modelling

MDPI

Article

Short-Term Highway Traffic Flow Prediction via Wavelet–Liquid Neural Network Model

Yongjun Wu [1,*], Hongyun Kang [1], Weipin Wang [2], Shuli Zhao [2], Xuening He [2] and Jingyao Chen [2]

[1] School of Traffic & Transportation, Chonqqing Jiaotong University, Chongqing 400074, China
[2] Chongqing Expressway Network Management Co., Ltd., Chongqing 401120, China
* Correspondence: yongjunwu@cqjtu.edu.cn

Abstract: Accurate, efficient, and reliable traffic flow prediction is pivotal for highway operation and management. However, traffic flow series present nonlinear, heterogeneous, and stochastic characteristics, posing significant challenges to precise prediction. To address this issue, this paper proposes a novel wavelet-LNN model, integrating the strengths of wavelet decomposition and liquid neural networks (LNNs). Initially, multi-scale wavelet decomposition is applied to the original traffic flow data to yield approximation components and detailed components. Subsequently, each component is trained using the LNN. Ultimately, the predicted results of all components of the LNN models are aggregated to derive the final traffic flow prediction. The experiments conducted on four highway datasets demonstrate that the proposed wavelet-LNN model surpasses SVR, LSSVM, LSTM, TCN, and transformer models in prediction performance across R2, MSE, and MAE metrics. Notably, the wavelet-LNN model features the fewest parameters (<2% of typical deep learning models).

Keywords: traffic flow prediction; wavelet decomposition; liquid neural network; intelligent transportation systems

check for updates

Academic Editor: Wei Gao

Received: 25 November 2024
Revised: 24 December 2024
Accepted: 25 December 2024
Published: 14 May 2025

Citation: Wu, Y.; Kang, H.; Wang, W.; Zhao, S.; He, X.; Chen, J. Short-Term Highway Traffic Flow Prediction via Wavelet–Liquid Neural Network Model. *Modelling* **2025**, *6*, 39. https://doi.org/10.3390/modelling6020039

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Highway traffic flow prediction is a vital component of intelligent transportation systems [1], and video-based traffic flow detection technology is increasingly being adopted to obtain real-time traffic flow data. Moreover, it is crucial for ensuring the smooth operation of transportation systems and enhancing road safety. Accurate short-term traffic flow prediction provides essential data for travelers' route choices and the development of traffic management and control strategies, which are key to easing traffic congestion and improving traffic efficiency. By predicting short-term traffic peak periods, authorities can adjust speed limits, signal timing, and lane allocation to manage traffic flow more effectively.

Traffic flow data have a time series structure, and various time series prediction methods have been applied to traffic flow prediction. These methods can be categorized into classical statistical learning methods, traditional machine learning methods, neural network methods, deep learning methods, and combination model methods.

Classical statistical learning approaches for traffic flow prediction mainly include vector autoregression (VAR) [2], the Kalman filter [3,4], and the autoregressive-integrated moving average model (ARIMA) [5]. The direct application of these methods for real traffic flow prediction typically yields unsatisfactory results, especially when dealing with nonlinear changes in traffic flow. Consequently, numerous variant methods have been proposed, such as the combination of topology-regularized autoregression with

vector autoregression for traffic forecasting advocated by Schimbinschi et al. [6], and the Bayesian vector autoregression method developed by Li et al. [7]. For short-term traffic flow forecasting, noise-immune methods have been introduced in the Kalman filter [8], and seasonal ARIMA (SARIMA) models [9,10] have been developed to account for the cyclical nature of traffic flow. Additionally, the hybrid dual Kalman filtering model [11] and Tensor extended Kalman filter [12] have been proposed for short-term traffic flow forecasting. However, these methods assume stable traffic flow time series and complete data, thus neglecting the nonlinearity of the data. As a result, they struggle to achieve satisfactory prediction results.

Traditional machine learning methods include k-nearest neighbor (KNN) methods [13], hidden Markov models (HMMs) [14], gradient boosting decision trees (GBDTs) [15], support vector regression (SVR) models [16], and Bayesian networks [17], among others. These methods can handle complex nonlinear correlations in traffic flow series. Improved versions of these methods have also been developed, such as the combination of SVR and KNN for traffic flow prediction by Lin et al. [18], the improved HMM developed for urban road networks by Zhu et al. [19], and the integration of LSSVM with hybrid optimization for short-term traffic flow prediction [20]. Although these methods demonstrate some nonlinear approximation capabilities, they only extract shallow features, limiting their effectiveness.

Neural network methods commonly used for time series prediction include long short-term memory (LSTM) methods [21], gated recurrent unit (GRU) methods [22], convolutional neural networks (CNNs) [23], bidirectional long short-term memory (BILSTM) [24], and temporal convolutional networks (TCNs) [25]. To enhance performance, various methods have been proposed, such as integrating attention mechanisms and optimization methods into LSTM [26,27] and designing shared-weight GRU networks [28]. Additionally, combination models like LSTM-BILSTM [29], Grey-CNN [30], and Conv-LSTM [31] have been introduced. While these neural network models perform well in nonlinear approximation for traffic flow series, they may still suffer from gradient vanishing in long sequences, affecting learning effectiveness.

In recent years, deep learning methods have made significant progress in image processing and natural language processing and have been applied to traffic flow prediction [32,33]. Representative models include deep belief networks [34,35], the transformer model [36], and graph convolutional networks [37,38]. The transformer model, in particular, uses a self-attention mechanism to simultaneously consider all positions in the input sequence, effectively capturing long-range dependencies and showing great potential. Consequently, variant transformer models such as RPConvformer [39] and MTS-Informer [40] have been proposed. These deep neural network-based methods perform well in predicting traffic flow but have some drawbacks, including high model complexity, large data sample requirements, and high computational complexity, making it difficult for the model to generalize to real-world scenarios.

To summarize, classical statistical learning methods and traditional machine learning methods struggle to adapt to the nonlinearity of traffic series. Neural network methods are not adept at capturing long-term dependencies and have weak generalization abilities. Deep learning methods often require high computational complexity and large data samples, and exhibit weak generalization abilities. To address the nonlinear characteristics and high noise interference in traffic flow series, many researchers have attempted to combine decomposition algorithms with neural networks. For example, wavelet decomposition has been integrated with graph convolutional networks [41], empirical mode decomposition (EMD) has been used in combination model fusion [42], and variational mode decomposition (VMD) has been employed for extreme learning machines [43]. These results indicate that component decomposition-based mixed traffic flow forecasting models are more effective in capturing the main regularities and random changes in traffic flow than

single models. In practical applications, wavelet decomposition is widely applicable and easy to use.

Meanwhile, the liquid neural network (LNN) has been proposed, which can continue adapting to new stimuli after training, demonstrating strong performances in time series prediction [44]. Additionally, the Liquid Foundation Models (LFMs), developed by Liquid AI, achieve industry-leading performance at different scales while maintaining smaller memory usage and more efficient inference capabilities compared to ChatGPT 3.5, which is based on the transformer architecture. The LNN model has been employed in robust flight navigation out of distribution, successfully performing vision-based fly-to-target tasks beyond its training environment [45]. By using an LNN with 19 nodes, autonomous driving control has been achieved [46], reducing the number of neurons by tens of thousands compared to conventional neural network methods. This highlights the efficiency, interpretability, robustness, and effectiveness of the LNN model.

In summary, real-time short-term traffic flow prediction is crucial. Traditional methods have small model scales, weak feature extraction capabilities, and poor generalization performance. Deep learning methods have strong feature extraction capabilities but large model sizes and high computational complexity. Leveraging the strengths of wavelet decomposition and the LNN model, this paper designs a novel wavelet-LNN model for traffic flow prediction. The original traffic flow series is decomposed using wavelet decomposition and LNN models are trained to predict each decomposed sub-series.

The structure of this paper is outlined as follows: Section 2 delves into the architecture of the wavelet-LNN model, elucidating its design and the algorithms that power it. Section 3 involves an exhaustive analysis, comparing the predictive accuracy of the wavelet-LNN model with various benchmark models. Section 4 wraps up the paper with a summary of the findings.

2. Materials and Methods

2.1. Multi-Scale Wavelet Decomposition

The time series of traffic flow exhibits instability, volatility, and intermittency. To address these challenges, wavelet decomposition is employed to handle the nonlinearity and high noise levels. The multi-scale wavelet decomposition, based on the wavelet transform, uses wavelet basis functions to approximate the original signal series. Specifically, multi-scale wavelet basis functions are utilized to implement the multi-scale wavelet decomposition. The wavelet basis function is designed as follows:

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}}\varphi\left(\frac{t-b}{a}\right) \tag{1}$$

where φ is the mother wavelet basis function, and a and b denote the scale parameter and shift parameter. The wavelet coefficients would be calculated as follows:

$$c_{a,b}(t) = \left\langle \varphi\left(\frac{t-b}{a}\right), x(t) \right\rangle = \int_{-\infty}^{\infty} \varphi\left(\frac{t-b}{a}\right) I(t)dt \tag{2}$$

where $c_{a,b}(t)$ denotes the wavelet coefficients of $I(t)$ at time t.

Using the wavelet coefficients, the traffic flow series can be reconstructed by using the inverse transform, and it can be expressed as follows:

$$\hat{I}(t) = \sum_{a,b} c_{a,b}\varphi_{a,b}(t) \tag{3}$$

where $\hat{I}(t)$ is the reconstructed series.

The orthogonal wavelet basis function is usually used, and the scale and shift parameters are modified as follows:

$$\begin{aligned} a &= a_0^m \\ b &= nb_b \end{aligned} \tag{4}$$

where m and n are integers. Moreover, choose $b_0 = \beta a_0^m$ to ensure fixed a cover $x(t)$. For the rapid calculation of the wavelet coefficients on the discrete time series, the a_0 and β are usually set to 2 and 1, respectively. Accordingly, the wavelet basis function would be expressed as follows:

$$\varphi_{m,n}(t) = 2^{-\frac{m}{2}} \varphi\left(\frac{t - n \cdot 2^m}{2^m}\right) \tag{5}$$

In order to perform wavelet decomposition on discrete traffic flow series, wavelet basis functions are employed to generate low-pass filters and high-pass filters. The wavelet approximation and detail coefficients can be calculated using these low-pass and high-pass filters, respectively. By applying the inverse transform, the corresponding approximation component and detail component are obtained.

By utilizing wavelet basis functions with parameters of different scales, multi-scale wavelet coefficients and components can be obtained. For discrete traffic flow series, multi-scale wavelet decomposition is carried out, and the processing flow is illustrated in Figure 1.

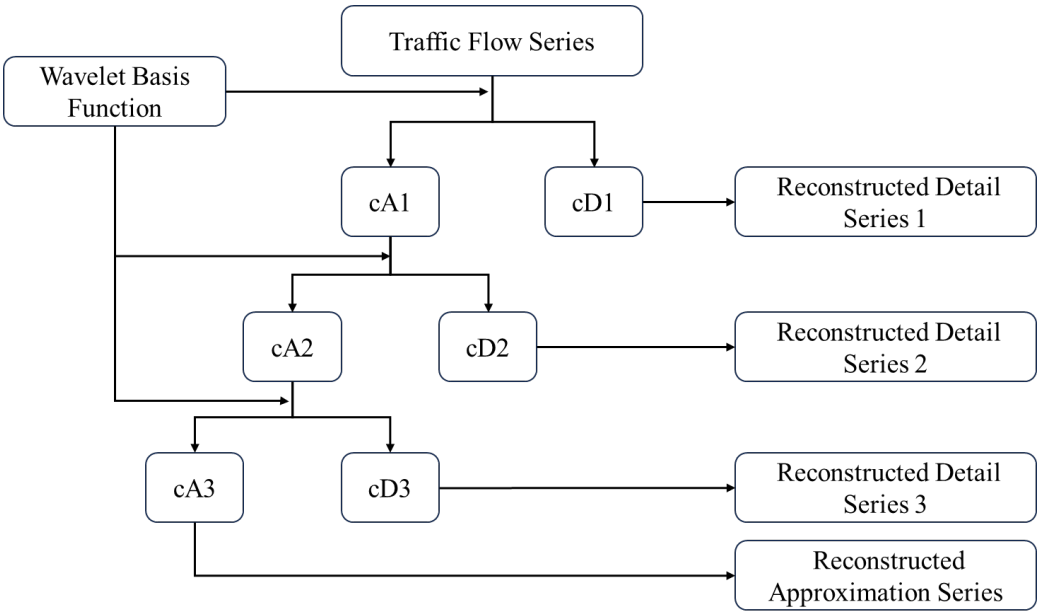


Figure 1. The structure of multi-scale wavelet decomposition.

As shown in Figure 1, three-layer wavelet decomposition is applied to traffic flow series, and four component series are generated.

2.2. Liquid Neural Networks

The liquid neural networks (LNNs) represent a type of recurrent neural network (RNN) that is time-continuous and inspired by the nervous system of the nematode. Continuous-time hidden-state RNNs, it can be represented by ordinary differential equations and it can be expressed as follows:

$$\frac{dx(t)}{dt} = f(x(t), I(t), t, \theta) \tag{6}$$

where $f(\ )$ is a neural network parametrized by θ, $x(t)$ denotes the hidden state, $I(t)$ is the input, and t represents time.

In order to determine stable continuous-time recurrent neural network, an improved ordinary differential equation is proposed:

$$\frac{dx(t)}{dt} = -\frac{x(t)}{\tau} + f(x(t), I(t), t, \theta) \tag{7}$$

where the term of $-\frac{x(t)}{\tau}$ is designed to assist the system to reach an equilibrium state with a time-constant τ. Alternative, the model would be rewritten as follows:

$$\frac{dx(t)}{dt} = -\frac{x(t)}{\tau} + S(t) \tag{8}$$

where $S(t)$ represents the following nonlinearity. This is determined as follows:

$$S(t) = f(x(t), I(t), t, \theta)(A - x(t)) \tag{9}$$

Combining Equations (8) and (9), the ordinary differential equation for LNNs is obtained:

$$\frac{dx(t)}{dt} = -\left[\frac{1}{\tau} + f(x(t), I(t), t, \theta)\right] x(t) + f(x(t), I(t), t, \theta)A \tag{10}$$

where A is the bias component of the neural network, and the structure of LNNs is shown in Figure 2.

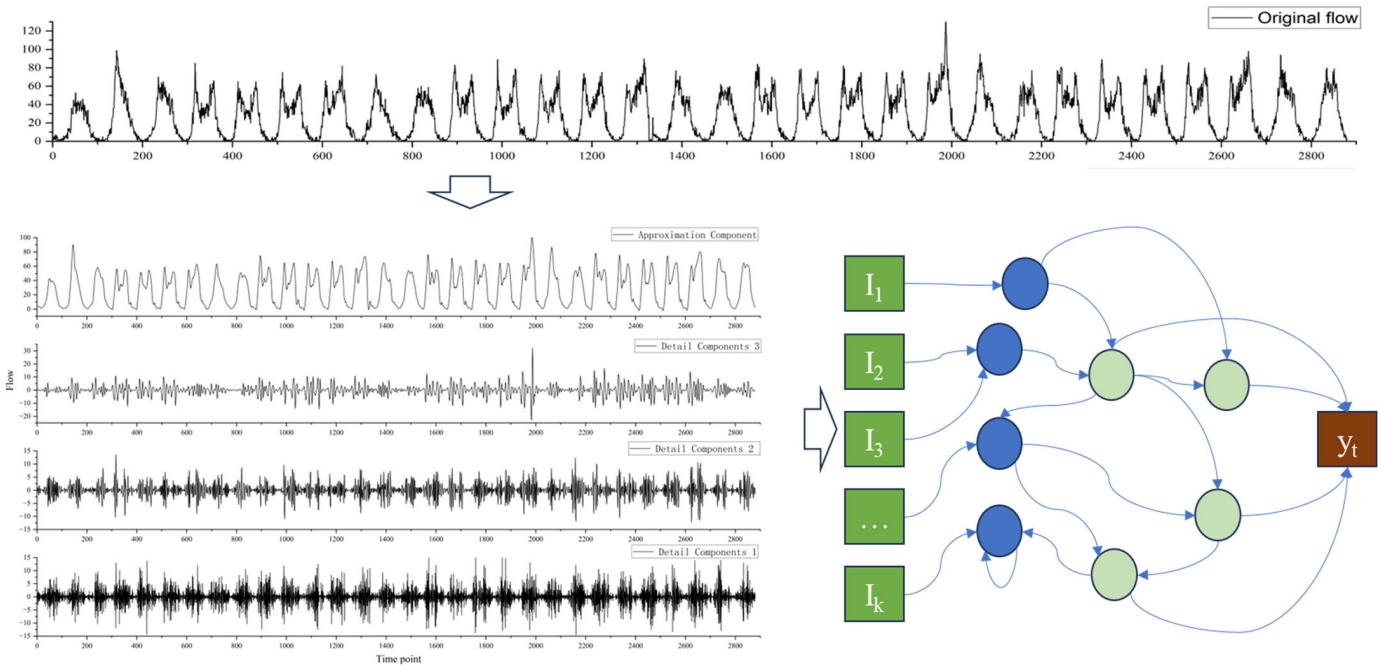


Figure 2. The structure of liquid neural networks.

The forward information transmission of LNNs is achieved by solving the corresponding ODE. For LNNs, any ODE solver can be used. A practical fixed-step ODE solver is used here, which combines the stability of implicit Euler methods with the computational efficiency of explicit Euler methods. It is necessary to decompose the interval [0, T] into time discretization $[t_0, t_1, \ldots, t_n]$. Therefore, the solving step only involves updating the hidden state from t_i to t_{i+1}. The iterative process is as follows:

$$x(t + \Delta t) = \frac{x(t) + \Delta t f(x(t), I(t), t, \theta)A}{1 + \Delta t(1/\tau + f(x(t), I(t), t, \theta))} \tag{11}$$

where Δt is the time intervals.

Through iteration, the hidden state of neurons at each moment can be obtained sequentially, thereby completing the forward transmission of LNNs. Moreover, an approximation of its closed-form solution is proposed in [17,47], and it can be expressed as follows:

$$x(t) \approx (x_0 - A)e^{-[\omega_t + f(I(t), t, \theta)]t} f(-I(t), t, \theta) + A \tag{12}$$

where $\omega_t = \frac{1}{\tau}$. By using approximate solutions, the model can be integrated into neural networks. For the hidden state in the D dimension, the expression at each time step is follows:

$$x(t) = (x_0 - A)e^{-[\omega_t + f(x,I;\theta)]t} f(-x, -I; \theta) + A \tag{13}$$

By replacing biases with learnable instances and setting the gating balance [17], the approximation of an ODE system is calculated quickly.

2.3. Wavelet-LNN Model

Although the LNN model exhibits good dynamic stability and superior expressivity, it struggles to achieve high accuracy when directly applied to traffic flow prediction. This is due to the nonlinearity, heterogeneity, and randomness of traffic flow series. To enhance the performance of LNNs in traffic flow prediction, wavelet decomposition is introduced, and the wavelet-LNN network is constructed. The structure of the wavelet-LNN network is shown in Figure 3.

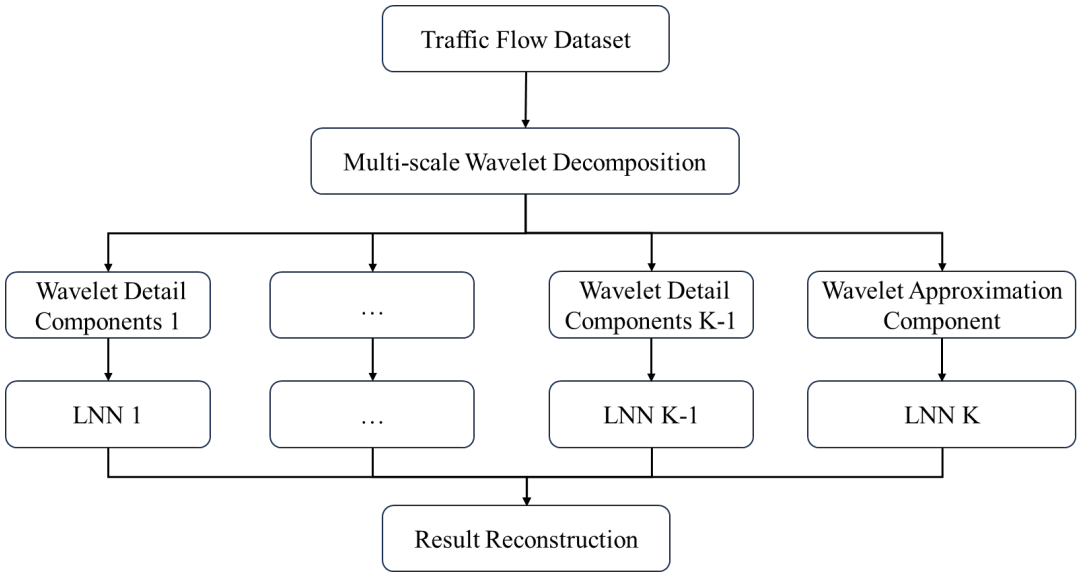


Figure 3. Structure of wavelet-LNN networks.

In the wavelet-LNN network, the wavelet decomposition is used to decompose the traffic flow time series into multiple sub-series. By setting the layers of multi-scale wavelet decomposition (MWD) at $K-1$, K sub-series are obtained:

$$[I_1(t), I_2(t), \cdots I_K(t)] = MWD(I(t)) \tag{14}$$

where $MWD(\ )$ is the function of multi-scale wavelet decomposition, $I(t)$ is thw input traffic flow time series, and $I_1(t), I_2(t), \cdots I_K(t)$ denote the decomposed sub-series.

For each sub-series, an LNN model is built to deal with it. This can be expressed as follows:

$$p_k(t) = LNN_k(I_k(t)) \tag{15}$$

where $LNN_k(\ )$ denotes the k-th LNN model and $p_k(t)$ is the predicted sub-series by LNN model. Each LNN model is trained under the supervision of the Mean Squared Error (MSE) loss function, and the MSE is calculated as follows:

$$Loss_k = \sum_{t=1}^{N} (I_k(t) - p_k(t))^2 \tag{16}$$

where N denotes the length of traffic flow time series.

One can then combine the predicted sub-series of each LNN model, and the final predicted result is obtained. This calculated as follows:

$$I_p(t) = \sum_{k=1}^{K} p_k(t) \tag{17}$$

where I_p denotes the predicted traffic flow series.

To sum up, the pseudocode of the proposed wavelet-LNN is shown as Algorithm 1.

Algorithm 1 The Wavelet-LNN algorithm

Input: traffic flow: $I(t)$, number of wavelet decomposition layers: K, maximum number of iterations: *MaxIter*, learning rate: lr, sliding window size: L
For k = 1 to K − 1
$I_k(t) \leftarrow MWD(I(t), k)$
For i = 1 to *MaxIter*
Initialize: θ_k
$I_{ki}(t) \leftarrow I_k(t-L) : I_k(t-1)$
$p_k(t) \leftarrow LNN_k(I_{ki}(t), \theta_k)$
$Loss_k \leftarrow MSE(I_k(t), p_k(t))$
$\theta_k \leftarrow Adam(MSE, lr)$
End
$I_p(t) \leftarrow I_p(t) + p_k(t)$
End
Output: $I_p(t)$

3. Experiments and Results

3.1. Dataset

In order to test the performance of the proposed wavelet-LNN model, the data sourced from the UK highway dataset https://webtris.highwaysengland.co.uk/, (accessed on 20 November 2024) is used for experimental validation. To test the robustness of the model, traffic flow tests were conducted at different times, locations, and vehicle types. To be specific, data are selected from observation point 7933M on M18 highway from 1 January 2022 to 30 January 2022 and from observation point 4919A on M25 highway from 1 August 2018 to 30 August 2018. The location map of dataset M18 and M25 are shown in Figure 4. Moreover, four sets of data with vehicle lengths of 0–5.2 m and 5.2–6.6 m were extracted as the source data for the experiment. The sampling frequency is set to so that there is a recording every 15 min. After preprocessing steps such as missing value filling, the length of each set of data is 2880, and description and statistics of datasets are shown in Table 1.

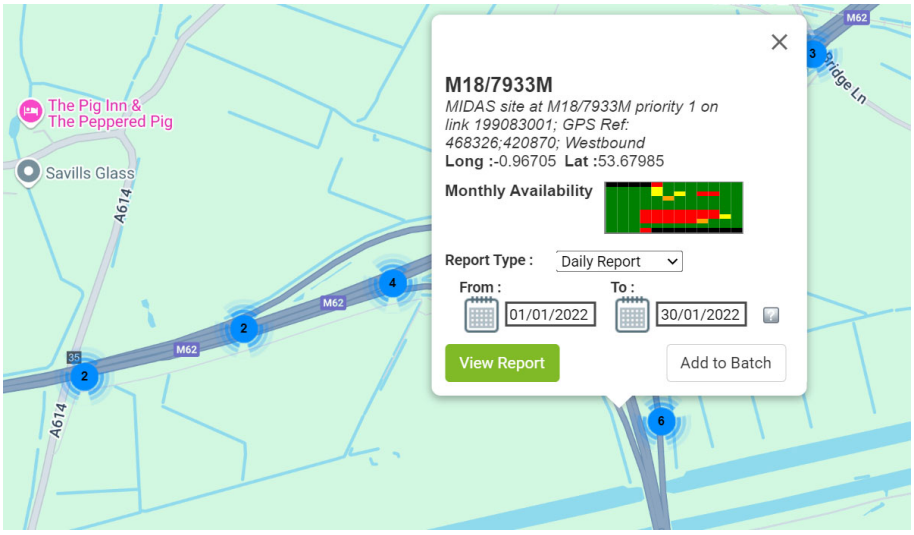


Figure 4. *Cont.*

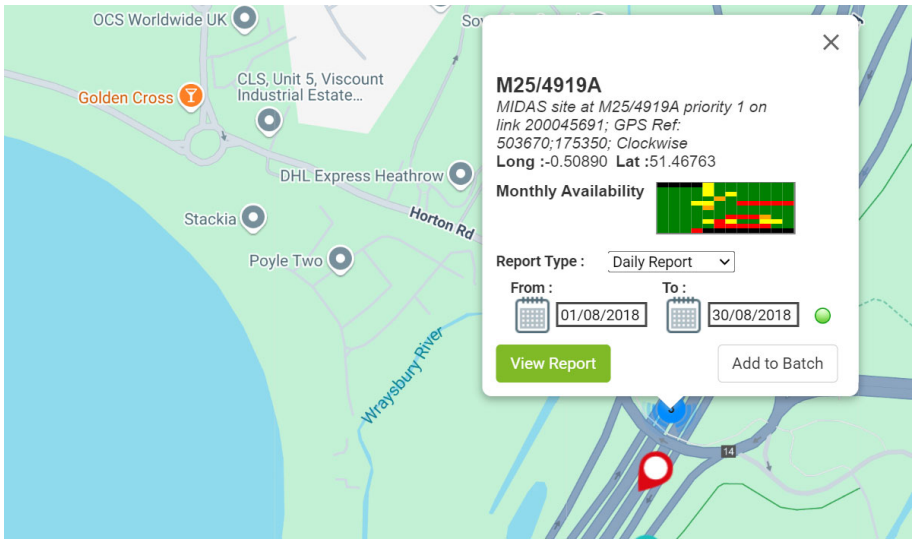


Figure 4. The location map of dataset M18 and M25. M18: GPS Ref (468,326, 420,870); M25: GPS Ref (503,670, 175,350).

Table 1. Description and statistics of Highways England datasets.

Dataset Name	Vehicle Lengths	Time Interval	Time Range	Data Points
M18 (0–5.2)	0–5.2 m	15 min	1 January 2022–30 January 2022	2880
M18 (5.2–6.6)	5.2–6.6 m	15 min	1 January 2022–30 January 2022	2880
M25 (0–5.2)	0–5.2 m	15 min	1 August 2018–30 August 2018	2880
M25 (5.2–6.6)	5.2–6.6 m	15 min	1 August 2018–30 August 2018	2880

For the four datasets of the M18 and M25, the histograms of traffic flow for each dataset are shown in Figure 5.

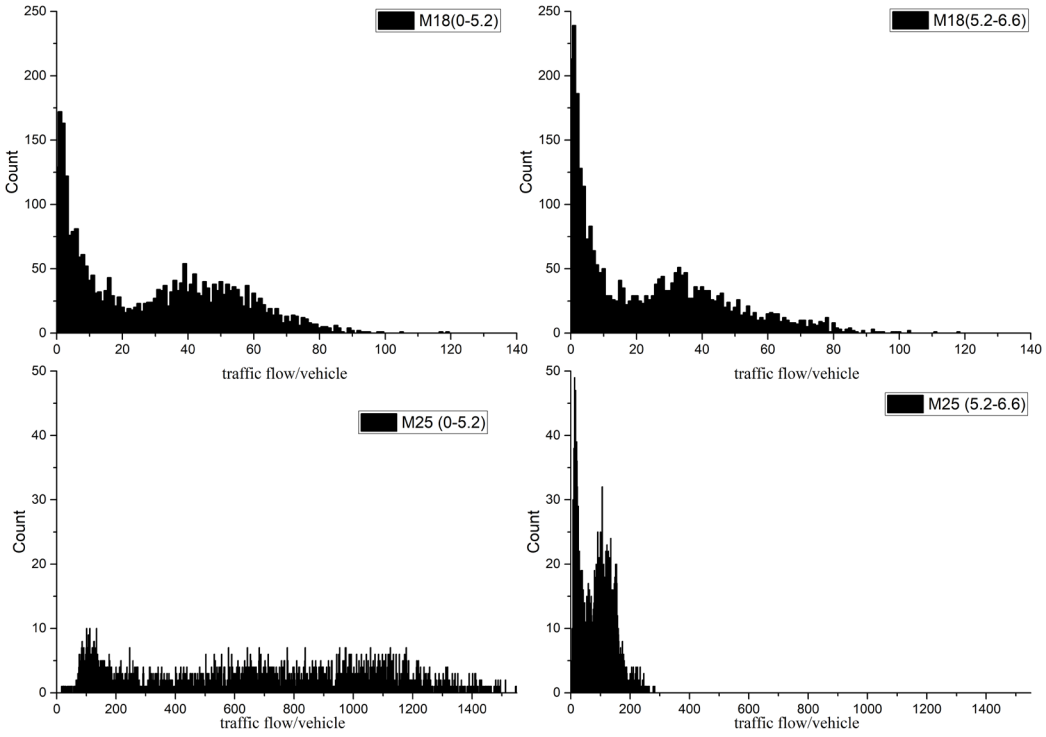


Figure 5. Histogram of M18 and M25 datasets.

As shown in Figure 5, the statistical distributions of the four datasets have significant differences. Each dataset in Table 1 is normalized using the Z-score method [48] and divided

into training and testing sets with a ratio of 7:3 (70% of the data as the training set and 30% as the test set). It is necessary to use the traffic flow data from the previous 16 points to predict the traffic flow data for the next time point and generate the corresponding dataset through a sliding window.

3.2. Experiment Settings

LNN [44] models are introduced for comparison with benchmarked methods such as the classical support vector regression (SVR) method [16], the least square support vector machine (LSSVM) [49], long short-term memory (LSTM) [50], temporal convolutional network (TCN) [51], and advanced transformer [3]. Moreover, to ensure fairness in the comparisons, combined model wavelet decomposition and LSSVM (wavelet-LSSVM), wavelet decomposition, and LSTM (wavelet-LSTM) are also used as benchmarked models. For each method, the parameter settings are as follows:

SVR: using cross validation (GridSearchCV) for parameter optimization, according to the order of the dataset in Table 1, the optimal parameters C for SVR are 12.91, 100.00, 0.5994, and 100.00, and gamma values are 0.5994, 0.0774, 0.5994, and 0.0774.

LSSVM: for the LSSVM model, the gamma parameter is 234, the kernel is RBF, and the sigma is 1.

LSTM: the epoch is 500, the batch size is 64, the number of hidden layers num_1ayers is 5, the hidden layer size hidden_2 is 64, and the learning rate is 0.001.

TCN: the kernel size is 3, and the numbers of convolutional channels are set to 32, 64, 128, and 256, respectively.

Transformer: the number of hidden layer neurons is 64, the number of attention mechanisms num_heads is 8, and the number of encoders and decoders is 3.

Wavelet-LNN: For the proposed wavelet-LNN model, the number of iterations for model training is 300, and batch size is set to be 64. Adam is employed as optimizer and the learning rate is 0.001.

The proposed wavelet-LNN model is implemented based on Python 3.8.3 and PyTorch 2.0.1 (with cuda version cu117). The optimizer is Adam and the learning rate is set to be 0.01. The maximum number of training iterations is 400. The selection of operations on a Windows 10 laptop is performed with the identifier LAPTOP-9EJMKSQA, and all benchmarked models are executed on a computer equipped with one RTX 3060Ti GPU.

3.3. Measures of Performance

To evaluate the performance of the proposed model, the correlation coefficient R^2, mean absolute error E_{MSE}, and root mean square error E_{MSE} are selected to evaluate the prediction results. The calculation formula is as follows:

$$E_{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|I_t - I_t'\right| \tag{18}$$

$$E_{MSE} = \frac{1}{N}\sum_{t=1}^{N}\left(I_t - I_t'\right)^2 \tag{19}$$

$$R^2 = \frac{N\sum_{t=1}^{N} I_t I_t' - \sum_{t=1}^{N} I_t \sum_{t=1}^{N} I_t'}{\sqrt{N\sum_{t=1}^{N} I_t^2 - \left(\sum_{t=1}^{N} I_t\right)^2}\sqrt{N\sum_{t=1}^{N} I_t^{2\prime} - \left(\sum_{t=1}^{N} I_t'\right)^2}} \tag{20}$$

where I_t and I_t' denote the predicted traffic flow series and N denotes the length of traffic flow time series.

3.4. Settings of Multi-Scale Wavelet Decomposition

For multi-scale wavelet decomposition, the wavelet basis function and decomposition levels are key parameters. Common wavelet basis functions include Haar, Daubechies, Symlets, Coiflets, Biorthogonal, and so on. In this paper, the Daubechies wavelet with a fourth-order vanishing moment (db4) and the Symlets wavelet with a fourth-order vanishing moment (sym4) are employed for the multi-scale wavelet decomposition. The sub-series of the M18 (0–5.2) dataset, decomposed using the db4 wavelet basis function with three levels, are shown in Figure 6.

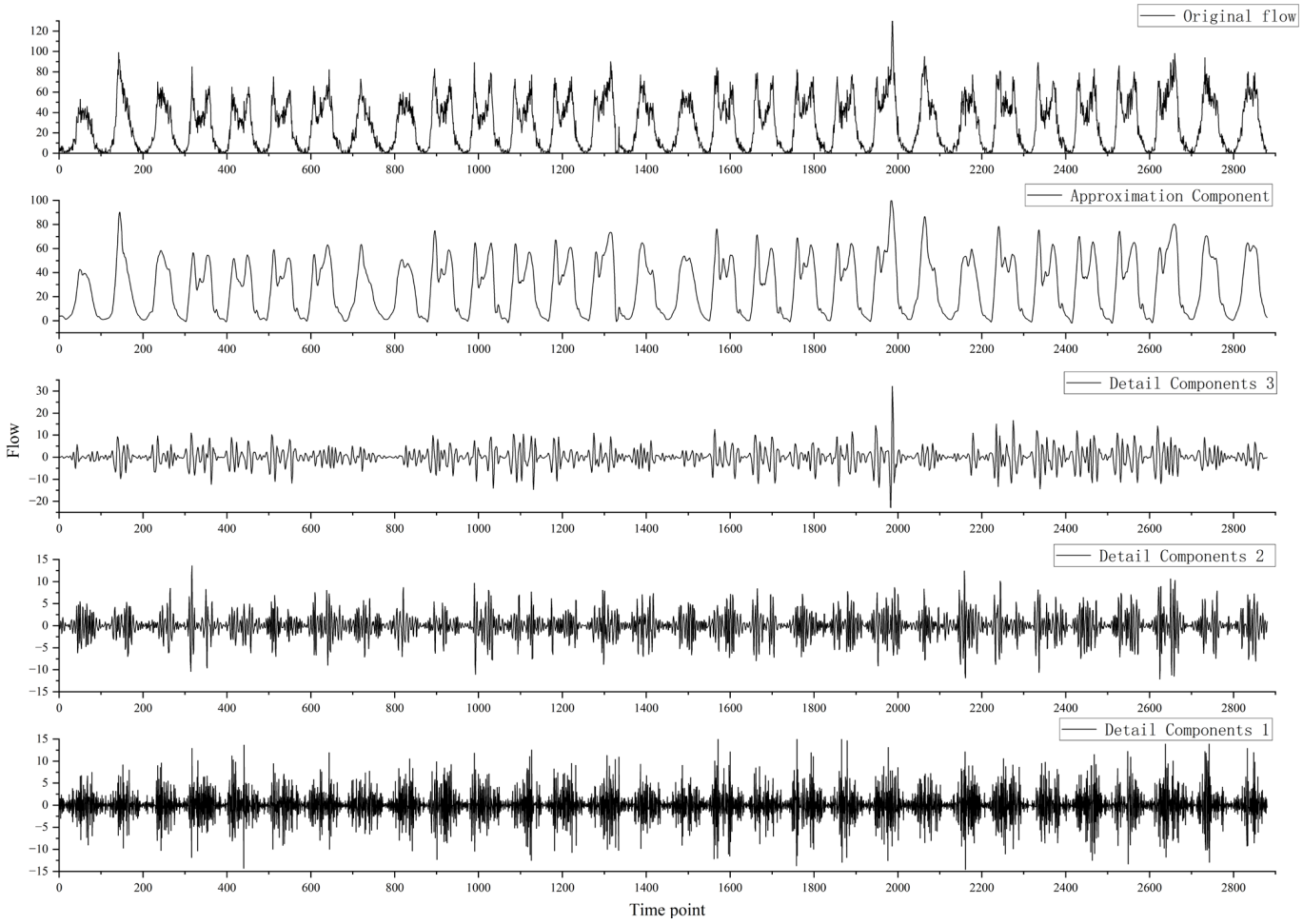


Figure 6. The 3-layer wavelet decomposition for M18 (0–5.2) dataset by the db4 wavelet basis function.

As shown in Figure 6, the original traffic flow time series has a certain periodicity, as well as some local high-frequency fluctuations. By using the wavelet decomposition, the approximation component and detail components are obtained, and it could be found that the approximation component is relatively smooth, has more obvious periodicity, and has no high-frequency fluctuations. This means that it is easier to model and predict the approximate component.

As for the detail components, they have a certain periodicity and also have some random fluctuations; as the decomposition scale increases, the frequency of the fluctuations gradually decreases. Most of the random noise and fluctuations are contained in low-detail-level components, and this means that it is more difficult to model and predict the detail approximate components than the approximate component. The lower the level of detail components, the greater the difficulty in prediction.

It is necessary to decompose the M18 (0–5.2) dataset using the db4 and sym4 wavelet basis functions, respectively; then, it is necessary to use the LNN model to train and predict each decomposed component separately. The results are shown in Table 2.

Table 2. Comparison of LNNs used on decomposed sub-series of M18 (0–5.2) dataset by db4 and sym4 wavelet basis functions.

Method	R^2 of db4 Wavelet Function	R^2 of sym4 Wavelet Function
Original flow	0.9105	0.9105
Approximation component (A3)	0.9975	0.9970
Detail component 3 (D3)	0.9517	0.9589
Detail component 2 (D2)	0.9343	0.9269
Detail component 1 (D1)	0.7552	0.5110

As shown in Table 2, the LNN model performs best on the approximation component, and performs worst in the detail component 1. For the detail component, the approximation effect of LNN model gradually improves with the increase in decomposition levels using both db4 and sym4 wavelet basis functions. For the approximation component, the LNN model has a very good approximation effect, with an R^2 index of over 0.997 under two wavelet basis functions. This means that setting the level of wavelet decomposition to 3 is appropriate. The R^2 index comparison of prediction results for each wavelet decomposed component under two wavelet basis functions is shown in Figure 7.

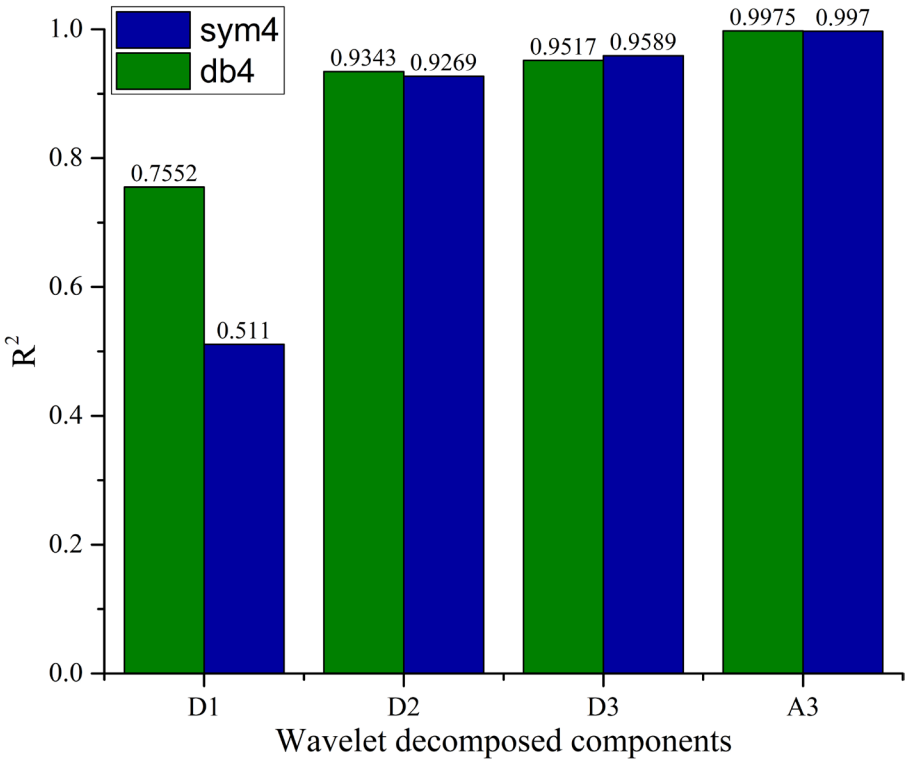


Figure 7. Histogram of prediction performance on decomposed sub-series of M18 (0–5.2) dataset using db4 and sym4 wavelet basis function.

As shown in Figure 7, the prediction accuracy for each decomposed component when using the db4 and sym4 wavelet basis functions is very close, except for detail component 1. As shown in Figure 6, detail component 1 exhibits small amplitude variation, and the prediction results for this component have a relatively small impact on the overall traffic flow. Therefore, the selection of wavelet basis functions has a relatively minor impact on the prediction results. Hence, considering the balance between prediction accuracy and

computational complexity, the decomposition level is set to 3, and the db4 wavelet basis function is chosen in this paper.

3.5. Results and Analysis

In order to verify the performance of proposed wavelet-LNN, experiments were conducted on four datasets. Then, the results were compared with those of original LNNs, benchmarked methods (including SVR [16], LSSVM [49], LSTM [50], TCN [51] and transformer [3]), and combined method (including wavelet-LSSVM and wavelet-LSTM). The results of R2, MSE, and MAE for each method are shown in Tables 3 and 4.

Table 3. Comparison of forecasting performance between wavelet-LNN and other baseline models on M18 dataset.

Method	M18 (0–5.2)			M18 (5.2–6.6)		
	R^2 ↑	MSE ↓ (Vehicle)	MAE ↓ (Vehicle)	R^2 ↑	MSE ↓ (Vehicle)	MAE ↓ (Vehicle)
SVR [16]	0.8949	68.254	6.3329	0.8914	54.5533	5.3725
LSSVM [49]	0.9031	62.9586	5.6119	0.8858	57.3758	5.0437
LSTM [50]	0.9058	61.2071	5.6254	0.9077	46.3674	4.5875
TCN [51]	0.8827	76.1518	3.5503	0.8848	57.8490	5.1027
Transformer [3]	0.8969	66.9624	5.8574	0.8873	63.6558	5.8198
EKF [52]	0.7997	130.0928	8.1349	0.7768	112.0995	7.2971
LinearRegression [52]	0.8997	65.1241	5.8392	0.8880	56.2343	5.2028
LNN	0.9107	58.0117	5.4603	0.9019	49.2557	4.7489
Wavelet-LSSVM	0.9528	30.6725	4.0256	0.9485	25.8742	3.5804
Wavelet-LSTM	0.9340	42.8779	4.7360	0.9329	33.7103	4.0539
Wavelet-LNN	0.9855	9.4203	2.1468	0.9825	8.7830	1.9323

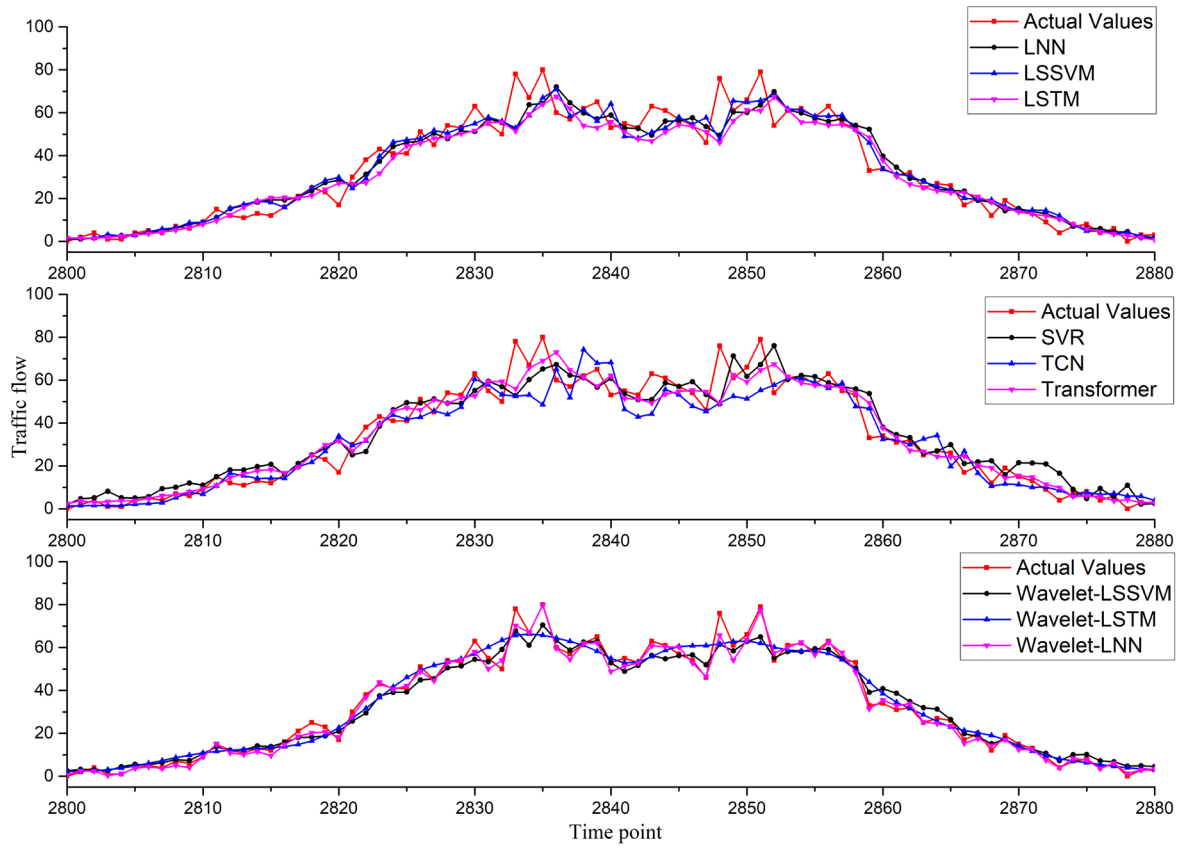
Table 4. Comparison of forecasting performance between wavelet-LNN and other baseline models on M25 dataset.

Method	M25 (0–5.2)			M25 (5.2–6.6)		
	R^2 ↑	MSE ↓ (Vehicle)	MAE ↓ (Vehicle)	R^2 ↑	MSE ↓ (Vehicle)	MAE ↓ (Vehicle)
SVR [16]	0.9428	8454.3086	74.5208	0.9203	248.9900	11.8717
LSSVM [49]	0.9554	6590.9204	58.4568	0.9228	241.3882	11.1793
LSTM [50]	0.9523	7047.5994	61.125	0.9130	271.7080	11.8218
TCN [51]	0.9485	7606.6784	62.4275	0.9274	226.8570	10.8716
Transformer [3]	0.9429	8438.7593	71.1325	0.9280	247.4395	11.2920
EKF [52]	0.9139	12,718.9128	77.8814	0.8267	541.5326	16.3594
LinearRegression [53]	0.9457	8021.4953	66.4183	0.8956	326.286	12.557
LNN	0.9528	6965.3570	60.2259	0.9053	295.7774	12.4653
Wavelet-LSSVM	0.9736	3900.8762	49.5527	0.9608	122.4717	8.2329
Wavelet-LSTM	0.9568	6379.8590	59.1426	0.9363	199.0141	10.3825
Wavelet-LNN	0.9915	1252.5928	25.8465	0.9856	45.1464	4.6492

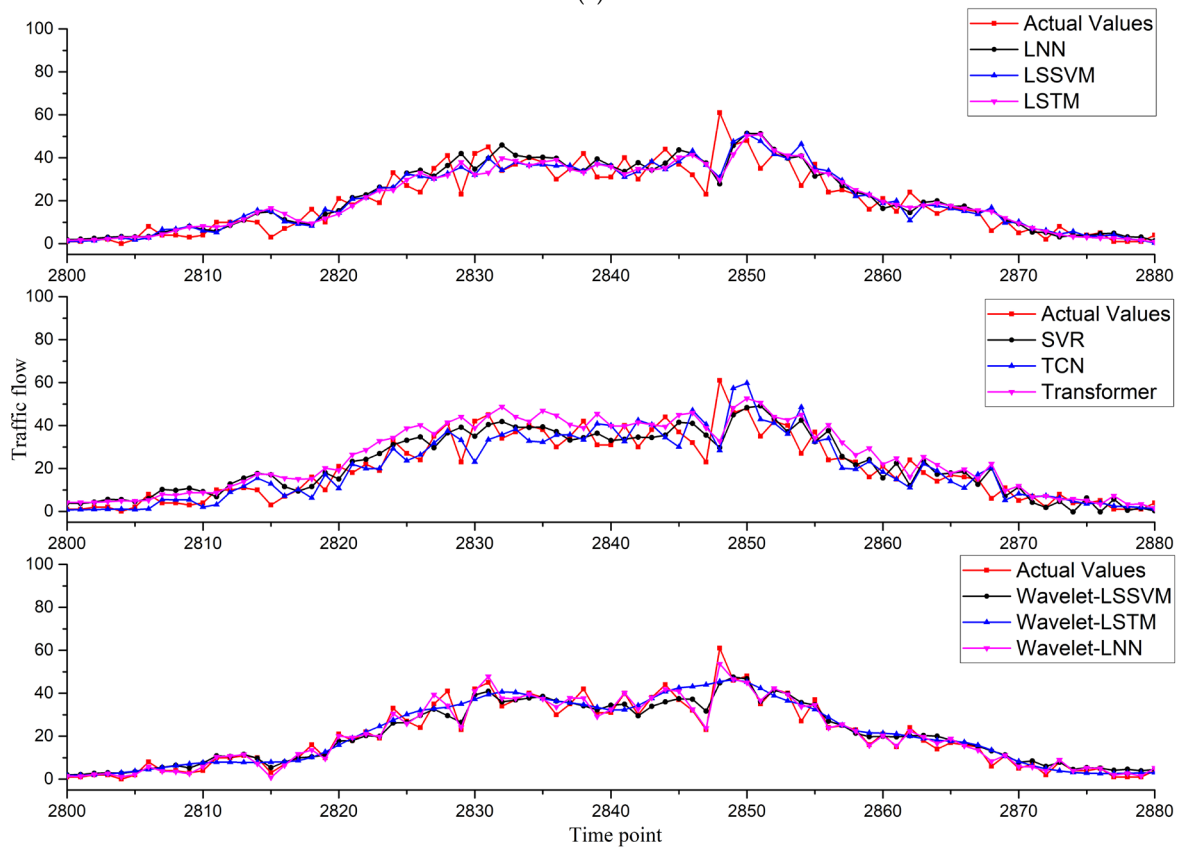
The ↑ denotes that the larger value is better and the ↓ denotes that the smaller value is better in Tables 3 and 4.

As shown in Tables 3 and 4, for the methods without wavelet decomposition, the LNN model performs best on the M18 (0–5.2) dataset, and the LSTM, LSSVM and transformer tend to be better than the LNN model when applied to the M18 (5.2–6.6), M25 (0–5.2) and M25 (5.2–6.6) datasets, respectively. Via combination with wavelet decomposition, the LSSVMs, LSTMs, and LNNs were all improved. The proposed wavelet-LNN achieves the best performance on all datasets. As for the LSSVM model, by using the wavelet decomposition, the wavelet-LSSVM also achieves good performance on all datasets. However, for the LSTM model, the wavelet decomposition did not bring significant improvement to the model.

To further illustrate the forecasting performance of the wavelet-LNN model and benchmarked methods, the values predicted from time point 2800 to 2880 on four datasets using each method are visualized and shown in Figure 8.

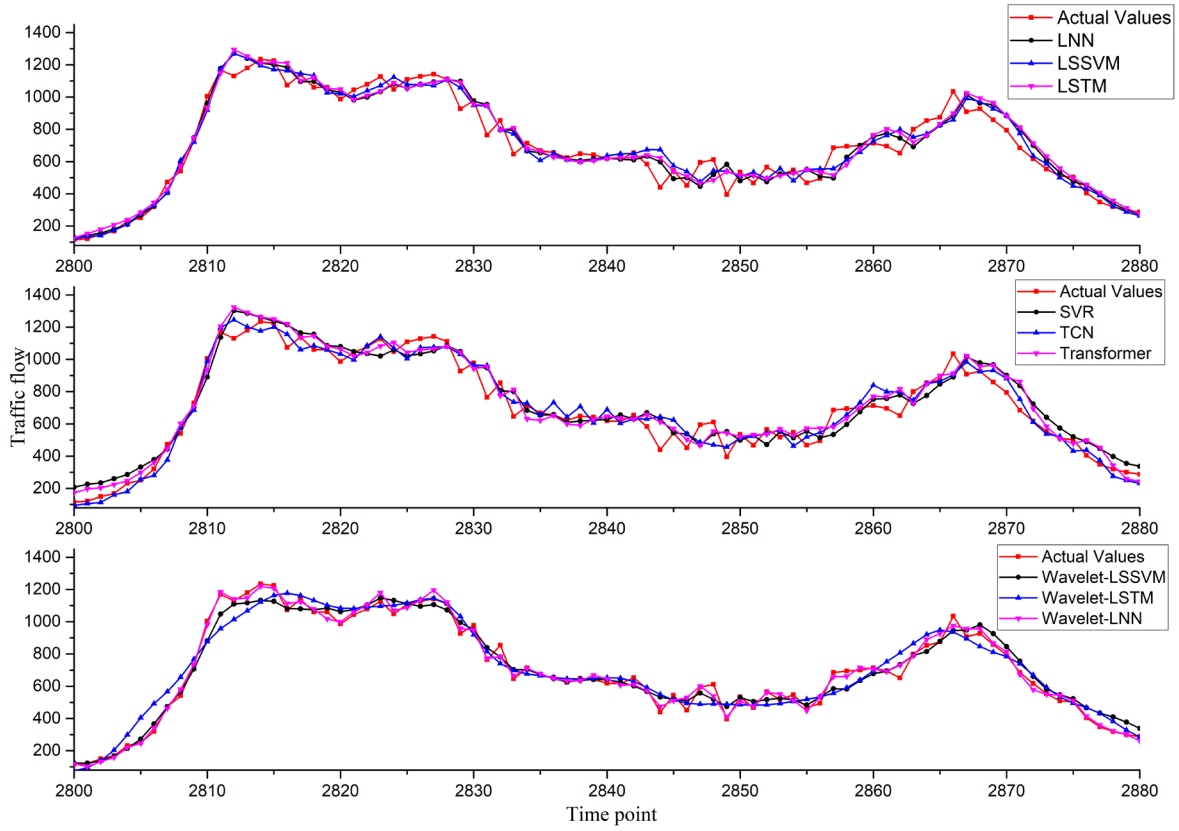


(**a**)

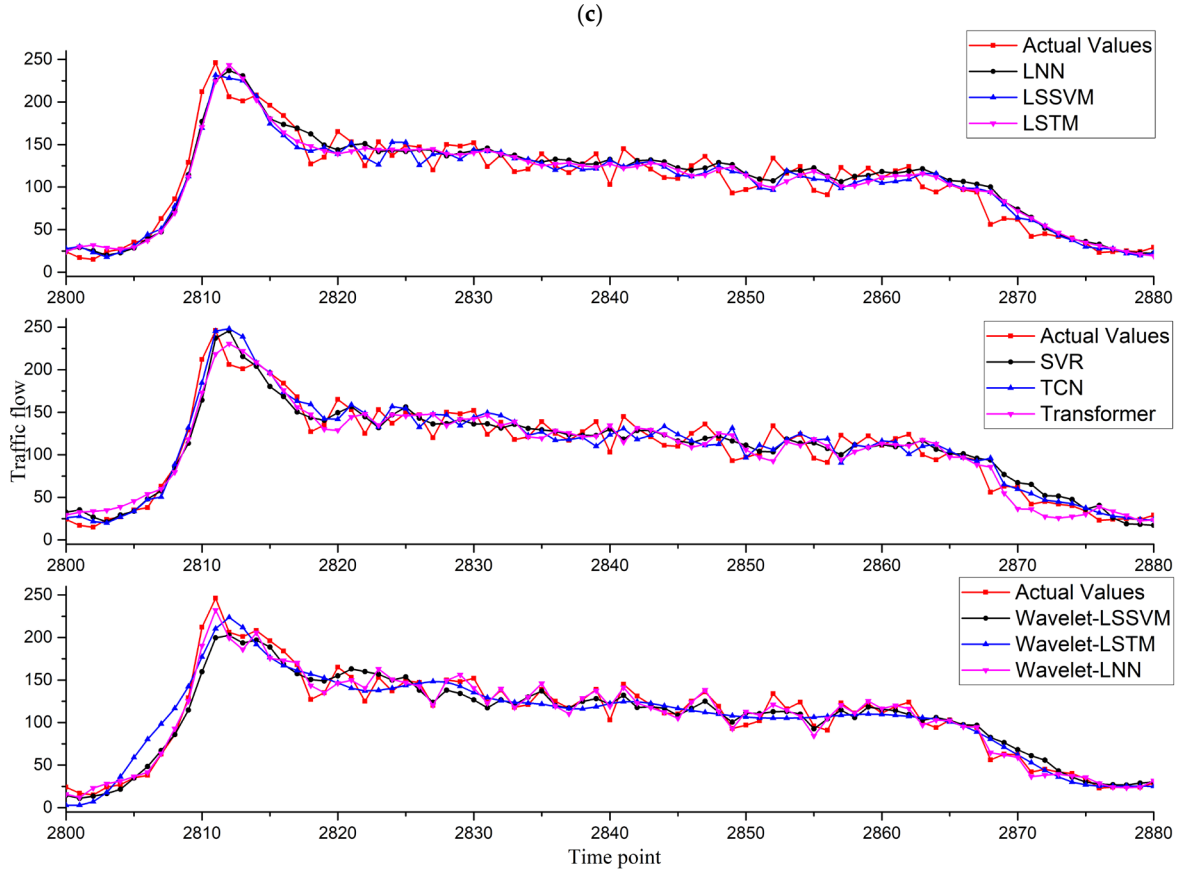


(**b**)

Figure 8. *Cont.*



(**c**)



(**d**)

Figure 8. Visualization of predicted values from time point 2800 to 2880 using different methods on four datasets: (**a**) predicted values for M18 (0–5.2) dataset, (**b**) predicted values for M18 (5.2–6.6) dataset, (**c**) predicted values for M25 (0–5.2) dataset, and (**d**) predicted values for M25 (5.2–6.6) dataset.

Based on the visualization of predicted values, we found that the proposed wavelet-LNN model achieves a good prediction performance on all four datasets. This means that the proposed wavelet-LNN model is better at capturing time-varying patterns of traffic flow than the benchmarked methods, and it is more resilient to noise and interference in the traffic flow series.

Moreover, in order to compare the efficiency and complexity of the models, Table 5 shows the total parameters of each model used.

Table 5. Comparison of total parameters wavelet-LNN and other baseline models.

Method	Total Parameters (Train)
SVR [16]	289
LSSVM [49]	4011
LSTM [50]	150,337
TCN [51]	828,097
Transformer [3]	26,881
EKF [52]	-
LinearRegression [53]	17
LNN	110
Wavelet-LSSVM	16,044
Wavelet-LSTM	601,348
Wavelet-LNN	440

As shown in Table 5, the total parameters of the LNN and proposed wavelet-LNN model are much smaller than those of other neural network models, being less than 1% of LSTM model and TCN model and about 2% of the transformer model. Although the total parameters of wavelet-LNN are small, the prediction performance is the best. This means that the proposed wavelet-LNN has more expressive power than neurons in conventional neural networks models.

4. Conclusions

This study proposes a wavelet-LNN model for highway traffic flow prediction by combining wavelet decomposition and the LNN model. By using the multi-scale wavelet decomposition algorithm to decompose the original traffic flow series, noise resistance is improved, and random fluctuation components are filtered out into the detail components. These detail components tend to have a relatively small impact on the overall traffic flow. For each decomposed component, an LNN model is trained for prediction, and the final predicted results are reconstructed by combining the predicted results of all components. Through experimental testing and verification on four traffic flow datasets, the following conclusions were drawn:

(1) The LNN model displays little difference in predicting the decomposition approximation components of different wavelet basis functions, and has good robustness in terms of approximation components.
(2) Wavelet decomposition can significantly improve the performance of LNN models and LSSVM models, but its improvement for LSTM is limited.
(3) The proposed wavelet-LNN model achieves the best performance on four different datasets and demonstrates good generalization performance.

Future research may delve into the application of deep reinforcement learning combined with multi-source data fusion methods in traffic volume prediction. By integrating real-time traffic flow data, weather information, and social media data, the timeliness and accuracy of the model could be further enhanced. Additionally, the development of cross-regional traffic volume prediction models would contribute to more refined traffic

management and optimization, promoting global coordination within intelligent transportation systems.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W. and H.K.; software, Y.W. and H.K.; validation, X.H. and J.C.; investigation, X.H.; writing—original draft preparation, Y.W.; supervision, S.Z. and W.W.; project administration, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Key Project of Chongqing Technical Innovation and Application Development (No. cstc2021jscx-gksbX0070, and the Scientific and Technological Research Program of Chongqing Municipal Education Commission (No. KJQN202100708), and the Natural Science Foundation of Chongqing (No. CSTB2022NSCQ-MSX0908).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset and source code generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: Authors Weipin Wang, Shuli Zhao, Xuening He, and Jingyao Chen were employed by Chongqing Expressway Network Management Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zou, G.; Lai, Z.; Wang, T.; Liu, Z.; Li, Y. MT-STNet: A Novel Multi-Task Spatiotemporal Network for Highway Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8221–8236. [CrossRef]
2. Chandra, S.R.; Al-Deek, H. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *J. Intell. Transp. Syst.* **2009**, *13*, 53–72. [CrossRef]
3. Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 50–64. [CrossRef]
4. Van Der Voort, M.; Dougherty, M.; Watson, S. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transp. Res. Part C Emerg. Technol.* **1996**, *4*, 307–318. [CrossRef]
5. Schimbinschi, F.; Moreira-Matias, L.; Nguyen, V.X.; Bailey, J. Topology-regularized universal vector autoregression for traffic forecasting in large urban areas. *Expert Syst. Appl.* **2017**, *82*, 301–316. [CrossRef]
6. Li, Z.; Yu, H.; Zhang, G.; Wang, J. A Bayesian vector autoregression-based data analytics approach to enable irregularly-spaced mixed-frequency traffic collision data imputation with missing values. *Transp. Res. Part C Emerg. Technol.* **2019**, *108*, 302–319. [CrossRef]
7. Cai, L.; Zhang, Z.; Yang, J.; Yu, Y.; Zhou, T.; Qin, J. A noise-immune Kalman filter for short-term traffic flow forecasting. *Phys. A Stat. Mech. Its Appl.* **2019**, *536*, 122601. [CrossRef]
8. Williams, B.M.; Hoel, L.A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J. Transp. Eng.* **2003**, *129*, 664–672. [CrossRef]
9. Wu, C.J.; Schreiter, T.; Horowitz, R.; Gomes, G. Traffic flow prediction using optimal autoregressive moving average with exogenous input-based predictors. *Transp. Res. Rec.* **2014**, *2421*, 125–132. [CrossRef]
10. Cai, P.; Wang, Y.; Lu, G.; Chen, P.; Ding, C.; Sun, J. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. Part C Emerg. Technol.* **2016**, *62*, 21–34. [CrossRef]
11. Qi, Y.; Ishak, S. A Hidden Markov Model for short term prediction of traffic conditions on freeways. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 95–111. [CrossRef]
12. Yang, S.; Wu, J.; Du, Y.; He, Y.; Chen, X. Ensemble learning for short-term traffic prediction based on gradient boosting machine. *J. Sens.* **2017**, *2017*, 7074143. [CrossRef]
13. Castro-Neto, M.; Jeong, Y.S.; Jeong, M.K.; Han, L.D. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [CrossRef]
14. Sun, S.; Zhang, C.; Yu, G. A Bayesian network approach to traffic flow forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. [CrossRef]
15. Lin, G.; Lin, A.; Gu, D. Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Inf. Sci.* **2022**, *608*, 517–531. [CrossRef]

16. Zhu, G.; Song, K.; Zhang, P.; Wang, L. A traffic flow state transition model for urban road network based on Hidden Markov Model. *Neurocomputing* **2016**, *214*, 567–574. [CrossRef]
17. Luo, C.; Huang, C.; Cao, J.; Lu, J.; Huang, W.; Guo, J.; Wei, Y. Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm. *Neural Process. Lett.* **2019**, *50*, 2305–2322. [CrossRef]
18. Tian, Y.; Zhang, K.; Li, J.; Lin, X.; Yang, B. LSTM-based traffic flow prediction with missing data. *Neurocomputing* **2018**, *318*, 297–305. [CrossRef]
19. Shu, W.; Cai, K.; Xiong, N.N. A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16654–16665. [CrossRef]
20. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Li, K.; Zeng, Z. Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **2020**, *14*, 42. [CrossRef]
21. Xing, L.; Liu, W. A data fusion powered bi-directional long short term memory model for predicting multi-lane short term traffic flow. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16810–16819. [CrossRef]
22. Ren, Y.; Zhao, D.; Luo, D.; Ma, H.; Duan, P. Global-local temporal convolutional network for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1578–1584. [CrossRef]
23. Fang, W.; Zhuo, W.; Yan, J.; Song, Y.; Jiang, D.; Zhou, T. Attention meets long short-term memory: A deep learning network for traffic flow forecasting. *Phys. A Stat. Mech. Its Appl.* **2022**, *587*, 126485. [CrossRef]
24. Naheliya, B.; Redhu, P.; Kumar, K. MFOA-Bi-LSTM: An optimized bidirectional long short-term memory model for short-term traffic flow prediction. *Phys. A Stat. Mech. Its Appl.* **2024**, *634*, 129448. [CrossRef]
25. Sun, X.; Chen, F.; Wang, Y.; Lin, X.; Ma, W. Short-term traffic flow prediction model based on a shared weight gate recurrent unit neural network. *Phys. A Stat. Mech. Its Appl.* **2023**, *618*, 128650. [CrossRef]
26. Ma, C.; Dai, G.; Zhou, J. Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5615–5624. [CrossRef]
27. Liu, Y.; Wu, C.; Wen, J.; Xiao, X.; Chen, Z. A grey convolutional neural network model for traffic flow prediction under traffic accidents. *Neurocomputing* **2022**, *500*, 761–775. [CrossRef]
28. Zheng, H.; Lin, F.; Feng, X.; Chen, Y. A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6910–6920. [CrossRef]
29. 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.* **2014**, *16*, 865–873. [CrossRef]
30. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [CrossRef]
31. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [CrossRef]
32. Zhao, L.; Zhou, Y.; Lu, H.; Fujita, H. Parallel computing method of deep belief networks and its application to traffic flow prediction. *Knowl. Based Syst.* **2019**, *163*, 972–987. [CrossRef]
33. Hu, H.X.; Hu, Q.; Tan, G.; Zhang, Y.; Lin, Z.Z. A Multi-Layer Model Based on Transformer and Deep Learning for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 443–451. [CrossRef]
34. Lv, M.; Hong, Z.; Chen, L.; Chen, T.; Zhu, T.; Ji, S. Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3337–3348. [CrossRef]
35. Qi, X.; Mei, G.; Tu, J.; Xi, N.; Piccialli, F. A deep learning approach for long-term traffic flow prediction with multifactor fusion using spatiotemporal graph convolutional network. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 8687–8700. [CrossRef]
36. Wen, Y.; Xu, P.; Li, Z.; Xu, W.; Wang, X. RPConvformer: A novel Transformer-based deep neural networks for traffic flow prediction. *Expert Syst. Appl.* **2023**, *218*, 119587. [CrossRef]
37. Liu, M.; Wang, W.; Hu, X.; Fu, Y.; Xu, F.; Miao, X. Multivariate long-time series traffic passenger flow prediction using causal convolutional sparse self-attention MTS-Informer. *Neural Comput. Appl.* **2023**, *35*, 24207–24223. [CrossRef]
38. Zheng, Y.; Wang, S.; Dong, C.; Li, W.; Zheng, W.; Yu, J. Urban road traffic flow prediction: A graph convolutional network embedded with wavelet decomposition and attention mechanism. *Phys. A Stat. Mech. Its Appl.* **2022**, *608*, 128274. [CrossRef]
39. Tian, Z. Approach for short-term traffic flow prediction based on empirical mode decomposition and combination model fusion. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 5566–5576. [CrossRef]
40. Yang, H.; Cheng, Y.; Li, G. A new traffic flow prediction model based on cosine similarity variational mode decomposition, extreme learning machine and iterative error compensation strategy. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105234. [CrossRef]
41. Hasani, R.; Lechner, M.; Amini, A.; Rus, D.; Grosu, R. Liquid time-constant networks. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 7657–7666. [CrossRef]
42. Chahine, M.; Hasani, R.; Kao, P.; Ray, A.; Shubert, R.; Lechner, M.; Rus, D. Robust flight navigation out of distribution with liquid neural networks. *Sci. Robot.* **2023**, *8*, eadc8892. [CrossRef] [PubMed]

43. Lechner, M.; Hasani, R.; Amini, A.; Henzinger, T.A.; Rus, D.; Grosu, R. Neural circuit policies enabling auditable autonomy. *Nat. Mach. Intell.* **2020**, *2*, 642–652. [CrossRef]
44. Karn, P.K.; Ardekani, I.; Abdulla, W.H. Generalized Framework for Liquid Neural Network upon Sequential and Non-Sequential Tasks. *Mathematics* **2024**, *12*, 2525. [CrossRef]
45. Hasani, R.; Lechner, M.; Amini, A.; Liebenwein, L.; Ray, A.; Tschaikowski, M.; Rus, D. Closed-form continuous-time neural networks. *Nat. Mach. Intell.* **2022**, *4*, 992–1003. [CrossRef]
46. Yu, H.; Li, T.; Yu, W.; Li, J.; Huang, Y.; Wang, L.; Liu, A. Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting. *arXiv* **2022**, arXiv:2210.06126.
47. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
48. Zhao, W.; Gao, Y.; Ji, T.; Wan, X.; Ye, F.; Bai, G. Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access* **2019**, *7*, 114496–114507. [CrossRef]
49. Cai, L.; Janowicz, K.; Mai, G.; Yan, B.; Zhu, R. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Trans. GIS* **2020**, *24*, 736–755. [CrossRef]
50. Bakibillah, A.S.M.; Tan, Y.H.; Loo, J.Y.; Tan, C.P.; Kamal, M.A.S.; Pu, Z. Robust estimation of traffic density with missing data using an adaptive-R extended Kalman filter. *Appl. Math. Comput.* **2022**, *421*, 126915. [CrossRef]
51. Apronti, D.; Ksaibati, K.; Gerow, K.; Hepner, J.J. Estimating traffic volume on Wyoming low volume roads using linear and logistic regression methods. *J. Traffic Transp. Eng.* **2016**, *3*, 493–506. [CrossRef]
52. Zhou, T.; Jiang, D.; Lin, Z.; Han, G.; Xu, X.; Qin, J. Hybrid dual Kalman filtering model for short-term traffic flow forecasting. *IET Intell. Transp. Syst.* **2019**, *13*, 1023–1032. [CrossRef]
53. Chang, S.Y.; Wu, H.C.; Kao, Y.C. Tensor extended Kalman filter and its application to traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 13813–13829. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.