sustainability

MDPI

Article

Real-Time Traffic Flow Forecasting via a Novel Method Combining Periodic-Trend Decomposition

Wei Zhou [1,2,3], Wei Wang [1,2,3], Xuedong Hua [1,2,3,*] and Yi Zhang [1]

[1] School of Transportation, Southeast University, Nanjing 211189, China; veager@seu.edu.cn (W.Z.); wangwei_transtar@163.com (W.W.); zhangyie@seu.edu.cn (Y.Z.)

[2] Jiangsu Key Laboratory of Urban ITS, Nanjing 211189, China

[3] Jiangsu Province Collaborative Innovation Centre of Modern Urban Traffic Technologies, Nanjing 211189, China

* Correspondence: qdurgk@163.com

Received: 26 May 2020; Accepted: 21 July 2020; Published: 22 July 2020

Abstract: Accurate and timely traffic flow forecasting is a critical task of the intelligent transportation system (ITS). The predicted results offer the necessary information to support the decisions of administrators and travelers. To investigate trend and periodic characteristics of traffic flow and develop a more accurate prediction, a novel method combining periodic-trend decomposition (PTD) is proposed in this paper. This hybrid method is based on the principle of "decomposition first and forecasting last". The well-designed PTD approach can decompose the original traffic flow into three components, including trend, periodicity, and remainder. The periodicity is a strict period function and predicted by cycling, while the trend and remainder are predicted by modelling. To demonstrate the universal applicability of the hybrid method, four prevalent models are separately combined with PTD to establish hybrid models. Traffic volume data are collected from the Minnesota Department of Transportation (Mn/DOT) and used to conduct experiments. Empirical results show that the mean absolute error (MAE), mean absolute percentage error (MAPE), and mean square error (MSE) of hybrid models are averagely reduced by 17%, 17%, and 29% more than individual models, respectively. In addition, the hybrid method is robust for a multi-step prediction. These findings indicate that the proposed method combining PTD is promising for traffic flow forecasting.

Keywords: intelligent transportation system; traffic flow forecasting; traffic flow decomposition; time series decomposition approach; hybrid prediction method

1. Introduction

Over the past decades, with the sharp increase of car ownership, congestion has become a most troubling problem in urban areas. This problem has seriously affected travel quality in terms of prolongation of travelling time, increase of fuel consumption, and pollution of automobile exhausts. To solve a series of traffic problems, one effective way, which is universally acknowledged in present research [1,2], is to develop intelligent transportation systems (ITS). A tremendous number of sensors are distributed throughout road networks, and the transportation information is integrated into ITS. Thus, ITS makes it possible for administrators to monitor real-time traffic status and respond to traffic incidents in a timely manner. Traffic flow forecasting is regarded as the most fundamental task for ITS. For administrators, the predicted results can give early warnings of burst traffic flow and provide reliable information for reasonable management. Meanwhile for travelers, the predicted traffic flow can be a reference to choose a route with the shortest travel time.

Generally, traffic flow is defined as a measure of traffic status (i.e., total volume [3–7], average speed [1,8–10], average travel time [11,12]) in a constant time interval at a target road segment. Obviously, the traffic flow is time-varying. Traffic flow forecasting tasks can be addressed as time series problems. In present studies [13,14], traffic flow has been demonstrated with significant autocorrelation, which means it is highly related to past data. Besides that, the trend and periodicity characteristics have been found; they are the important characteristics of traffic flow. Since Ahmed and Cook [15] first proposed the traffic flow forecasting problem in 1979, much effort has been devoted to the development of methods to predict traffic flow more accurate. According to some researches [16–18], the prediction methods are mainly divided into three categories: statistical models, artificial intelligent (AI) models, and hybrid models. Owing to flexible frameworks, hybrid models have attracted more attention than others. Tremendous studies have demonstrated that combining different models is an effective approach to improve prediction accuracy. In recent studies [3,10,19–25], a novel method combining time series decomposition approaches has attracted extensive interest. The main idea of this kind of models is "decomposition first and forecasting last". In detail, the decomposition approach is employed to disaggregate the original complex time series into several regular and simple components, and then these components are predicted separately. The final outcomes are obtained by summing those predicted results. However, most decomposition approaches are originally developed to analyze digital signals. The decomposition results are a series of components with different frequencies and amplitudes.

In this paper, in order to give an insight into the characteristics of traffic flow, we proposed a novel method combining periodic-trend decomposition (PTD). PTD is developed to disaggregate the original data into three additive components, including trend, periodicity, and remainder. These three components show different characteristics of original traffic flow: the trend represents the day-to-day fluctuation; the periodicity represents variety within a day; the remainder represents noise. In the prediction stage, the periodicity is a strict period function and predicted by cycling. The trend and remainder are predicted by modelling. To test and evaluate the proposed method, traffic volume data of three detection stations, which are provided by the Minnesota Department of Transportation (Mn/DOT), are utilized to conduct experiments and analyze contrastively.

The rest of the content of this paper is organized as follows: Section 2 reviews existing literature in traffic flow forecasting. Section 3 formulates the proposed PTD approach and the hybrid prediction method. Section 4 elaborates the datasets' description and the process of experiments. Section 5 presents the results and discussions. Finally, Section 6 concludes this work.

2. Literature Review

2.1. Traffic Flow Forecasting Models

Due to time-varying nature, traffic flow is usually regarded as a kind of time series in most research. Over the past several decades, lots of models have been introduced and developed to improve the prediction accuracy. A summary of the studied literature is presented in Table 1.

Table 1. The summary of the reviewed literature.

Ref.	Method	Area	Traffic State	Prediction Interval (Minutes)	Prediction Horizon (Steps)
	Statistical models				
[14]	ARIMA	arterial	volume; speed	5	12
[11]	KF	arterials	travel time	5	1
[26]	KF	expressway	volume	5	1

Table 1. *Cont.*

Ref.	Method	Area	Traffic State	Prediction Interval (Minutes)	Prediction Horizon (Steps)
	Artificial intelligent (AI) models				
[27]	KNN	highway	volume	60	1
[28]	KNN	urban road	speed	5	5
[12]	SVR	freeway	travel time	60	1
[29]	SVR	expressway	speed	1	1
[5]	SVR	urban road	volume	5	3
[30]	ANN	urban road	volume	3	5
[31]	ANN	highway	volume	5	1
[7]	WNN	freeway	volume	60	1
[9]	LSTM	expressway	speed	2	4
[4]	SAE	freeway	volume	5;15;30;45;60	1
[32]	CNN	urban road	speed	10;20	1
[33]	HA; ARIMA; ANN; KNN	freeway	volume	15	1
[34]	DBN; LSTM	arterial	volume	10;30	1
[8]	ANN; SVR; RF; LR	urban road	speed	15	1
	Hybrid models				
[35]	ARIMA+ANN	highway	volume	8	1
[36]	ARIMA+ANN	highway	volume	60	3
[37]	ARIMA+WNN	urban road	volume	15	1
[38]	ARIMA+SVR	freeway	volume	5	1
[39]	ES+ANN	freeway	speed	1	6
[40]	ES+ELM	freeway	speed	1	1
[6]	KNN+LSTM	freeway	volume	5	1
[41]	GA+SVR	motorway	volume	60	1
[42]	GA+DBN	freeway	volume	15	1
[43]	PSO+SVR	freeway; urban road	volume	5	1
[19]	FT+SVR	freeway	volume	60	1
[10]	EMD+ARIMA	freeway	volume; speed	5	4
[3]	EMD+ANN	freeway	volume	1;2;10	10
[22]	EMD+SVR	urban road	volume	5	1
[21]	SSA+KNN	urban road	volume	0.6	1
[23]	SSA+ELM	urban road	volume	5	1
[25]	WD + GCRNN +ARIMA	urban road	speed	15	1

ARIMA: autoregressive integrated moving average; KF: Kalman filtering; KNN: k-nearest neighbor; SVR: support vector regression; ANN: artificial neural network; WNN: wavelet neural network; ELM: extreme learning machine; LSTM: long short-term memory neural network; DBN: deep belief network; CNN: convolutional neural network; SAE: stacked auto encoders; HA: historical average; ES: exponential smoothing; RF: random forecast; LR: linear regression; GA: genetic algorithm; PSO: particle swarm optimization; FT: Fourier transform; EMD: empirical mode decomposition; SSA: singular spectrum analysis; WD: wavelet decomposition; GCRNN: graph convolutional recurrent neural network.

According to studies [16–18], These models are roughly classified into three categories: statistical models, artificial intelligent (AI) models, and hybrid models.

1. Statistical models. In early studies, many researchers proposed statistical models to predict traffic flow in the assumption of linearity and stationarity. The classical statistical models include autoregressive integrated moving average (ARIMA) [14,15,33,44], Kalman filtering (KF) [11,26], historical average (HA) [33], and exponential smoothing [39,40]. These models are simple and have low computational complexity. However, because of both linearity and nonlinearity, traffic flow is complex and volatile [4,10]. For example, traffic volume increases rapidly in rush hours and decreases rapidly after rush hours. Due to the assumption of linearity, these statistical models cannot well learn those characteristics of traffic flow.
2. Artificial intelligent (AI) models. In the later studies, it has been proven that traffic flow is nonlinear and nonstationary [4,5,31]. Plenty of researchers developed AI models to capture the nonlinearity of traffic flow. These models mainly contain *k*-nearest neighbor (KNN) [27,28,33],

support vector regression (SVR) [5,12,29,41], artificial neural network (ANN) [7,30,31,33], wavelet neural network (WNN) [7,37], and extreme learning machine (ELM) [23,40]. Due to the ability of nonlinear modelling, when a data scale is larger, the AI models usually perform better than the statistical models. Besides that, the deep learning models, which are improved from ANN and have more complex neural network structures, have been introduced and developed, to realize a more accurate prediction in recent years. The famous deep learning models include stacked auto encoders (SAE) [4], long short-term memory neural network (LSTM) [9,34,45], deep belief network (DBN) [16,42], and convolutional neural network (CNN) [32]. LSTM, especially, have shown superiority in traffic flow forecasting task due to the ability of temporal modelling [9,34,45]. Because of the universal approximation capability, the deep learning models can approximate any nonlinear function, and have shown outstanding performance for multi-source data. Nevertheless, owing to high computational complexity, these models will consume a large amount of time during training.

3. Hybrid models. Any method combining two or more models can be treated as a hybrid model. Hybrid models have flexible frameworks and can integrate the merits of different methods. In addition, both theoretical and empirical findings have indicated that the hybridization of different models is an effective way of improving prediction accuracy. Thus, hybrid models have been paid more attention than the aforementioned two kinds. Zeng et al. [35] and Tan et al. [36] both proposed combining ARIMA with ANN, to capture both the linearity and nonlinearity of traffic flow. In order to mine the spatial-temporal features of traffic flow, Luo et al. [6] proposed the hybrid model of KNN and LSTM, and Li et al. [38] proposed the hybrid model of ARIMA and SVR. To enhance the prediction performance of models, a genetic algorithm (GA) was employed by Hong et al. [41] and Zhang et al. [42], to optimize the hyperparameters of SVR and DBN. Feng et al. [43] employed particle swarm optimization (PSO) to optimize the hyperparameters of SVR.

As a novel interest of hybrid models, the methods combining time series decomposition approaches have received great attention. These hybrid models are based on the principle of "decomposition first and forecasting last": the decomposition approach is utilized to extract several components, and then these components are predicted respectively. The decomposition approaches have been successfully applied in the analysis and prediction of traffic flow include Fourier transform (FT) [19], singular spectrum analysis (SSA) [21,23], empirical mode decomposition (EMD) [3,10,22], and wavelet decomposition (WD) [25]. Luo et al. [19] employed FT to decompose the original traffic flow into trend and residual series. Then, they predicted the trend series by extreme extrapolation and predicted residual series by SVR. Guo et al. [21] implemented SSA to disaggregate traffic volume into smoothed series and residue, then, they forecasted the smoothed series by KNN and predicted residue by historical average. Shang et al. [23] developed SSA to disaggregate the traffic volume series into several components and predicted these components by ELM. Chen et al. [3], Wang et al. [10], and Chen et al. [22] all utilized EMD to decompose the original traffic flow into several intrinsic mode functions (IMFs), then, they predicted each IMF by ANN [3], ARIMA [10], and SVR [22], respectively. Zhang et al. developed WD to disaggregated the traffic spend series into and one low-frequency sequence and three high-frequency sequences. They proposed a graph convolutional recurrent neural network (GCRNN) to forecast the low-frequency component and ARIMA to forecast the high-frequency components.

In addition, many researchers [13,46–48] have suggested that traffic flow have the instincts of trend and periodic characteristics, and consideration for these characteristics during modelling can improve prediction accuracy. For those mentioned decomposition approaches, they are designed to analyze digital signals initially, and the decomposed results are a set of sub-time series with different frequency and amplitude. These approaches fail to extract the trend and periodic characteristics of traffic flow.

2.2. STL and the Proposed PTD Approach

As a time series decomposition approach, seasonal-trend decomposition procedure based on LOESS (STL) [49] can disaggregate the original data into trend, seasonal, and remainder components. STL is firstly proposed by Cleveland [49] to decompose long-term monthly CO_2 concentrations sequences with seasonal characteristics. Because of a strong robustness to outliers, STL is extensively employed to decompose time series with significant seasonality [50–54]. In addition, some studies have found that combining STL with prediction models can promote the accuracy of times series forecasting tasks. For example, Xiong et al. [53] utilized STL to decompose the seasonal agricultural commodity price, and employed extreme learning machines (ELM) to predict every component. In a similar way, in Qin et al.'s [54] study, they combined STL with echo state network (ESN), to improve the performance of monthly passenger flow forecasting. However, STL has two shortcomings which make it fail in the analysis of real-time traffic flow. One is that the decomposed seasonal component has a few gaps in different periods. In other words, the seasonal component is not a strict period function. The other is that STL is specially designed for macroscopic time series. It is a static procedure only for historical data, and it is unable to deal with out-of-sample data. Specifically, the STL is applicable to complete and static time series without updating, whereas real-time traffic flow is dynamic, and the new data comes as time goes on.

In this study, by overcoming these two shortcomings of STL, the periodic-trend decomposition (PTD) approach is proposed and formulated in Section 3.1. For the first shortcoming, PTD adjusts the seasonality into periodicity and the periodicity is a strict period function. For the second one, the procedure of out-of-sample decomposition is formulated additionally, to deal with out-of-sample data. Thus, with full consideration of the dynamicity of real-time traffic flow, PTD can decompose the original data into trend, periodicity, and remainder.

2.3. Contributions

In this paper, we propose a novel method combining PTD to decompose and predict real-time traffic flow. In order to demonstrate the universal applicability of the PTD approach, both statistical and AI models (i.e., ARIMA, SVR, ANN, LSTM) are combined with PTD to establish hybrid models (i.e., PTD-ARIMA, PTD-SVR, PTD-ANN and PTD-LSTM). These models are tested and evaluated based on real-world traffic volume data collected from Mn/DOT. The main contributions of this study are summarized as follows:

1. A novel PTD approach is specially formulated for real-time traffic flow composition. We develop the PTD approach to extract trend and periodic characteristics of traffic flow. Fully considering the dynamicity of traffic flow in the real world, PTD is improved from STL to decompose traffic flow data. PTD can decompose the original traffic flow three additive components, including trend, periodicity, and remainder. The components can reveal the inner characteristics of traffic flow.
2. A novel method combining PTD is developed to predict traffic flow more accurately. After completing decomposition, the periodicity is predicted by cycling, and the trend and remainder are respectively predicted by modelling. Then, the three predicted results are summed as final outcomes.
3. Multi-step prediction is implemented, to provide more traffic flow information of the future. Traditional single-step prediction is unable to provide enough information for further plans and decisions of ITS. Thus, multi-step is necessary. Based on an iterated strategy, a multi-step prediction approach is developed, to extend the proposed hybrid method.

3. Methodology

In this section, the method combining PTD to forecast traffic flow is elaborated. First, the PTD method is formulated. Then, four prevalent prediction models are presented. Finally, the hybrid prediction method is illustrated.

3.1. Periodic-Trend Decomposition (PTD) Approach

3.1.1. Locally Weighted Regression (LOWESS)

Locally weighted regression (LOWESS, also aliased as LOESS) was first proposed by Cleveland [49] for smoothing scatterplots in STL. In this study, LOWESS is also employed in PTD to smooth curves, to extract trend and periodic components. Shown in Figure 1, LOWESS is a nonparametric regression model. The target point is evaluated by k nearest neighbors with distance weights.

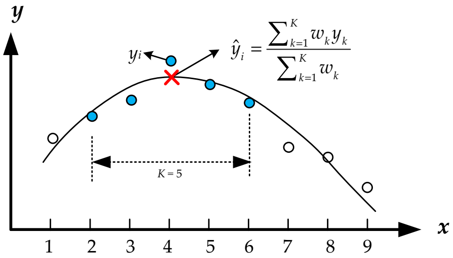

y
y_i
$\hat{y}_i = \frac{\sum_{k=1}^{K} w_k y_k}{\sum_{k=1}^{K} w_k}$
K = 5
x
1
2
3
4
5
6
7
8
9

Figure 1. A simple case of LOWESS.

Suppose a group of independent variables x_i and dependent variables y_i ($i = 1, 2, \ldots$). The estimated value $\hat{y}_i$ is evaluated by K nearest neighbors of y_i based on the equation:

$$\hat{y}_i = \frac{\sum_{k=1}^{K} w_k y_k}{\sum_{k=1}^{K} w_k} \tag{1}$$

where w_k is the weight of near neighbor w_k and computed as:

$$w_k = D\left(\frac{|x_i - x_k|}{\lambda}\right) \tag{2}$$

In this equation, D(x) denotes *Epanechnikov* kernel function, expressed as:

$$D(t) = \begin{cases} \frac{3}{4}\left(1 - t^2\right), & \text{while } |t| < 1 \\ 0, & \text{or therwise} \end{cases} \tag{3}$$

where λ denotes the kernel width of *Epanechnikov* and represents the distance between x_i and its Kth nearest neighbor $x_{[K]}$, expressed as:

$$\lambda = \left|x_i - x_{[K]}\right| \tag{4}$$

3.1.2. Decomposition for In-Sample

The in-sample data are the historical traffic flow with several periods. Given an in-sample traffic flow $Y(t)$, PTD method can disaggregate it into three additive components, including trend $T(t)$, periodicity $P(t)$ and remainder $R(t)$:

$$Y(t) = T(t) + P(t) + R(t) \tag{5}$$

For in-sample decomposition, PTD is an iterative method which is similar to STL. The flow chart of PTD is shown in Figure 2. During each iteration process, periodicity and trend are respectively updated by periodic smoothing and trend smoothing. Suppose $Y(t)$ denotes the original traffic flow, C and M denote the length of one period and the number of periods, respectively. Thus, the number of samples $Y(t)$ is MC. In the initialization, the trend is set as $T(t) = 0$, $t = 1, 2, \ldots, MC$. The loop process is composed of six steps, as follows:

Step 1: Detrending. Trend $T(t)$ is removed from original data $Y(t)$, then de-trended series $\widetilde{T}(t)$ is obtained:

$$\widetilde{T}(t) = Y(t) - T(t) \tag{6}$$

Step 2: Cycle-subseries smoothing. The details of this step are displayed in Figure 2b. Each cycle-subseries $S_c(t)$ of $\widetilde{T}(t)$ is smoothed by LOWESS with K_1, and extended one period forward and backwards. Specifically, for independent variables $X = [1, 2, 3, 4, \ldots, M]^{\mathrm{T}}$ and dependent variables $Y = S_c(t) = [Y(c), Y(C+c), Y(2C+c), \ldots, Y((M-1)C+c)]^{\mathrm{T}}$, the estimated values $\hat{Y}_{\mathrm{c}}$ are estimated by LOWESS with K_1 at $X = [0, 1, 2, 3, 4, \ldots, M, M+1]^{\mathrm{T}}$, then all the $\hat{Y}_c$, $c = 1, 2, \ldots, C$ are arranged in chronological order. Finally, they are recombined and generate temporal time series $\hat{S}(t)$, $t = -C + 1, -C + 2, \ldots, MC + C$. The cycle-subseries $S_c(t)$ represents a series composed of elements at the same position of each period, as follows:

$$S_{\mathrm{c}}(t) = \{Y(c), Y(C{+}c), Y(2C+c), \ldots, Y((M-1)C+c)\}, c = 1, 2, \ldots, C \tag{7}$$

Step 3: Low-pass filtering of the cycle-subseries smoothing. The temporal time series $\hat{S}(t)$, $t = -C + 1, -C + 2, \ldots, MC + C$ are smoothed by a low-pass filter to obtain the series $L(t)$, $t = 1, 2, \ldots, MC$. The low-pass filter consists of a moving average of length C, followed by another moving average of length C, followed by another moving average of 3, and followed by LOWESS smooth with K_2.

Step 4: Detrending of Smoothed Cycle-subseries. The preliminary periodicity is calculated as:

$$F(t) = \hat{S}(t) - L(t), t = 1, 2, \ldots, MC \tag{8}$$

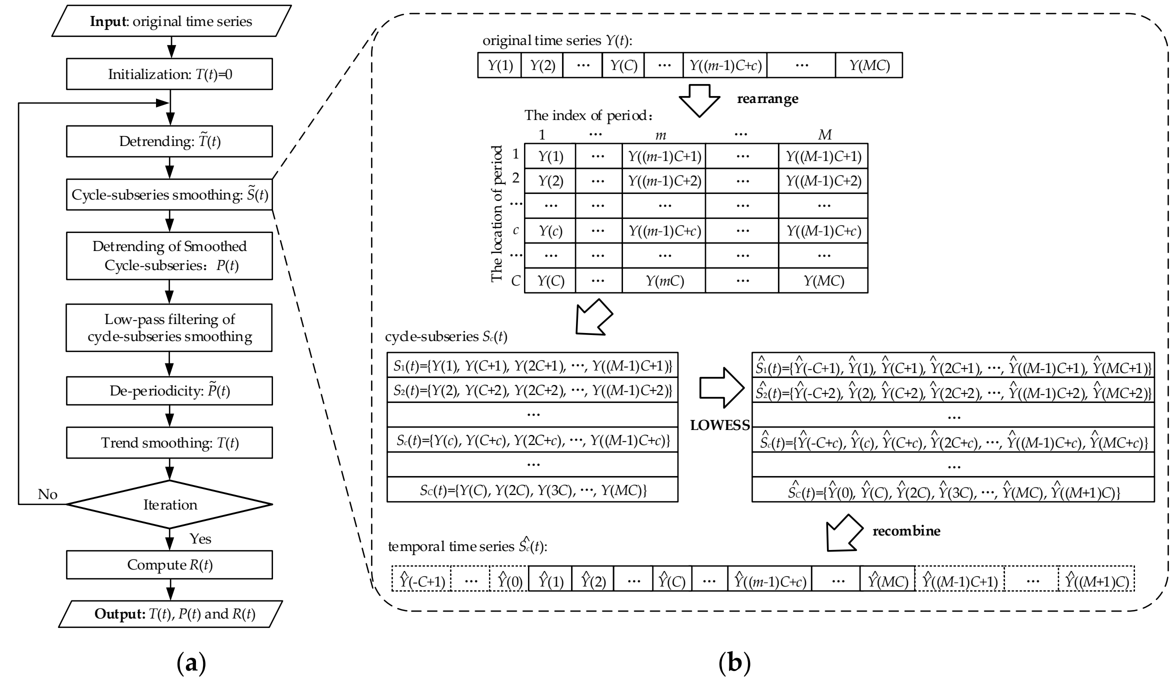

Input: original time series
Initialization: $T(t)$=0
Detrending: $\widetilde{T}(t)$
Cycle-subseries smoothing: $\widetilde{S}(t)$
Detrending of Smoothed Cycle-subseries: $P(t)$
Low-pass filtering of cycle-subseries smoothing
De-periodicity: $\widetilde{P}(t)$
Trend smoothing: $T(t)$
Iteration
No
Yes
Compute $R(t)$
Output: $T(t)$, $P(t)$ and $R(t)$
original time series $Y(t)$:
$Y(1)$ | $Y(2)$ | ··· | $Y(C)$ | ··· | $Y((m\text{-}1)C\text{+}c)$ | ··· | $Y(MC)$
rearrange
The index of period:
1 ··· m ··· M
The location of period
1 | $Y(1)$ | ··· | $Y((m\text{-}1)C\text{+}1)$ | ··· | $Y((M\text{-}1)C\text{+}1)$
2 | $Y(2)$ | ··· | $Y((m\text{-}1)C\text{+}2)$ | ··· | $Y((M\text{-}1)C\text{+}2)$
··· | ··· | ··· | ··· | ··· | ···
c | $Y(c)$ | ··· | $Y((m\text{-}1)C\text{+}c)$ | ··· | $Y((M\text{-}1)C\text{+}c)$
··· | ··· | ··· | ··· | ··· | ···
C | $Y(C)$ | ··· | $Y(mC)$ | ··· | $Y(MC)$
cycle-subseries $S_c(t)$
$S_1(t)$={$Y(1)$, $Y(C\text{+}1)$, $Y(2C\text{+}1)$, ···, $Y((M\text{-}1)C\text{+}1)$}
$S_2(t)$={$Y(2)$, $Y(C\text{+}2)$, $Y(2C\text{+}2)$, ···, $Y((M\text{-}1)C\text{+}2)$}
···
$S_c(t)$={$Y(c)$, $Y(C\text{+}c)$, $Y(2C\text{+}c)$, ···, $Y((M\text{-}1)C\text{+}c)$}
···
$S_C(t)$={$Y(C)$, $Y(2C)$, $Y(3C)$, ···, $Y(MC)$}
LOWESS
$\hat{S}_1(t)$={$\hat{Y}(\text{-}C\text{+}1)$, $\hat{Y}(1)$, $\hat{Y}(C\text{+}1)$, $\hat{Y}(2C\text{+}1)$, ···, $\hat{Y}((M\text{-}1)C\text{+}1)$, $\hat{Y}(MC\text{+}1)$}
$\hat{S}_2(t)$={$\hat{Y}(\text{-}C\text{+}2)$, $\hat{Y}(2)$, $\hat{Y}(C\text{+}2)$, $\hat{Y}(2C\text{+}2)$, ···, $\hat{Y}((M\text{-}1)C\text{+}2)$, $\hat{Y}(MC\text{+}2)$}
···
$\hat{S}_c(t)$={$\hat{Y}(\text{-}C\text{+}c)$, $\hat{Y}(c)$, $\hat{Y}(C\text{+}c)$, $\hat{Y}(2C\text{+}c)$, ···, $\hat{Y}((M\text{-}1)C\text{+}c)$, $\hat{Y}(MC\text{+}c)$}
···
$\hat{S}_C(t)$={$\hat{Y}(0)$, $\hat{Y}(C)$, $\hat{Y}(2C)$, $\hat{Y}(3C)$, ···, $\hat{Y}(MC)$, $\hat{Y}((M\text{+}1)C)$}
recombine
temporal time series $\hat{S}_c(t)$:
$\hat{Y}(\text{-}C\text{+}1)$ | ··· | $\hat{Y}(0)$ | $\hat{Y}(1)$ | $\hat{Y}(2)$ | ··· | $\hat{Y}(C)$ | ··· | $\hat{Y}((m\text{-}1)C\text{+}c)$ | ··· | $\hat{Y}(MC)$ | $\hat{Y}((M\text{-}1)C\text{+}1)$ | ··· | $\hat{Y}((M\text{+}1)C)$
(**a**)
(**b**)

Figure 2. The flow chart of PTD for in-sample decomposition: (**a**) PTD; (**b**) Cycle-subseries smoothing.

In the original STL method [49], $F(t)$ is regarded as a seasonal component, which shows obvious periodicity. However, the values at the same time in each period have a few differences. Thus, it is unable to directly forecast future values by cycling. In this study, this component is normalized with the further process as the following: firstly, one period length time series of periodicity $Q(t)$, $t = 1, 2, \ldots, C$ is computed as:

$$Q(t) = \frac{1}{M}\sum_{m=0}^{M-1} F(mC+t),\ t = 1,2,..,C \tag{9}$$

Then, the $Q(t)$ is extended to M periods to obtain the whole periodicity $P(t)$, t = 1, 2,..., MC.

Step 5: De-periodicity. The original series $Y(t)$ subtracts the periodicity $P(t)$ to obtain a periodically adjusted series $\widetilde{P}(t)$:

$$\widetilde{P}(t) = Y(t) - P(t) \tag{10}$$

Step 6: Trend smoothing. the $\widetilde{P}(t)$ is smoothed by LOWESS with K_3 to obtain the trend $T(t)$ = 0, t = 1, 2,..., MC.

After completing iterations, the remainder $R(t)$ = 0, t = 1, 2,..., MC is calculated as:

$$R(t) = Y(t) - T(t) - P(t) \tag{11}$$

Following the above steps, the original times series $Y(t)$ is decomposed into three additive components: trend $Y(t)$, periodicity $P(t)$ and remainder $R(t)$, as Equation (5) expressed.

3.1.3. Decomposition for Out-Of-Sample

Owing to the dynamicity of real-time traffic flow, new data will come as time goes on. Hence, traffic flow data is also updated simultaneously. The new coming data is regarded as out-of-sample. The proposed PTD also provides the method to decompose out-of-sample data in a timely manner, which can satisfy the demand of real-time traffic flow forecasting.

For a new coming data z_t at time t, it can also be disaggregated into three additive components, including trend t_t, periodicity p_t and remainder r_t as the following four steps:

Step 1: Calculating periodicity. The periodicity p_t is calculated according to its position of the period in $Q(t)$ (obtained from Equation (9)).

Step 2: De-periodicity. The original data z_t subtract the periodicity p_t to obtain a periodically adjusted datum $\widetilde{z}_t$:

$$\widetilde{z}_t = z_t - p_t \tag{12}$$

Step 3: Calculating trend component. The periodically adjusted datum $\widetilde{z}_t$ is appended to the end of the periodically adjusted series $\widetilde{P}(t)$(obtained from Equation (10)). The next point of the new series is evaluated by LOWESS with K_4, to obtain trend t_t.

Step 4: Calculating remainder. The remainder r_t is computed as:

$$r_t = z_t - t_t - p_t \tag{13}$$

Following the above four steps, the out-of-sample of traffic flow data is decomposed into trend t_t, periodicity p_t and remainder r_t.

3.2. Prediction Models for the Decomposed Components

After completing decomposition, the prediction models are implemented to forecast trend and residual. In order to test the universal applicability of the proposed hybrid method, four well-established prediction models, including both statistical and AI models, are separately combined with PTD. The four prediction models are ARIMA, SVR, ANN, and LSTM. Periodicity is predicted by cycling in the place of modelling since it is a strict period function.

3.2.1. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a statistical time series model proposed by Bartholomew et al. [55], based on the assumption of stationarity and linearity of time series. Generally, ARIMA is composed of three parts: autoregressive (AR), integration (I) and moving average (MA). The performance of ARIMA is affected by three parameters: autoregressive order p, difference order d and moving average order q. The ARIMA (p,d,q) model can be expressed as follows:

$$\nabla^d \hat{x}_t = \varphi_0 + \varphi_1 x_{t-1} + \cdots + \varphi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q} \tag{14}$$

where $\nabla = (1 - \mathrm{B})$ refers to the backward shift operator for $B(x) = x_{t-1}$, $\hat{x}_t$ is the predicted value, ε_t is the residual component distributed as Gaussian distribution with zero mean and σ^2 variance.

The difference order d is relevant to the difference method which can convert a nonstationary time series to a stationary time series. In this study, the augmented Dickey–Fuller (ADF) test (unit root test) is utilized, to diagnose the stationarity of traffic flow and determined the parameter d. In addition, the parameters p and q are selected based on the minimum Bayesian information criterion (BIC) value, as follows:

$$\mathrm{BIC} = k \ln(n) - 2\ln(L) \tag{15}$$

where k is the number of model parameters; n is the number of samples; L is the maximum likelihood value.

3.2.2. Support Vector Regression (SVR)

Initially, a support vector machine (SVM) was proposed by Boser et al. [56] to solve classification problems, and Cortes and Vapnik [57] extended SVM for regression problems and renamed it as support vector regression (SVR). SVR is based on the structured risk minimization principle, which minimizes the upper bound of the generalization error by using subset data points (i.e., support vectors). For time series modelling, the output of SVR is expressed as:

$$\hat{x}_t = f(\boldsymbol{x}_{t-1}) = \boldsymbol{w}^{\mathrm{T}} \varphi(\boldsymbol{x}_{t-1}) + b \tag{16}$$

where $\hat{x}_t$ is the predicted value; $\boldsymbol{x}_{t-1} = [x_{t-1}, x_{t-2}, \ldots, x_{t-n}]^{\mathrm{T}}$ is input vector with n time lag; $\boldsymbol{w}$ and b are weights and bias, respectively; $\varphi(\boldsymbol{x})$ represents the nonlinear mapping function and maps input space $\boldsymbol{x}$ into a high-dimensional feature. In this study, ε-SVR is employed and slack variables ξ and ξ^* are introduced. The slack variables measure the errors which occurred by values outside the boundaries, and errors are determined by the ε-tube. The SVR is an optimization problem with objective function as Equation (17) and constraints as Equation (18).

$$\min_{\boldsymbol{w},\mathrm{b},\boldsymbol{\xi},\boldsymbol{\xi}^*} \frac{1}{2}\|\boldsymbol{w}\|^2 + H\sum_{i=1}^{m} (\xi_i + \xi_i^*) \tag{17}$$

$$\text{subject to}: \begin{cases} \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i) + b - y_i \leq \varepsilon + \xi_i \\ y_i - \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x}_i) - b \leq \varepsilon + \xi_{\mathrm{i}}^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{18}$$

where H ($H > 0$) represents the regularization factor; m represents the number of training samples.

If this optimization problem can be transformed into a dual problem, then the SVR function can be expressed as follows:

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*)\kappa(\boldsymbol{x}_i, \boldsymbol{x}) \tag{19}$$

where α_i and ${\alpha_i}^*$ are the Lagrange multipliers; $\kappa(\boldsymbol{x_i},\boldsymbol{x})$ is the kernel function that equals the inner product of $\varphi(\boldsymbol{x}_i)$ and $\varphi(\boldsymbol{x})$. The radial basis function (RBF) is widely used as kernel function in previous literature [5,12,19,41], and it is also adopted in this study. BRF is expressed as:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}\|^2\right) \tag{20}$$

where γ is the coefficient of RBF.

3.2.3. Artificial Neural Network (ANN)

ANN is abstracted from neurons in the human brain. ANN has the universal approximation ability, which indicates that ANN can estimate any nonlinear continuous function up to any desired degree of accuracy. Hence, this model has a good ability to model nonlinear data. Because of this advantage, ANN has been widely introduced and developed for prediction tasks, as well as traffic flow forecasting. A typical ANN model consists of three parts: an input layer, a hidden layer, and an output layer. The structure of ANN is flexible and it depends on the number of neurons in each layer. In this paper, a one-hidden-layer and single-output neural network is employed. The relationship between the output x_t and the inputs $x_{t-1}, x_{t-2}, \ldots, x_{t-n}$ is expressed as follows:

$$x_t = \sigma\left(b_0 + \sum_{i=1}^{u} b_i \sigma\left(a_{i0} + \sum_{j=1}^{n} a_{ij} x_{t-j}\right)\right) + \varepsilon_t \tag{21}$$

where a_{ij}, $j = 1, 2, \ldots, n$, represent connection weights from the input layer to hidden neuron ith; a_{i0} represents the biases in hidden neuron ith; b_i, $i = 1, 2, \ldots, u$, represents connection weights from hidden layer to output neuron; b_0 represents the bias in output neuron; n and u are the number of input neurons and hidden neurons, respectively; and ε_t is the error term. The $\sigma(x)$ is the sigmoid activation function and is formulated as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{22}$$

Therefore, the ANN model in fact performs a nonlinear functional mapping from the historical values $x_{t-1}, x_{t-2}, \ldots, x_{t-n}$ to the future value x_t:

$$x_t = f_{\text{ANN}}(x_{t-1}, x_{t-2}, \ldots, x_{t-n}) + \varepsilon_t \tag{23}$$

3.2.4. Long Short-Term Memory Neural Network (LSTM)

LSTM is proposed by Hochreiter and Schmidhuber [58] to model for time series data. LSTM is different to traditional ANN in terms of the hidden layers. The hidden neurons of LSTM are designed specially to capture the temporal characteristics. It is realized through one memory cell and 3 gates of input gate, forget gate and output gate. At time t, input gate processes new coming data and the state of the memory cell at time $t-1$. Forget date decides when to forget the information of memory cell and selects the optimal lag for the input time series. Output gate controls the output information transmitting to the next neural layer. Suppose x_t denotes the input time series at t time; i_t, f_t and o_t denote the state of input gate, forget gate and output gate, respectively; c_{t-1} denotes the state of the memory cell at $t-1$ time. The equations to update the three gates are expressed as follows:

$$i_t = \sigma(w_i x_t + u_i c_{t-1} + b_i) \tag{24}$$

$$f_t = \sigma(w_f x_t + u_f c_{t-1} + b_f) \tag{25}$$

$$o_t = \sigma(w_o x_t + u_o c_{t-1} + b_o) \tag{26}$$

At time t, the state of memory cell c_t is controlled by input gate i_t, forget gate f_t and the last time state of memory cell c_{t-1}:

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(w_c x_t + u_c c_{t-1} + b_c) \tag{27}$$

The information of LSTM unit output to next layer units is controlled by output gate o_t and the state of memory cell c_t:

$$h_t = o_t \circ \tanh(c_t) \tag{28}$$

where h_t represents the final output of hidden neuron; operator "$\circ$" is Hadamard product; w_i, w_f, w_o, w_c, u_i, u_f, u_o and u_c are connection weights; b_i, b_f, b_o and b_c are activation biases; $\sigma(x)$ and $\tanh(x)$ are activation functions; $\sigma(x)$ is expressed as Equation (22) and $\tanh(x)$ is expressed as: $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

3.3. Multi-Step Prediction

One-step prediction is extensively studied in the previous study [6,7,12,19,32,38], whereas, it cannot always satisfy the applications in ITS. In this study, a multi-step prediction strategy is well-designed for the proposed hybrid method.

Firstly, predict periodicity. Periodicity is predicted by cycling. Since the periodicity $P(t)$ extracted from PTD is a strictly function with C period length (see Equation (9)), the predicted value at time t is the same to historical value at time t−C:

$$\hat{p}_t = p_{t-C}; \quad \hat{p}_{t+1} = p_{t-C+1}; \quad \ldots \tag{29}$$

Secondly, predict trend and remainder respectively. The iterated strategy, which is widely used in multi-step time series forecasting, is adopted to the multi-step prediction of trend and remainder. Based on an already constructed model with one-step prediction, the iterated strategy uses the predicted value as an input for the same model to forecast the next time value (see Equation (30)). This process is iterated up to the maximum prediction horizon. The iterated strategy has two outstanding advantages: one is that only one model must be established. The other is that the prediction steps are unlimited.

$$\begin{array}{rcl} \hat{t}_t = f_T(t_{t-1}, t_{t-2}, \ldots, t_{t-n}) & ; & \hat{r}_t = f_R(r_{t-1}, r_{t-2}, \ldots, r_{t-n}) \\ \hat{t}_{t+1} = f_T(\hat{t}_t, t_{t-1}, \ldots, t_{t-n+1}) & ; & \hat{r}_{t+1} = f_R(\hat{r}_t, r_{t-1}, \ldots, r_{t-n+1}) \\ \cdots & ; & \cdots \end{array} \tag{30}$$

where t_t and r_t respectively represent the true values of trend and remainder at time t; $\hat{t}_t$ and $\hat{r}_t$ respectively represent the predicted values of trend and remainder at time t; f_T and f_R represent the already constructed model for trend and remainder, respectively.

Lastly, the predicted values of the three components are summed as final outcomes $\hat{y}_t$:

$$\begin{array}{c} \hat{y}_t = \hat{t}_t + \hat{p}_t + \hat{r}_t \\ \hat{y}_{t+1} = \hat{t}_{t+1} + \hat{p}_{t+1} + \hat{r}_{t+1} \\ \cdots \end{array} \tag{31}$$

3.4. Hybrid Prediction Models

Based on the principle of "decomposition first and forecasting last", the procedure of the proposed hybrid method for traffic flow forecasting is displayed in Figure 3; it contains three steps, as listed:

Step 1: In-sample decomposition and models training. Based on the in-sample decomposition of PTD, the in-sample data is decomposed into three components: trend, periodicity, and remainder. Then, each component is used to train the models (described in Section 3.2) separately.

Step 2: Out-of-sample decomposition and multi-step prediction. Based on the out-of-sample decomposition of PTD, the out-of-sample data is also decomposed into trend, periodicity, and remainder. Then, each component of out-of-sample decomposition is input into the corresponding trained model. The predicted results are obtained from the output of the models.

Step 3: Integration for the final outcomes. The predicted results of the three components are summed as the final outcomes.

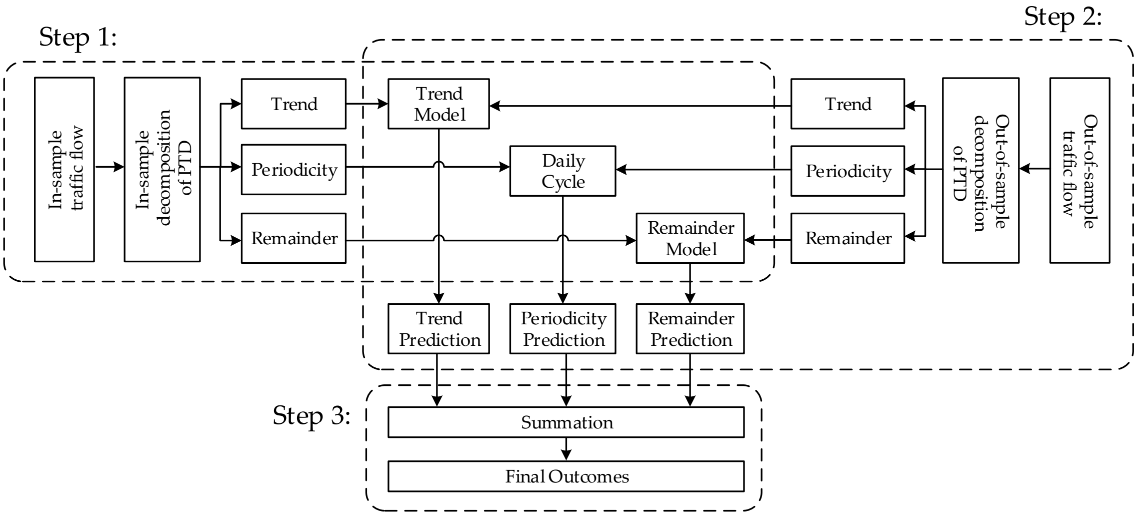


Figure 3. The procedure of the proposed hybrid prediction method.

4. Experiments

4.1. Data Description

In this study, the traffic flow data are available from the Minnesota Department of Transportation (Mn/DOT) [59]. The Mn/DOT provides traffic volume and occupancy data with a 30-second interval of freeways in Minnesota, USA. In our experiments, only traffic volume data are used, and the data are collected from three detection stations (i.e., S188, S195 and 818) at the interchange of I-494 and TH100 (see Figure 4). As shown in Figure 4, S188 and S195 are located in two opposite directions of I-94 freeway and 818 is located at an exit ramp. The three cases with different patterns are used to test the adaptability of the hybrid models. We selected four weeks of data, from 29 April to 24 May in 2019. Since the patterns of traffic volume on weekdays and weekends are different, the data on weekends is discarded. The original traffic volume data is aggregated into a 5-minute interval referred to in previous studies [4,5,14,21,60]. The data of the first two weeks are used as the training dataset (i.e., from 29 April to 10 May, 2880 samples totally) to train the models. The next week is used as the validation dataset (i.e., from 12 to 16 in May, 1440 samples totally) to determine the hyper-parameters of the models. The last week (i.e., from 20 to 24 in May, 1440 samples totally) is used as the testing dataset to evaluate the prediction results.

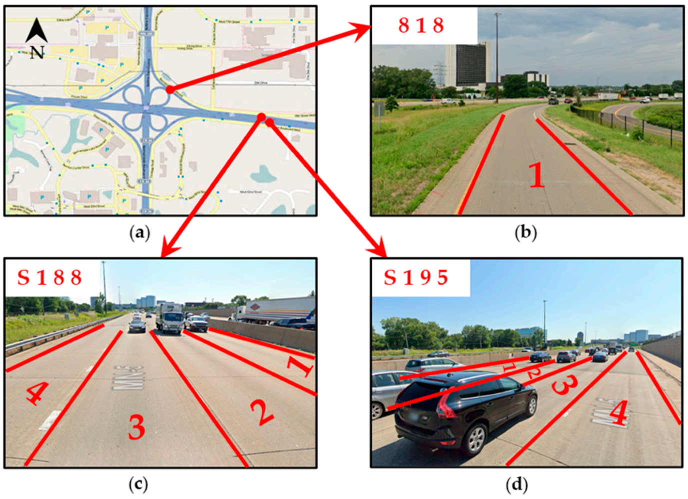


Figure 4. The location of selected stations: (**a**) Interchange of I-494 and TH100; (**b**) 818; (**c**) S188; (**d**) S195.

The training dataset of collected traffic volume is shown in Figure 5. It can be seen that the three stations have different distributions, which is to evaluate the adaptability of proposed hybrid models. The distribution features of traffic flow are agreed to their locations: S188 and S195 are located at the four-lane highway with large traffic volume in the daytime, and the peak-volume of these two stations are both over 600 vehicles per 5-minute; 818 is located at the ramp with only one lane and the traffic volume is much smaller, and the peak volume of it is about 180 vehicles per 5-minute. Furthermore, S188 has two obvious peaks in the morning and afternoon, and S195 only has one peak in the morning. Moreover, 818 only has one peak in the afternoon. Despite different distributions and peak patterns, all the three stations have obvious one-day periodicity.

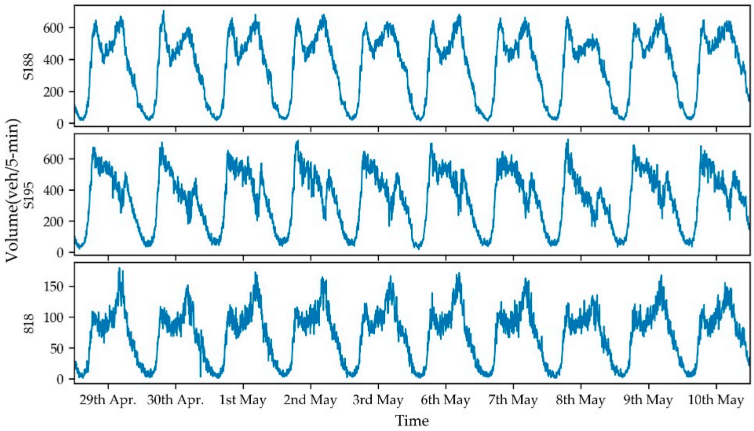


Figure 5. The traffic volume of the training dataset, i.e., from 29 April to 10 May 2019.

4.2. Design of Experiments

To demonstrate the contribution of the proposed PTD approach, four hybrid models combining PTD (i.e., PTD-ARIMA, PTD-SVR, PTD-ANN and PTD-LSTM) and four individual models (i.e., ARIMA, SVR, ANN and LSTM) are employed to experiment and comparatively analyze by using the prepared data.

In the PTD procedure, the training datasets are regarded as in-sample, and the validation datasets and testing datasets are regarded as out-of-sample, to simulate real-time traffic flow. The period length C is 288 (i.e., 24 (hour) * 60 (minute)/5 (minute) = 288), and the period number M is 10 (i.e., workdays of two weeks). According to the studies [49,53,54], the nearest neighbor number K of LOWESS should be set to cover one-period length. In our study, for in-sample decomposition, the LOWESS smoothen original series contains both past and future. Consequently, K_1, K_2, and K_3 (see Step 2, Step 3, and Step 6 in Section 3.1.2.) are all set as 144. Meanwhile, for out-of-sample decomposition, the LOWESS smoothen original times series containing only past and K_4 (see Step 4 in Section 3.1.3.) is set as 288.

In the prediction stage, the Min-Max normalization technology (expressed as Equation (32)) is employed to scale the data into the range of [0, 1] before feeding into the model. This technology can accelerate learning and convergence during training model. The outputs of the model are rescaled by reversed Min-Max normalization (expressed as Equation (33)), to obtain the final predicted results.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{32}$$

$$x = x' \times (\max(x) - \min(x)) + \min(x) \tag{33}$$

It is worth noting that the hyper-parameters of prediction models can significantly affect their performance. Hence, the well-established grid-search and cross-validation method is implemented, to select appropriate hyper-parameters for each model. Specifically, all the models with different hyper-parameters are trained by the same training dataset, but only the model which has the best performance on validation dataset is selected. The models with determined hyper-parameters are evaluated by testing dataset. The critical hyper-parameters in each model are as follows:

- **ARIMA:** the difference order d of ARIMA is set based on ADF unit root test and the autoregressive order p and moving average order q are both selected from 0 to 24, based on minimum BIC value (see Equation (15)).
- **SVR:** The RBF kernel function coefficient γ, and the regularization factor H, the tube width ε are all selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$.
- **ANN:** According to previous research [30,31], a one-hidden-layer neural network is frequently utilized and also adopted in our experiment. The number of hidden neurons u is selected from 2 to 40 with step 2. The model is optimized by Adam algorithm with mean square error (MSE) loss function. The learning rate is set as 0.001, the batch-size is 256, and the epochs is 500.
- **LSTM:** A one-hidden-layer LSTM is utilized reference to existing studies [9,34,60]. The number of hidden neurons is also selected from 2 to 40 with step 2, and the other hyper-parameters are the same as ANN.

In addition, the input step is set as 12 and output step is set as 1. The prediction horizons are set as 6. The final determined hyper-parameters are shown in Table 2. The periodicity is not in this table, because this component can be predicted by cycling.

Table 2. The determined hyper-parameters of testing models.

Model	Component	Hyper-Parameters	Detection Station		
			S188	**S195**	**818**
ARIMA	-	p, d, q	14, 0, 1	2, 0, 4	6, 0, 11
SVR	-	γ, H, ε	$1^{-1}, 10^4, 10^{-4}$	$1, 1, 10^{-2}$	$10^{-1}, 10, 10^{-2}$
ANN	-	u	16	16	18
LSTM	-	u	30	12	12
PTD-ARIMA	trend	p, d, q	1, 0, 14	2, 0, 1	18, 0, 16
	remainder	p, d, q	1, 0, 1	1, 0, 1	1, 0, 1
PTD-SVR	trend	γ, H, ε	$10, 10^{-1}, 10^{-4}$	$10^{-3}, 10^3, 10^{-4}$	$10^{-4}, 10^4, 10^{-4}$
	remainder	γ, H, ε	$10^{-1}, 10, 10^{-1}$	$10^{-3}, 10^4, 10^{-3}$	$10^{-2}, 10^2, 10^{-2}$
PTD-ANN	trend	u	30	16	30
	remainder	u	16	40	40
PTD-LSTM	trend	u	20	32	16
	remainder	u	10	34	4

4.3. Performance Measures

In our study, three widely used measures are utilized to evaluate the performance of models, including mean absolute error (MAE), mean absolute percentage error (MAPE) and mean square error (MSE). Denote true value and predicted value as y_t and $\hat{y}_i$, respectively; n is the number of testing data. The equations are expressed as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{34}$$

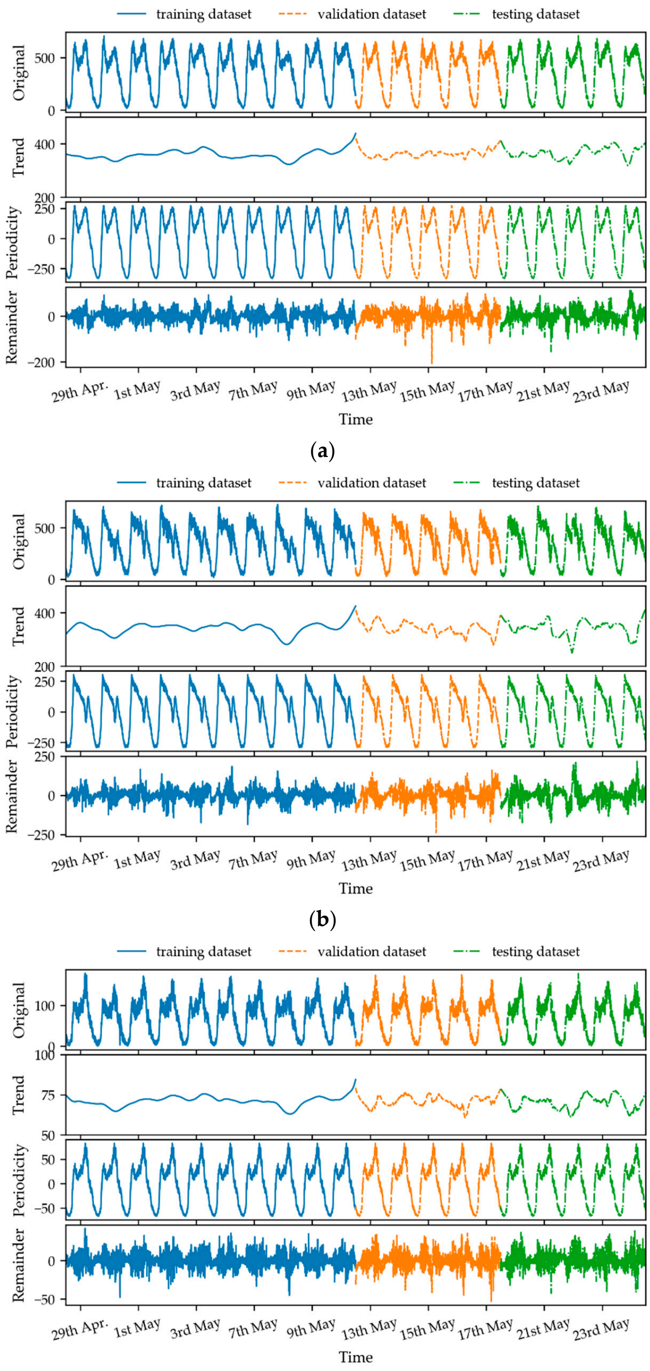
$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{35}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{36}$$

5. Results and Discussions

5.1. Analysis of Decomposition Results

The decomposition results are shown in Figure 6. It can be seen that the original traffic flow is decomposed into trend, periodic, and remainder components. For trend, this component shows the day-to-day characteristics, and it fluctuates moderately. Furthermore, trend extracted through in-sample decomposition (i.e., training dataset) is smoother than that through out-of-sample decomposition (i.e., validation dataset and testing dataset). The reason for this is that the trend is extracted by bilateral smoothness, containing both past and future during in-sample decomposition. Meanwhile, during out-of-sample decomposition, the trend is extracted through only past smoothness. The periodicity shows the within-day characteristics and it is a strict period function, which means the values at the same time on different days are identical. The peak features of traffic flow are also distinctly displayed in periodicity, which is consistent with the original data. As for the remainder, it fluctuates irregularly and can be regarded as noise.



(**a**)

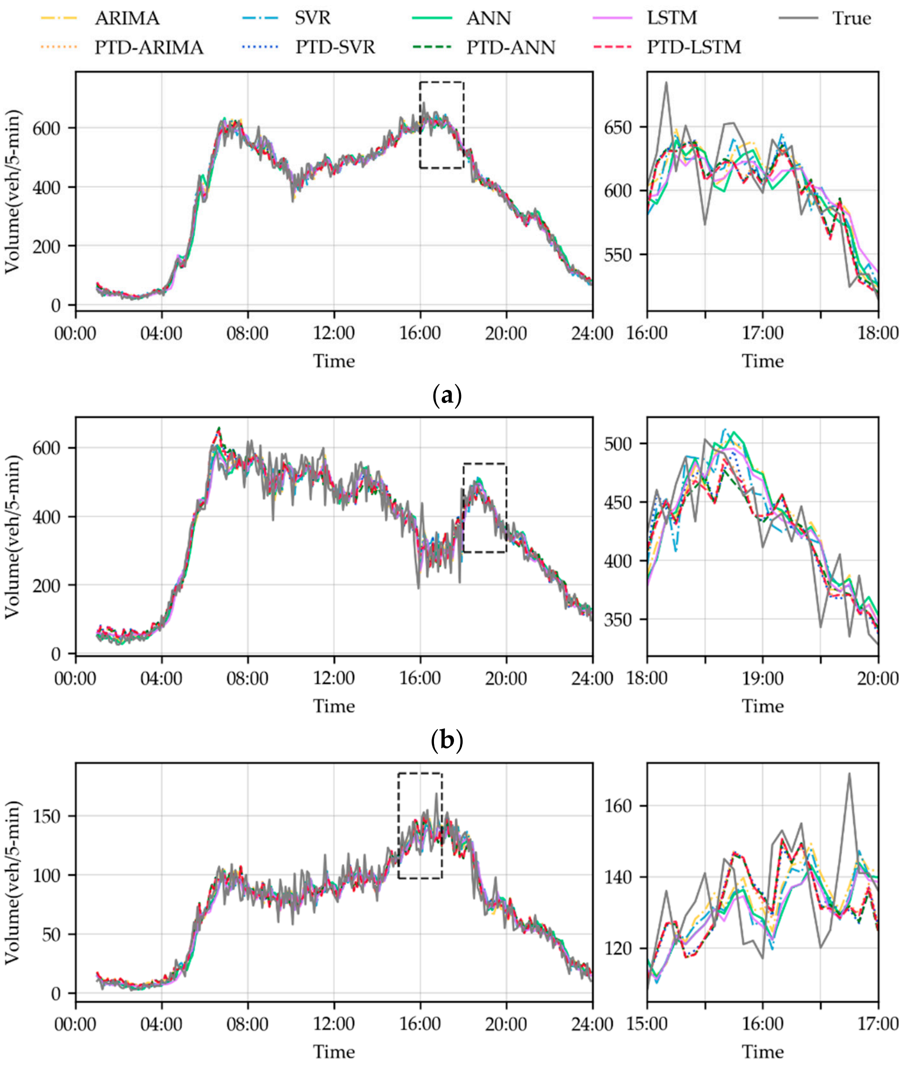
(**b**)

(**c**)

Figure 6. Decomposition Results: (**a**) S188; (**b**) S195; (**c**) 818.

5.2. Analysis of Prediction Results

Each component is predicted separately, and the final outcomes are obtained by summing the three predicted results. Taking the first day of the testing dataset (i.e., 20 May 2019) as an example, the true values and one-step predicted results are shown in Figure 7. The left subplots are the whole day predicted results, and the right subplots zoom in on two rush hours in the evening. It is apparent that the predicted results of hybrid models are closer to true values than those results of individual models, which indicates that the hybrid models perform better. In addition, the predicted results of the four hybrid models are almost the same, which suggests that the four models combining PTD have almost similar performances.



(**a**)

(**b**)

(**c**)

Figure 7. The prediction results of traffic volume. The left subplots are the whole day on 20 May 2019, and the right subplots are two rush hours in the evening: (**a**) S188; (**b**) S195; (**c**) 818.

Table 3 provides the average measures evaluated across all the six-step prediction horizons. Overall, all hybrid models perform better than individual models. Among hybrid models, it seems that there is no specific hybrid model that always performs best in all testing cases. However, after further investigation, it is found that the gaps among different hybrid models are slightly small. In particular, the differences of MAE among different hybrid models are less than 1. This finding suggests that combination with PTD is a universally applicable method to improve prediction accuracy effectively.

In addition, the improved items represent the improved performance from the hybrid model to the individual model. This item can also be regarded as the contribution of PTD approach to reduce the prediction errors. On average, the MAE, MAPE and MSE are respectively reduced by 23%, 25% and 39% on station S188, 11%, 11% and 19% on station S195, and 16%, 13% and 28% on station 818. Over all testing stations, the hybrid method reduces the MAE, MAPE, and MSE by 17%, 17%, and 29%. These results demonstrate that the hybrid models combining PTD are superior to the individual models.

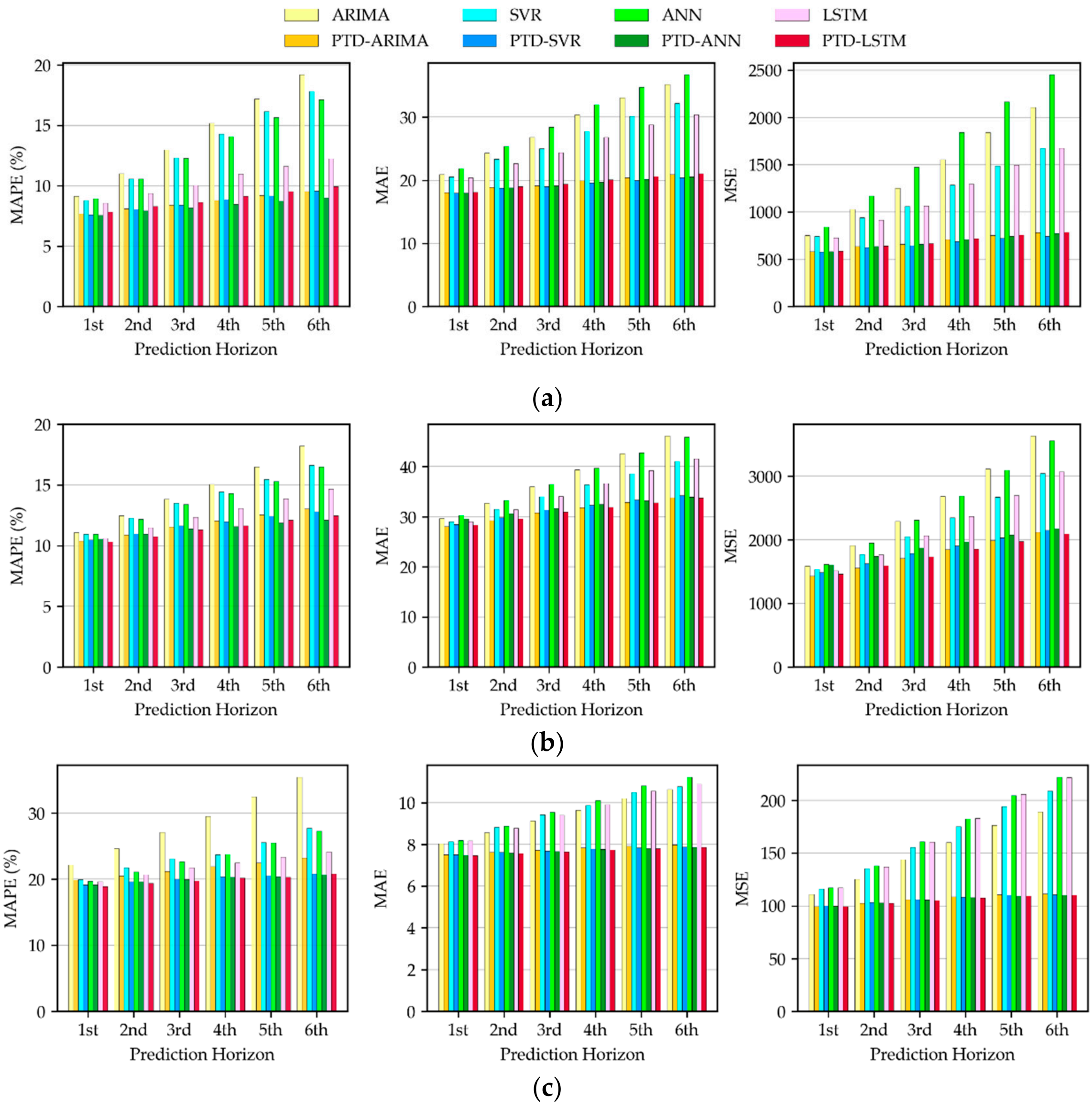
Table 3. Average measures evaluated across all the six-step prediction.

Models	S188			S195			818		
	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE
ARIMA	28.47	14.12%	1422.04	37.75	14.52%	2536.03	9.37	28.55%	151.08
PTD-ARIMA	19.56	8.62%	685.39	31.09	11.73%	1772.60	7.77	21.53%	106.51
improved	31.32%	38.95%	51.80%	17.65%	19.21%	30.10%	17.10%	24.60%	29.50%
SVR	26.53	13.32%	1198.28	35.09	13.87%	2234.97	9.59	23.61%	164.26
PTD-SVR	19.30	8.60%	665.39	31.60	11.72%	1830.18	7.72	20.08%	106.47
improved	27.25%	35.41%	44.47%	9.96%	15.53%	18.11%	19.44%	14.93%	35.18%
ANN	29.87	13.11%	1657.62	38.06	13.77%	2534.79	9.79	23.33%	171.05
PTD-ANN	19.40	8.31%	682.92	31.89	11.40%	1905.69	7.69	20.01%	106.11
improved	35.05%	36.62%	58.80%	16.22%	17.17%	24.82%	21.45%	14.21%	37.96%
LSTM	25.59	10.47%	1194.49	35.31	12.67%	2245.46	9.62	22.01%	170.88
PTD-LSTM	19.73	8.90%	692.90	31.23	11.44%	1784.21	7.68	19.88%	105.73
improved	22.89%	15.02%	41.99%	11.57%	9.74%	20.54%	20.20%	9.65%	38.12%

There is one thing that we need to explain; it seems that the testing models do not perform so well in case 818, in terms of MAPE. The MAPE of the best model PTD-LSTM is close to 20%. This is because that MAPE is sensitive to small true values (i.e., the y_i in Equation (35)). The traffic volume of 818 is much smaller than S188 and S195 (see Figure 5). When the value in the denominator is small, a small deviation in error will create a large change in the absolute percentage error (APE). For example, suppose two samples y_1 = 10 and y_2 = 100, and the predicted values of them are $\hat{y}_1 = 20$ and $\hat{y}_2 = 110$. The absolute error (AE) and square error (SE) of them are equal (i.e., AE_1 = AE_2 = 10 and SE_1 = SE_2 = 10^2 = 100). However, the APE of them is quite different (i.e., APE_1 = 100% and APE_2 = 10%). When the AEs or SEs are equal, the true values are smaller; the APEs are larger. As for MSEs and MAEs, these results are normal because the models are trained based on MSE loss.

5.3. Analysis of Multi-Step Prediction Errors

In order to analyze the multi-step prediction errors, the evaluation measures of each prediction horizon are presented in Figure 8. An aspect that should be noted is that all prediction errors increase along prediction horizons. This is owing to the cumulative errors, which stem from feeding predicted values into the models during a multi-step prediction. What stands out in this figure is that the errors of the individual models increase more rapidly along with prediction horizons than the hybrid models. Furthermore, the increased rates of hybrid models are almost the same. These findings indicate that no matter which prediction model it is integrated into, the hybrid model can always reduce cumulative errors effectively.



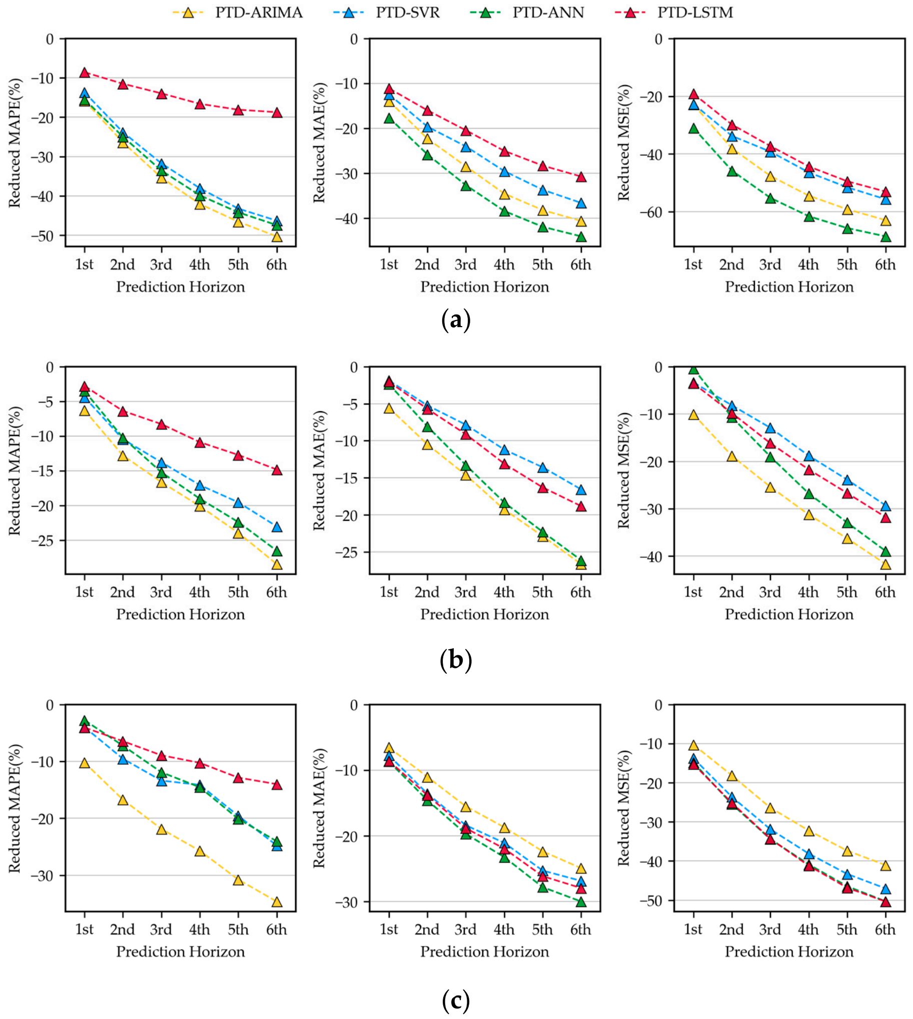
(**a**)

(**b**)

(**c**)

Figure 8. Results of multi-step prediction: (**a**) S188; (**b**) S195; (**c**) 188.

To elaborate on the contribution of PTD, the rate of reduced error is introduced, which is equal to the reduced error divided by the error of the individual model. The rates are computed and plotted in Figure 9. The negative rate represents that the hybrid model performs better than the corresponding individual model. The absolute value of the rate is larger, the improvement of performance is larger. It is apparent that the rates of reduced errors decline, along with prediction horizons. This represents that hybrid models are able to reduce the cumulative errors. The PTD approach can decompose the complex traffic flow into trend, periodicity, and remainder. These components vary more regularly and are modelled more easily. This is why the hybrid method can improve the prediction performance. From these findings, it demonstrates that the method combining PTD shows robustness for multi-step traffic flow forecasting.



(**a**)

(**b**)

(**c**)

Figure 9. Rates of reduced errors: (**a**) S188; (**b**) S195; (**c**) 818.

6. Conclusions

In this paper, to extract trend and periodic characteristics of traffic flow and achieve more accurate forecasting, a novel method combining PTD is proposed. PTD is developed to decompose the original traffic flow into trend, periodicity, and remainder. Then, the periodicity is predicted by cycling, while the trend and remainder by modelling. To test the applicability of the proposed method for different traffic flow patterns, three datasets with different distributions collected from Mn/DOT are utilized to conduct experiments and analyze comparatively. Our main work is concluded as follows:

1. A novel decomposition approach for traffic flow PTD is formulated. Fully considering the dynamicity of real-time traffic flow, PTD is formulated to decompose both in-sample and out-of-sample data. This approach can decompose the original traffic flow into trend, periodicity, and remainder.
2. A novel hybrid method combining PTD for traffic flow forecasting is developed. To demonstrate the universal applicability of the PTD approach, both statistical and AI models (i.e., ARIMA, SVR, ANN, LSTM) are combined with PTD, to establish hybrid models (i.e., PTD-ARIMA, PTD-SVR, PTD-ANN and PTD-LSTM). Empirical results show that the MAE, MAPE, and MSE of hybrid models are averagely reduced by 17%, 17%, and 29% than individual models, respectively.
3. After investigating multi-step prediction results, it is found that the hybrid models combining PTD can not only reduce the prediction errors, but also reduce cumulative errors. It suggests that the proposed hybrid method is robust for multi-step traffic flow forecasting.

Three limitations remain in our study and will be addressed in future study. Firstly, only traffic flow data on workdays is studied. The patterns of traffic flow in weekdays are different to that on weekends. In future studies, the PTD approach will be improved for whole week traffic flow decomposition. Secondly, only traffic volume data are tested and evaluated. In further studies, other measures of traffic status (such as traffic speed, travel time) will be collected to carry out experiments. Thirdly, only historical data are considered. In further studies, the spatial-temporal features or topology features of the road network will be considered for a more accurate prediction.

Author Contributions: Conceptualization, W.Z., W.W. and X.H.; Data curation, W.Z. and X.H.; Formal analysis, W.Z.; Funding acquisition, W.W. and X.H.; Investigation, W.Z. and Y.Z.; Methodology, W.Z.; Project administration, W.W. and X.H.; Resources, W.Z. and W.W.; Software, W.Z.; Supervision, W.W. and X.H.; Validation, W.Z.; Visualization, W.Z.; Writing—original draft, W.Z.; Writing—review and editing, W.Z., W.W., X.H. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 51878166 and 71801042; Natural Science of Jiangsu Province, grant number BK20180381.

Acknowledgments: The authors are grateful to the Minnesota Department of Transportation (Mn/DOT) for opening the traffic flow data.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Zahid, M.; Chen, Y.; Jamal, A.; Mamadou, C.Z. Freeway short-term travel speed prediction based on data collection time-horizons: A fast forest quantile regression approach. *Sustainability* **2020**, *12*, 646. [CrossRef]
2. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we're going. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 3–19. [CrossRef]
3. Chen, X.; Lu, J.; Zhao, J.; Qu, Z.; Yang, Y.; Xian, J. Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network. *Sustainability* **2020**, *12*, 3678. [CrossRef]
4. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [CrossRef]
5. Zhang, M.; Zhen, Y.; Hui, G.; Chen, G. Accurate multisteps traffic flow prediction based on SVM. *Math. Probl. Eng.* **2013**, *2013*. [CrossRef]
6. Luo, X.; Li, D.; Yang, Y.; Zhang, S. Spatiotemporal traffic flow prediction with KNN and LSTM. *J. Adv. Transp.* **2019**, *2019*. [CrossRef]
7. Jiang, X.; Adeli, H. Dynamic wavelet neural network model for traffic flow forecasting. *J. Transp. Eng.* **2005**, *131*, 771–779. [CrossRef]
8. Bratsas, C.; Koupidis, K.; Salanova, J.M.; Giannakopoulos, K.; Kaloudis, A.; Aifadopoulou, G. A comparison of machine learning methods for the prediction of traffic speed in Urban places. *Sustainability* **2020**, *12*, 142. [CrossRef]
9. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C* **2015**, *54*, 187–197. [CrossRef]
10. Wang, H.; Liu, L.; Dong, S.; Qian, Z.; Wei, H. A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework. *Transp. B* **2016**, *4*, 159–186. [CrossRef]
11. Yang, J.S. Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; Volume 3, pp. 2128–2133. [CrossRef]
12. Zhang, Y.; Liu, Y. Traffic forecasting using least squares support vector machines. *Transportmetrica* **2009**, *5*, 193–213. [CrossRef]
13. Zou, Y.; Hua, X.; Zhang, Y.; Wang, Y. Hybrid short-term freeway speed prediction methods based on periodic analysis. *Can. J. Civ. Eng.* **2015**, *42*, 570–582. [CrossRef]
14. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 606–616. [CrossRef]
15. Ahmed, M.S.; Cook, A.R. Analysis of Freeway Traffic Time-Series Data By Using Box-Jenkins Techniques. *Transp. Res. Rec.* **1979**, 1–9.

16. Li, L.; Qin, L.; Qu, X.; Zhang, J.; Wang, Y.; Ran, B. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowl.-Based Syst.* **2019**, *172*, 1–14. [CrossRef]
17. Tselentis, D.I.; Vlahogianni, E.I.; Karlaftis, M.G. Improving short-term traffic forecasts: To combine models or not to combine? *IET Intell. Transp. Syst.* **2015**, *9*, 193–201. [CrossRef]
18. Lin, L.; Wang, Q.; Huang, S.; Sadek, A.W. On-line prediction of border crossing traffic using an enhanced Spinning Network method. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 158–173. [CrossRef]
19. Luo, X.; Li, D.; Zhang, S. Traffic flow prediction during the holidays based on DFT and SVR. *J. Sens.* **2019**, *2019*. [CrossRef]
20. Kolidakis, S.; Botzoris, G.; Profillidis, V.; Lemonakis, P. Road traffic forecasting—A hybrid approach combining Artificial Neural Network with Singular Spectrum Analysis. *Econ. Anal. Policy* **2019**, *64*, 159–171. [CrossRef]
21. Guo, F.; Krishnan, R.; Polak, J.W. Short-term traffic prediction under normal and incident conditions using singular spectrum analysis and the k-nearest neighbour method. In Proceedings of the IET and ITS Conference on Road Transport Information and Control (RTIC 2012), London, UK, 25–26 September 2012. [CrossRef]
22. Chen, W.; Shang, Z.; Chen, Y.; Chaeikar, S.S. A Novel Hybrid Network Traffic Prediction Approach Based on Support Vector Machines. *J. Comput Netw. Commun.* **2019**, *2019*. [CrossRef]
23. Shang, Q.; Lin, C.; Yang, Z.; Bing, Q.; Zhou, X. A hybrid short-term traffic flow prediction model based on singular spectrum analysis and kernel extreme learning machine. *PLoS ONE* **2016**, *11*. [CrossRef] [PubMed]
24. Zheng, C.; Li, L. The improvement of the forecasting model of short-term traffic flow based on wavelet and ARMA. In Proceedings of the SCMIS 2010—2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering, Hong Kong, China, 6–9 October 2010.
25. Zhang, N.; Guan, X.; Cao, J.; Wang, X.; Wu, H. Wavelet-HST: A Wavelet-Based Higher-Order Spatio-Temporal Framework for Urban Traffic Speed Prediction. *IEEE Access* **2019**, *7*, 118446–118458. [CrossRef]
26. Okutani, I.; Stephanedes, Y.J. Dynamic prediction of traffic volume through Kalman filtering theory. *Transp. Res. Part B* **1984**, *18*, 1–11. [CrossRef]
27. Gong, X.; Wang, F. Three improvements on KNN-NPR for traffic flow forecasting. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Singapore, 6 September 2002; pp. 736–740. [CrossRef]
28. Yu, B.; Song, X.; Guan, F.; Yang, Z.; Yao, B. K-Nearest Neighbor Model for Multiple-Time-Step Prediction of Short-Term Traffic Condition. *J. Transp. Eng.* **2016**, *142*. [CrossRef]
29. Xiao, J.; Wei, C.; Liu, Y. Speed estimation of traffic flow using multiple kernel support vector regression. *Phys. A Stat. Mech. Appl.* **2018**, *509*, 989–997. [CrossRef]
30. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transp. Res. Part. C Emerg. Technol.* **2005**, *13*, 211–234. [CrossRef]
31. Kumar, K.; Parida, M.; Katiyar, V.K. Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport* **2015**, *30*, 397–405. [CrossRef]
32. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* **2017**, *17*, 818. [CrossRef]
33. Smith, B.L.; Demetsky, M.J. Traffic flow forecasting: Comparison of modeling approaches. *J. Transp. Eng.* **1997**, *123*, 261–266. [CrossRef]
34. Jia, Y.; Wu, J.; Xu, M. Traffic flow prediction with rainfall impact using a deep learning method. *J. Adv. Transp.* **2017**, *2017*. [CrossRef]
35. Zeng, D.; Xu, J.; Gu, J.; Liu, L.; Xu, G. Short term traffic flow prediction using hybrid ARIMA and ANN models. In Proceedings of the 2008 Workshop on Power Electronics and Intelligent Transportation System, PEITS 2008, Guangzhou, China, 2–3 August 2008; pp. 621–625. [CrossRef]
36. Tan, M.C.; Wong, S.C.; Xu, J.M.; Guan, Z.R.; Zhang, P. An aggregation approach to short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 60–69. [CrossRef]
37. Hou, Q.; Leng, J.; Ma, G.; Liu, W.; Cheng, Y. An adaptive hybrid model for short-term urban traffic flow prediction. *Phys. A Stat. Mech. Appl.* **2019**, *527*. [CrossRef]
38. Li, L.; He, S.; Zhang, J.; Ran, B. Short-term highway traffic flow prediction based on a hybrid strategy considering temporal–spatial information. *J. Adv. Transp.* **2016**, *50*, 2029–2040. [CrossRef]

39. Chan, K.Y.; Dillon, T.S.; Singh, J.; Chang, E. Traffic flow forecasting neural networks based on exponential smoothing method. In Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, ICIEA 2011, Beijing, China, 21–23 June 2011; pp. 376–381. [CrossRef]
40. Yang, H.F.; Dillon, T.S.; Chang, E.; Chen, Y.P.P. Optimized Configuration of Exponential Smoothing and Extreme Learning Machine for Traffic Flow Forecasting. *IEEE Trans. Ind. Inform.* **2019**, *15*, 23–34. [CrossRef]
41. Hong, W.C.; Dong, Y.; Zheng, F.; Wei, S.Y. Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. *Appl. Math. Comput.* **2011**, *217*, 6733–6747. [CrossRef]
42. Zhang, Y.; Huang, G. Traffic flow prediction model based on deep belief network and genetic algorithm. *IET Intell. Transp. Syst.* **2018**, *12*, 533–541. [CrossRef]
43. Feng, X.; Ling, X.; Zheng, H.; Chen, Z.; Xu, Y. Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2001–2013. [CrossRef]
44. Nihan, N.L.; Holmesland, K.O. Use of the box and Jenkins time series technique in traffic forecasting. *Transportation* **1980**, *9*, 125–143. [CrossRef]
45. Yang, B.; Sun, S.; Li, J.; Lin, X.; Tian, Y. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing* **2019**, *332*, 320–327. [CrossRef]
46. Li, Z.; Li, Y.; Li, L. A comparison of detrending models and multi-regime models for traffic flow prediction. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 34–44. [CrossRef]
47. Chen, C.; Wang, Y.; Li, L.; Hu, J.; Zhang, Z. The retrieval of intra-day trend and its influence on traffic prediction. *Transp. Res. Part C Emerg. Technol.* **2012**, *22*, 103–118. [CrossRef]
48. Dai, X.; Fu, R.; Zhao, E.; Zhang, Z.; Lin, Y.; Wang, F.Y.; Li, L. DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending. *Transp. Res. Part C Emerg. Technol.* **2019**, *103*, 142–157. [CrossRef]
49. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A seasonal-trend decomposition procedure based on loess. *J. Off. Stat.* **1990**, *6*, 3–73.
50. Rojo, J.; Rivero, R.; Romero-Morte, J.; Fernández-González, F.; Pérez-Badia, R. Modeling pollen time series using seasonal-trend decomposition procedure based on LOESS smoothing. *Int. J. Biometeorol.* **2017**, *61*, 335–348. [CrossRef] [PubMed]
51. Theodosiou, M. Forecasting monthly and quarterly time series using STL decomposition. *Int. J. Forecast.* **2011**, *27*, 1178–1195. [CrossRef]
52. Stow, C.A.; Cha, Y.; Johnson, L.T.; Confesor, R.; Richards, R.P. Long-term and seasonal trend decomposition of maumee river nutrient inputs to western lake erie. *Environ. Sci. Technol.* **2015**, *49*, 3392–3400. [CrossRef]
53. Xiong, T.; Li, C.; Bao, Y. Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China. *Neurocomputing* **2018**, *275*, 2831–2844. [CrossRef]
54. Qin, L.; Li, W.; Li, S. Effective passenger flow forecasting using STL and ESN based on two improvement strategies. *Neurocomputing* **2019**, *356*, 244–256. [CrossRef]
55. Bartholomew, D.J.; Box, G.E.P.; Jenkins, G.M. Time Series Analysis Forecasting and Control. *Oper. Res. Q. (1970-1977)* **1971**, *22*, 199. [CrossRef]
56. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. Training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152. [CrossRef]
57. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
58. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
59. The Minnesota Department of Transportation. Mn/DOT Traffic Data. Available online: http://data.dot.state.mn.us/datatools/ (accessed on 11 March 2020).
60. Tian, Y.; Pan, L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the 2015 IEEE International Conference on Smart City, Chengdu, China, 19–21 December 2015; pp. 153–158. [CrossRef]