the differencing order.

2.1.1 Time Series Dataset Construction This paper obtains road segment data L9001, L9002, L9003, and L9004 from the Dalian traffic database, comprising two months of bus trajectory data from November 1, 2017, to December 29, 2017 (daily from 6:00 AM to 9:00 PM), with a 10-second time interval. The corresponding four road segments have lengths of 0.84 km, 0.98 km, 0.85 km, and 0.97 km. shows sample trajectory data. The collected trajectory data is processed using a road segment average speed estimation algorithm to generate the experimental dataset $V(t)$, where $V_{L900i}(t) = \{V_1(t), V_2(t)\}$ (for $i = 1, 2, 3, 4$).

Real-time data $V_1(t)$ represents continuous p_1 time points of road segment average speed information before a certain moment on a given day:

$$V_1(t) = \{v(t-1), v(t-2), \dots, v(t-p_1)\}$$

Historical data $V_2(t)$ represents road segment average speed information for the same time point across continuous p_2 weeks before a given moment:

$$V_2(t) = \{v(t-7\times 24\times 1), v(t-7\times 24\times 2), \dots, v(t-7\times 24\times p_2)\}$$

2.1.2 Time Series Model Construction This section selects real-time data $V_1(t)$ and historical data $V_2(t)$ from road segment L9001 for experimental analysis, with sample input data shown in and .

1) Unit Root Test

Stationarity testing is performed on time series data $V_1(t)$ from road segment L9001. As shown in , when the test equation includes an intercept term, the t-statistic for the unit root under the null hypothesis $H_0 : \delta = 0$ is -3.6492. At significance levels of 1%, 5%, and 10%, the critical values for the unit root test are -3.5683, -2.9211, and -2.5985 respectively. Since the test statistic is smaller than the corresponding critical values, the null hypothesis H_0 is rejected, indicating that the sequence is stationary and suitable for modeling.

2) Model Selection

By examining the ACF and PACF of time series $V_1(t)$—where ACF measures correlation between any two values in the time series at a specific time shift (lag), and PACF measures correlation between any two points at a specific lag while controlling for intermediate values—the characteristics of different time series models can be identified. summarizes the ACF/PACF patterns for AR(p), MA(q), and ARMA(p,q) models. Observing the ACF/PACF plots reveals that ACF shows tailing behavior at lag=2, and PACF also shows tailing at lag=2, suggesting an ARMA model with p and q values limited to the interval [1,2].

3) Lag Order Selection

AIC, SC, and HQC information criteria are used to evaluate ARMA model fit and determine optimal lag orders p and q. Based on the parameter intervals, four models are constructed: ARMA(2,2), ARMA(2,1), ARMA(1,2), and ARMA(1,1). For each model, three simulation experiments are conducted with sample sizes of 40, 50, and 60, selecting the minimum AIC, SC, and HQC values across all experiments. As shown in , ARMA(1,2) is selected as the final model because it yields the smallest AIC, SC, and HQC values, indicating optimal predictive performance.

4) Model Validation

Model validation assesses whether the model adequately represents the time series. The method examines correlation and partial autocorrelation in the residual sequence. If the model fits well, residuals should exhibit white noise characteristics. Figure 4: see original paper shows the ACF and PACF of the residual sequence, confirming it has become white noise. Figure 4: see original paper displays the fit between model predictions and actual values. Notably, during early morning peak hours, the fitting performance noticeably degrades, likely due to complex traffic conditions causing abrupt traffic flow velocity changes that reduce model performance. Repeating these steps for time series data $V_2(t)$ yields the ARIMA(2,1,2) model.

2.1.3 Model Analysis This section analyzes model performance under different prediction horizons (PH) and prediction times.

Hypothesis 1: When the prediction horizon does not exceed a certain threshold ϕ (i.e., PH $< \phi$), the ARMA(1,2) model provides more accurate predictions. When the prediction horizon exceeds this threshold (i.e., PH $> \phi$), the ARIMA(2,1,2) model yields better predictions.

Using Mean Absolute Percentage Error (MAPE) to evaluate prediction accuracy:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{X_i - Y_i}{X_i}\right| \times 100\%$$

where X_i and Y_i represent actual and predicted traffic speeds, respectively, and n denotes the number of predictions. As shown in [Figure 5: see original paper], when PH < 5 (prediction horizon less than 5 minutes), the ARMA(1,2) model demonstrates superior performance. However, when PH > 5 (prediction horizon exceeds 5 minutes), the ARIMA(2,1,2) model begins to produce better predictions. Setting $\phi = 5$, Hypothesis 1 is validated.

Hypothesis 2: The ARIMA(2,1,2) model can efficiently predict traffic flow velocity changes during peak hours, whereas the ARMA(1,2) model struggles to capture rapid road state transitions.

With a fixed prediction horizon PH $= 5$ ($\phi = 5$), [Figure 6: see original paper] compares actual speed values with ARMA(1,2) and ARIMA(2,1,2) predictions across different times on a typical workday. During the morning peak around 6:50 AM, the ARIMA(2,1,2) model shows minimal error, while ARMA(1,2) exhibits significant errors at certain time points. Similarly, near the peak period boundary around 9:05 AM, ARIMA(2,1,2) accurately predicts post-congestion speeds, whereas ARMA(1,2) produces substantially larger errors. Results demonstrate that during peak hours, ARIMA(2,1,2) achieves higher prediction accuracy than ARMA(1,2), confirming Hypothesis 2.

2.2 BP Neural Network

A BP neural network is a multi-layer feedforward neural network trained using error backpropagation. Its fundamental concept is gradient descent, employing gradient search techniques to minimize the mean squared error between actual and expected output values. The learning process consists of forward propagation and backward propagation. During forward propagation, input information passes through hidden layers for layer-by-layer processing toward the output layer. Each neuron' s state only affects the next layer' s neurons, producing an output and calculating the mean squared error between actual and expected values. If the output layer fails to achieve the desired output, the process switches to backward propagation, returning error signals along original connection paths and adjusting neuron weights to minimize the error signal.

2.2.1 BP Neural Network Structure The BP neural network comprises an input layer, hidden layer, and output layer. This paper selects four parameters—$V_{\text{real-time}}$, $V_{\text{historical}}$, time, and PH—as input layer nodes, with traffic flow velocity as the output layer. To determine the number of neurons in the hidden layer, experiments comparing 3 to 15 neurons were conducted using data from four road segments from November 1 to December 22 as the training set and December 23-29 as the test set. As shown in [Figure 7: see original paper], a BP network with 12 hidden neurons achieves optimal performance. Therefore, the hidden layer size is set to 12, establishing a neural network with 4 input nodes, 1 output node, 12 hidden neurons, a sigmoid activation function for the hidden layer, and ReLU for the output layer.

2.2.2 BP Neural Network Modeling Steps

a) **Network Structure Determination:** Define network layers, neuron quantities per layer, and activation functions.

b) **Initialization:** Set hidden layer neuron thresholds θ_j, output layer neuron threshold θ, minimum error E, learning rate η, and maximum training iterations under identical sampling data.

c) **Forward Calculation:** Compute hidden layer outputs and output layer outputs based on input samples.

d) **Error Calculation:** Calculate the mean squared error between actual and expected outputs. Terminate training if the error meets requirements.

e) **Weight Adjustment:** Repeatedly adjust each neuron' s weight values by iterating steps c)-e) until error E falls within the required range.

3 Experiments and Analysis

3.1 Evaluation Metrics

This simulation experiment employs Root Mean Square Error (RMSE) as the evaluation metric:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2}$$

where X_i and Y_i represent actual and predicted speeds, respectively, and n denotes the number of predictions.

3.2 Experimental Comparison and Analysis

Three experiments were conducted with prediction horizons PH set to 4, 5, and 6 minutes.

When PH = 4, traffic flow velocity prediction analysis was performed separately for peak and off-peak periods. As shown in Figure 8: see original paper, during off-peak periods, ARMA(1,2), ARIMA(2,1,2), and TS-ANN show generally consistent prediction trends. However, TS-ANN better reflects traffic flow velocity compared to the ARMA(1,2) model. During peak periods, the ARIMA(2,1,2) model more effectively fits the linear relationships in time series data, thereby improving prediction accuracy. The proposed TS-ANN demonstrates significantly superior prediction performance over ARIMA(2,1,2), though some time points remain poorly predicted. The primary reasons for prediction failures are: (1) these moments occur at peak period start/end boundaries, and (2) the original time series data used in experiments is limited—only two months of traffic flow velocity information. Acquiring more time series data would enable better fitting to actual values.

When PH = 6, Figure 9: see original paper reveals that during off-peak periods, ARMA(1,2), ARIMA(2,1,2), and TS-ANN show consistent data trends, with ARIMA(2,1,2) outperforming ARMA(1,2), consistent with Hypothesis 1 but without significant differences. TS-ANN better displays actual road operating speeds. During peak periods, ARMA(1,2) demonstrates the poorest fitting performance with substantial prediction errors at certain time points, while the proposed TS-ANN achieves the most accurate predictions, with RMSE as low as 4.35% during peak hours.

When PH = 5, [Figure 10: see original paper] shows the fitting between actual and predicted values at different times. Between 6:00-6:50, both ARMA(1,2) and TS-ANN exhibit the highest similarity to actual values, while ARIMA(2,1,2) shows significant deviations at certain moments—for example, at 6:15, ARIMA(2,1,2) predicts 27.6 km/h versus an actual value of 43.6 km/h. During the early peak around 6:50, the ARMA(1,2) model fails to capture actual road conditions, only reflecting congestion after approximately 7:13.

In contrast, TS-ANN predicts the onset of congestion with minimal error and successfully forecasts the road’ s true state at the end of the morning peak. Results demonstrate that TS-ANN effectively handles the influence of prediction horizon and prediction time in short-term traffic flow velocity prediction, exhibiting certain adaptability.

---

4 Conclusion

Traffic road evolution processes exhibit complexity and uncertainty, making it difficult to achieve satisfactory prediction results using single models alone. This paper proposes an urban road traffic flow velocity prediction model that fully leverages the advantages of time series models and artificial neural networks in linear and nonlinear modeling. Based on predictability analysis, time series data subsets are selected from both real-time and historical data for modeling and prediction. The predictions from both sources are then fused and adjusted through a BP neural network to obtain final results, thereby improving short-term traffic flow velocity prediction performance to some extent. However, this study does not consider the impact of weather and emergency events on road segment speeds, making the analysis of such factors the primary focus of future research.

References

[1] Feng Kai. Development status and trend of urban intelligent transportation system [J]. Global Market Information Guide, 2017, 6(1): 111-112.

[2] Sun Xiaoliang. Urban road traffic state evaluation and prediction methods and applications [D]. Beijing: Beijing Jiaotong University, 2013.

[3] Ma Jian, Sun Shouzeng, Rui Haitian, et al. Review on China’ s traffic engineering research progress [J]. China Journal of Highway and Transport, 2016, 29(6): 1-161.

[4] Lai Qingying, Liu Jun, Luo Yongji, et al. A hybrid short-term forecasting model of passenger flow on high-speed rail considering the impact of train service frequency [J]. Mathematical Problems in Engineering, 2017, 2017(3): 1-9.

[5] Moreno E G, Romana M G, Martínez O. A first step to diagnostic of urban transport operations by means of GPS receiver [J]. Procedia Computer Science, 2016, 83(44): 305-312.

[6] Cao Hongmei, Han Feng. The urban arterial traffic flow forecasting based on BP neural network [C]// Proc of the 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control. 2014: 393-397.

[7] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model [J]. Neurocomputing, 2003, 50(1): 159-175.

[8] Li Kuilin, Zhai Chunjie J, Xu Jianmin. Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN [C]// Proc of Chinese Automation Congress. 2017: 2804-2807.

[9] Box G E P, Jenkins G M, Reinsel G C, et al. Time series analysis: forecasting and control [M]. Hoboken: Wiley, 2015.

[10] Anokye R, Acheampong E, Owusu I, et al. Time series analysis of malaria in Kumasi: using ARIMA models to forecast future incidence [J]. Journal of Cogent Social Sciences, 2018, 4(9): 1-15.

[11] Bogl M, Aigner W, Filzmoser P, et al. Visual analytics for model selection in time series analysis [J]. IEEE Trans on Visualization and Computer Graphics, 2013, 19(12): 2237-2246.

[12] Medel C A, Salgado S C. Does the BIC estimate and forecast better than the AIC [J]. Revista de Analisis Economico–Economic Analysis Review, 2013, 28(1): 47-64.

[13] Wang Huiqing, Guo Zhirong, Bai Yingying. Time series classification model based on BP and Naive Bayes [J]. Application Research of Computers, 2019, 36(8). http://www.arocmag.com/article/02-2019-08-020.html

Note: Figure translations are in progress. See original paper for figures.