Short-Term Traffic Flow Prediction Based on CNN-BILSTM with Multicomponent Information

Weiqing Zhuang 1,* and Yongbo Cao 2

1 School of Internet Economics and Business, Fujian University of Technology, Fuzhou 350014, China
2 School of Transportation, Fujian University of Technology, Fuzhou 350108, China
* Correspondence: zhuangweiqing@fjut.edu.cn

Abstract: Problem definition: The intelligent transportation system (ITS) plays a vital role in the construction of smart cities. For the past few years, traffic flow prediction has been a hot study topic in the field of transportation. Facing the rapid increase in the amount of traffic information, finding out how to use dynamic traffic information to accurately predict its flow has become a challenge. Methodology: Thus, to figure out this issue, this study put forward a multistep prediction model based on a convolutional neural network and bidirectional long short-term memory (BILSTM) model. The spatial characteristics of traffic data were considered as input of the BILSTM model to extract the time series characteristics of the traffic. Results: The experimental results validated that the BILSTM model improved the prediction accuracy in comparison to the support vector regression and gated recurring unit models. Furthermore, the proposed model was comparatively analyzed in terms of mean absolute error, mean absolute percentage error, and root mean square error, which were reduced by 30.4%, 32.2%, and 39.6%, respectively. Managerial implications: Our study provides useful insights into predicting the short-term traffic flow on highways and will improve the management of traffic flow optimization.

Keywords: traffic flow forecast; convolutional neural network; bidirectional long short-term memory network model; multicomponent information

1. Introduction

Road safety and traffic control have been paid extensive attention to by researchers, owing to the rapid development of the social economy. A marked increase in the number of vehicles has resulted in a serious imbalance between traffic supply and demand, causing growing serious issues in traffic congestion and information lag, which consequently intensifies the social and economic burden. The essential problem of ITS is to achieve traffic guidance and control, and in the scientific control of the transportation system, the realization of real-time prediction of short-term traffic flow is a prerequisite [1]. Therefore, predicting short-term traffic flow in real time, and accurately, is of great significance for the effective management of transportation departments, as well as daily commuters. Owing to the limited location space, traffic congestion at any instant cannot possibly be reduced by only expanding roads and infrastructure. Real-time accurate prediction would assist policymakers, administrators, commuters, and travelers toward effective organization, such that managers could provide basic information regarding roads in the future to transfer the traffic pressure on congested roads [2]. However, the arrival of the era of big data makes the traffic situation more and more complicated. Owing to the randomness, nonlinearity, and dynamic nature of traffic data, traditional predictive analysis models pose various limitations, which raises challenges for short-term traffic flow prediction.

Previous short-term traffic flow prediction models can be roughly divided into three categories: parametric models, nonparametric models, and hybrid models [3]. Among them, the time series methods and nonlinear algorithms employ parametric models, which
are mainly represented by the autoregressive integrated moving average (ARIMA) model and its varieties, e.g., Van et al. [4] used the hexagonal Kohonen self-organizing map as a classifier and stored a separate ARIMA model in each class for simplified adjustment. The separation of function approximation and classification display improved the prediction performance of the model. The seasonal autoregressive integrated moving average model (SARIMA) proposed by the literature [5,6] uses the autocorrelation function to determine the order of the model after differencing the input time series data, and it uses the maximum likelihood estimate to determine the hyper-parameters of the model and substitute the experimental data for prediction and comparison. In addition, the Kalman filtering [7,8] algorithm is often used for traffic flow forecasts. Nevertheless, due to the strong randomness characteristics of traffic flow data, big data training must be applied to ensure a certain accuracy, and a simple parametric model cannot be perfectly trained with big data, which would probably produce certain errors. To resolve these problems, nonparametric models should be avoided. For instance, as mentioned in the literature [9], the K-nearest neighbor (KNN) method extracts temporal features by using multistep time steps in the time series, and it inputs the extracted features into the KNN model for prediction. In view of the universality of research traffic, the literature [10] proposes traffic forecasting under accident conditions and atypical situations; that is, the data are classified into severe weather, weekends, holidays, and traffic accidents for input, and an online regression support vector machine model is used (OL-SVR) for the supervised learning of traffic features under different classifications. Although these models have improved performance compared to time series models, they also have other problems. The KNN model in the literature [9] lacks the extraction of spatial features of the data. Although the literature [10] makes traffic prediction more universal by predicting data under different classifications, it ignores the temporal correlation characteristics of the data itself and the spatial correlation between roads or between different detection points, as well as other complex characteristics, thus lacking in prediction performance. With the recent developments in deep learning, the deep learning models used to study traffic flow prediction are diverse, e.g., the deep belief net (DBN), stacked autoencoder (SAE), convolutional neural network (CNN), and long short-term memory (LSTM) network. Generally, these simple neural network models can be applied to obtain only a single traffic data characteristic—CNN extracts spatial features, whereas the LSTM model extracts temporal features. Therefore, they fail to completely capture the complex characteristics of traffic. Although several studies have developed hybrid models, most of them estimated the predictions from individualistic traffic flow data and independent processing.

To address these issues and obtain real-time, accurate traffic predictions, this paper proposes a combined CNN-BILSTM model that integrates flow and speed characteristics for short-term traffic flow forecasts. The major contributions of this paper are stated below:

1. We propose a CNN-BILSTM hybrid model for predicting traffic flow by extracting the spatial features via CNN and the temporal features via the BILSTM model.
2. As the majority of existing research is limited to predicting traffic flow only, neglecting the influence of other factors, in this study, when predicting traffic flow, the influence of traffic flow and speed on traffic flow were considered at the same time.
3. Due to the unique periodic nature of traffic flow data, this research established two cyclic components—weekly and daily—to predict the traffic flow.

The main research purpose of this study was to predict traffic flow by proposing a CNN-BILSTM model. First, through data preprocessing, the processed data were used as input into the CNN model to extract spatial features, then the BILSTM model was used to extract temporal features, and finally, the model was tested by substituting the dataset to predict its performance. The rest of the paper is structured as follows: Section 2 mainly reviews the literature on this research direction, at home and abroad, in recent years; Section 3 describes the algorithm framework and algorithm steps proposed in this paper in detail; in Section 4, the data and experimental parameters are explained in detail, as well as the selection of evaluation indicators; Section 5 is the specific analysis
of the experimental results. By setting different components and different time steps, the experimental prediction results are analyzed in detail; Section 6 is the conclusion and outlook.

2. Literature Review

In this section, we primarily discuss the existing literature on parametric, nonparametric, and hybrid models that have been applied to traffic flow forecasting.

In a general way, the parameter model based on time series prediction is widely used. Jan F et al. [11] applied the time series model to short-term electricity price forecasting. By proposing a P-order function autoregressive model, the final prediction error (FFPE) in the function was used to independently select the dimension and lag structure of the model, the example data were used for prediction evaluation, and the results showed that the method has superiority in prediction performance. Jan F et al. [12] further predicted short-term electricity demand by proposing function autoregression (FAR), FAR with exogenous variables (FARX) and the classical univariate AR model, and further verified the data through Nord Pool electricity market data. The FARX model is more superior in training prediction. In 1979, Ahmed et al. [13] first proposed the application of the ARIMA model to study expressway traffic flow. Subsequently, numerous variant models have been put forward to heighten the prediction ability of the ARIMA model. In this context, Van et al. [4] came up with a hybrid model integrating the Kohonen method and ARIMA method to forecast short-term traffic flow, which enhanced prediction accuracy by simplifying the problem of class definition. Yu et al. [14] developed a comprehensive and multiplicative seasonal ARIMA model for forecasting the traffic sequence based on its characteristics. Thereafter, Kumar et al. [6] advanced a model focused on the data available to satisfy the requirement of the prior models in terms of a sound database. Accordingly, they proposed a seasonal ARIMA (SARIMA) model to resolve this limitation. Duan et al. [15] designed a unified spatiotemporal model on account of spatiotemporal ARIMA (STARIMA). The prediction period parameters of the model can be conveniently modified to adapt to the varying complex road structure, and the spatiotemporal characteristics between roads can be captured to improve the degree of the preciseness of traffic flow forecasts. The above-mentioned variant models resolved the problems that could not be solved in the previous ARIMA models, such as the low prediction accuracy caused by the inability to manipulate nonlinear data and the randomness of traffic data. However, both the ARIMA model and the variant model have the same problem; that is, the structure of the model itself is relatively simple, which is not suitable for processing complex data, and the model is more for processing linear data, while traffic flow data itself has complex spatial characteristics, and the time dependence is nonlinear and has large randomness and periodicity. Therefore, a more complex model structure is required for training, and the data characteristics of traffic flow are deeply excavated.

In case of a surge in traffic flow data, nonparametric methods based on an extensive amount of data can more effectively train the data, and shallow machine learning models can capture the key features of traffic. Zhang et al. [16] established the KNN model through three aspects: temporal data, parameters, and a prediction mechanism. First, by standardizing the initial data, the magnitude difference between the data samples was reduced to improve the prediction accuracy. Using the traffic flow data of the fast road section in Shanghai, and comparing and analyzing different model algorithms, the results showed how the KNN model proposed in this paper could be used. In regards to its feasibility in predicting short-term traffic flow, however, the spatiotemporal correlation of traffic flow was seriously neglected in this study. Xu et al. [17] first extracted the data of several typical road traffic states, then constructed a running feature sequence between the roads, constructed the kernel function of the time series to select the nearest road traffic states for prediction, and then used a selection of actual traffic data in Beijing to predict the traffic flow. The kernel function of the KNN model proposed in this paper is feasible and has high accuracy. Through analysis, it was found that when extracting the spatial features
of the data, this study simply used the connection between the roads as the spatial feature and did not deeply mine the spatial features of the data itself, so it could be improved in this aspect. Cheng et al. [18] proposed an adaptive K-nearest neighbor model (adaptive STKNN) for traffic flow prediction, given the fixedness and uncertainty of spatiotemporal dependence in current traffic flow prediction models. Firstly, the size of the spatial field was calculated by the cross-correlation function to extract the spatial dependence between the roads. Secondly, the autocorrelation function was used to determine the time window length of the time series. Finally, the space-time weight was introduced into the distance function to optimize the search mechanism of KNN. The dataset of highways in California, USA was verified, and the comparison with the four traditional models showed that the modified model has better prediction performance. A key contribution of this study is showing the spatial heterogeneity of urban transportation networks, that is, the spatial dependence of traffic flow data. However, the problems existing in the KNN model itself cannot be ignored. First, the calculation amount of KNN is relatively large. When the amount of data and feature points increases, the KNN algorithm is unsuitable. Secondly, the interpretability is not strong. During the screening, due to its characteristics, it selects the features with many similar surrounding features, forcibly assimilates other data features, and cannot fully mine the unique features of the data itself. Castro-Neto [10] proposed an online regression support vector machine (OL-SVR) model and employed supervised learning techniques to forecast traffic flow under atypical and typical circumstances, which attained suitable results.

Recently, various methods for deep learning have been commonly used for traffic flow prediction, and deep learning is advantageous in autonomously learning the complexity, randomness, and hierarchical features of traffic data [19]. Initially, Hinton et al. [20,21] advanced a fast-learning algorithm on account of DBN and further improved it. Based on this, Huang et al. [22] pioneered the application of DBN in short-term traffic flow forecasting and categorized the deep structure into the DBN at the bottom layer, which performs unsupervised learning of traffic flow features, as well as the multitask regression layer at the top that performs supervised prediction. Using large amounts of unlabeled data to learn features, labeled data for supervised adjustment, and then training on real datasets, the results showed the promise of deep learning in the transportation domain. However, although the DBN network shows greater advantages in processing complex data, because there is no connection between units in the network, the learning process is slow in the training process, and improper parameter selection can easily lead to local maximum superior issues. Moreover, Jin et al. [23] advanced an improved SAE on account of the greedy layer-by-layer training method. By stacking the autoencoder layer by layer, the data are input into the autoencoder for encoding and decoding, so that the dimensions of the input data and the output data are consistent, the data features are preserved to the greatest extent, and redundant information is reduced. Its structure is similar to that of DBN, both of which are unsupervised learning by stacking network layers. Zhang et al. [24] determined the spatiotemporal characteristics of traffic flow data using a selection algorithm and converted the data into a two-dimensional matrix to serve as an input for the CNN model for feature learning and prediction. Liu et al. [25] combined CNN and LSTM to extract the spatiotemporal information of the traffic flow and utilized the BILSTM model to analyze the historical traffic flow of the target detection point for obtaining periodic features. When the CNN or LSTM model is used to process the data alone, the characteristics of the CNN model determine its advantages in processing the spatial characteristics of the data. The LSTM model has advantages in processing short-term time series data. Facing long-term time series data, LSTM cannot fully exploit the impact of future data on model training, because it can only perform forward propagation. As CNN has been expanded to the graph convolutional neural network (GCN), numerous types of research have been conducted for learning the characteristics of traffic flow graph structure data on account of spectrogram theory [26]. Zhang et al. [27] came up with a hybrid graph convolutional network (HGCN), combined with weather conditions, to forecast the traffic flow of highway detection points,
and Li et al. [28] advanced an LSTM network model on account of graph and attention mechanisms, using GCN to capture the spatial features of traffic flow, employing the LSTM model to extract temporal features, assigning weights to the local correlation size of temporal and spatial features through an attention mechanism, and achieving better performance results in training on actual datasets. Although the GCN model has been the frontier in recent years, it was not suitable for the data prediction of this study, because the location distribution of the data detection points in this study was relatively regular, and it is not suitable for graph structure processing; that is, it does not need to use the Laplace matrix. The adjacency matrix is used to express the direct connection relationship between nodes, so as not to increase the complexity of the model.

Compared to the existing traffic flow prediction models, this study uniquely came up with a new CNN and BILSTM hybrid model to forecast traffic flow with the establishment of matrices describing the traffic flow and speed, respectively. In addition, we applied weighted fusion to obtain the periodic law of the traffic flow data based on the training and prediction of weekly and daily periodic data.

3. Methods

3.1. CNN Model

The convolutional neural network (CNN), as a kind of artificial neural network, first emerged from research on the cat visual system, wherein convolutional layers were constructed by introducing receptive field features into the neural network. The preliminary model of the CNN is LeNet, which delivers a high training speed with accuracy [29]. In the traffic field, the traffic flow of the target station is in connection with the adjacent station space, as well as the traffic flow and congestion of the upstream and downstream stations. Therefore, the CNN model could be adopted in this study to draw the spatial characteristics of the traffic monitoring points. Overall, CNN contains convolutional layers, activation layers, pooling layers, and fully connected layers.

3.1.1. Convolutional Layer

The convolution layer occupies a central position in convolutional neural networks, which performs the convolution operation via the convolution kernel. In particular, its essence is the discrete convolution operation of two two-dimensional matrices. To predict traffic flow, the convolution kernel extracts additional traffic data features to yield the output, and the activation function of each convolution layer processes the nonlinear mapping relation between the output and input of the traffic flow, which serves as the subsequent input of the convolution layer. Comprehensively, its operation function is as follows:

\[
G(x, y) = \sum_u \sum_v M(u, v) N(x - u, y - v)
\]

where \(m\) denotes the size of the convolution kernel, \(M\) represents the parameter matrix of the convolution kernel, and \(u\) and \(v\) indicate the step size of the convolution kernel in the input data dimension \((x, y)\).

3.1.2. Activation Layer

In the CNN model, the primary role of the activation function is to process nonlinear data and extract data features through nonlinear mapping, thereby improving the expressiveness of the network. The commonly used activation functions are sigmoid, tanh, and rectified linear unit (ReLU), which are respectively expressed as:

\[
\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}
\]

\[
\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\]
ReLU = \begin{cases} 
\max(0, x) & x > 0 \\
0 & x < 0 
\end{cases} 
(4)

In this study, the ReLU function was added between the BN algorithm layer and the pooling layer to develop the nonlinear processing capability of the model.

### 3.1.3. Pooling Layer

The pooling layer is also known as the downsampling layer. In principle, the maximum value in a specific region is selected as the value of a new region by applying the size of the convolution kernel as the step, which reduces the number of parameters and the computational complexity to a certain extent; this reduces overfitting as much as possible.

### 3.1.4. Fully Connected Layer

The fully connected layer is connected to every neuron of all previous layers, which weights the previous features, differentiates and integrates the local category information in the convolutional layer and the pooling layer, and expands it into a one-dimensional vector form to ensure that the front role of the distributed features learned by other layers is mapped onto the sample label space. The following equation was applied:

\[
z_j^{m+1} = \sum_{i=1}^{n} W_{ij}^m a_j^m + b_j^m 
\]

(5)

where \( m \) denotes the number of layers, \( i \) and \( j \) indicate the number of neurons, \( z \) denotes the output value, \( w \) represents the weight, and \( b \) represents the offset.

This research designed a CNN prediction model based on its data characteristics. First, the data features were inputted, the time steps of the series were convoluted using the sequence folding layer, and a convolution layer was connected with the BN (batch normalization layer) algorithm layer in the feature extraction to normalize the BN algorithm. This ensured the improvement in network generalization ability, acceleration of the convergence process, and transfer of the normalized data to the lower sampling layer. Moreover, all the nodes of the subsampling layer were extracted into a feature vector, and the sequence structure of the input data was recovered through the sequence unfolding layer to input the data into the BILSTM network structure for feature learning and prediction.

### 3.2. BILSTM Model

#### 3.2.1. LSTM Model

The randomness and dynamics of traffic flow data suggest that short-term traffic flows exhibit stronger correlations. In general, traffic information displays a stronger long-term dependence in terms of temporal features, as a certain time interval is required between the traffic flow passing through the adjacent detection points, especially in the case of traffic congestion. The basic recurrent neural network (RNN) utilizes the output of the preceding neuron as the input of the subsequent neuron during time series processing, such that the RNN contains a short-term memory function to process traffic data information. However, it cannot perform memory processing for long-term dependencies, namely the vanishing and exploding gradient problem. To resolve this issue, the LSTM model with a gate structure provides a new method [30]. Similar to the RNN model, LSTM is a chain structure composed of repeating unit cells. Nonetheless, the difference between the LSTM and RNN is that the LSTM cell contains four layers inside, instead of the single-layered unit construction of the RNN. The specific structure of the LSTM is illustrated in Figure 1. In principle, the LSTM selects and controls the information flow using the input gate, forget gate, and output gate, including filters that save or delete the information as required. The fundamental equations of the evaluation procedure are stated as follows:

\[
G_f = \sigma(W_f \cdot [T_{t-1}, I_t] + b_f) 
\]

(6)
where $G_f, G_i,$ and $G_o$ represent the forget gate, input gate, and output gate, respectively; $W$ denotes the weight, $b$ represents the bias term, $T_{t-1}$ refers to the input of the previous unit, $S_{t-1}$ denotes the state of the previous memory unit, and $\sigma$ represents the sigmoid function.

Figure 1. LSTM model structure diagram.

3.2.2. BILSTM Model

During the processing of historical data, the single-layer LSTM model evaluates the prediction of the succeeding instance. Accordingly, the output value is related only to the information of the previous instance. In the actual traffic information, the target value to be predicted is not only related to the information of the previous instance, but also influenced by that of the future instance. Therefore, we executed the BILSTM model for prediction. The BILSTM structure comprises two unidirectional LSTM, stacked up and down. The input at time $t$ is predicted based on the forward and reverse propagation, and the output is jointly determined by the two LSTMs [31], which improves the stability of the network during the processing of the short-term traffic flow time series in both directions. The structure diagram is illustrated in Figure 2.
where $Y(t)$

Set the time step window length to 7, add 1 each time, and then traverse all the data.

the historical traffic flow and speed data matrix. Let $X_t$ denote the traffic flow through the fully connected layer.

extracted the temporal features for the input of BILSTM, and, productively, forecast the

CNN-BILSTM model selected the spatial features of the traffic flow data through CNN, correlation between traffic flow and speed for obtaining a more precise forecast. The proposed

3.3. CNN-BILSTM Model

Prior to introducing the model algorithm process in detail, we initially constructed

detection points, as follows:

Prior to introducing the model algorithm process in detail, we initially constructed

$Y(t)$ can be expressed as

$$
\begin{bmatrix}
Y_{m,t-n}^o \\
Y_{m,t-n+1}^o \\
\vdots \\
Y_m^o \\
\end{bmatrix}^T =
\begin{bmatrix}
x_{m,t-n}^o \\
x_{m,t-n+1}^o \\
\vdots \\
x_m^o \\
\end{bmatrix}
$$

Figure 2. BILSTM model structure diagram.

Given the above analysis of the CNN and BILSTM models, we considered the correlation between traffic flow and speed for obtaining a more precise forecast. The proposed CNN-BILSTM model selected the spatial features of the traffic flow data through CNN, extracted the temporal features for the input of BILSTM, and, productively, forecast the traffic flow through the fully connected layer.

The specific prediction algorithm steps of the CNN-BILSTM model structured in this paper are stated below:

1. Preprocess the traffic flow data as well as speed data, fill in missing values, and delete outliers.
2. Substitute the data into the model and classify the data, of which 90% is used for training, and the remainder is used as the test data. Thereafter, the data are normalized.
3. Set the time step window length to 7, add 1 each time, and then traverse all the data.
4. The processed data serve as the input into the CNN model to extract spatial characteristics.
(5) The traffic flow spatial feature data output by the CNN network is used as the input of the BILSTM network model to further extract the spatial features of the data.  

(6) Define the super parameters of the model for network training prediction.  

The general framework of the CNN−BILSTM model is shown in Figure 3:

![Figure 3. CNN−BILSTM algorithm framework diagram.](image)

4. Experiment

4.1. Experimental Environment

The computer configuration used in this study was 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30 GHz, GPU: NVIDIA GeForce RTX 3060 Laptop GPU, computation capability: '8.6', and RAM: 16 GB. The operation environment was created in MATLAB R2021b.

4.2. Parameter Settings

Based on multiple experiments, the CNN model was stacked with the convolution layer, BN layer, pooling layer, expansion layer, and smoothing layer. In addition, the BILSTM model was set as a two-layer BILSTM layer, and the number of hidden units was set to 128, with 32 dropout units. The layer was set to 0.25, and finally, the output layer dimension was adjusted with a fully connected layer. The parameters were set to optimize the Adam function through 100 iterations of model training, with a batch size of 2400 and an initial learning rate of 0.005. The learning rate decreasing period was set at 20, and the learning rate was adjusted with a weight factor of 0.8. After adjustment, the learning rate
was evaluated as the product of the current learning rate and weight factor. Moreover, the dropout value was set to 0.2, and L2 regularization was employed to prevent overfitting.

4.3. Data Collection

To test and verify the effect of the model advanced in this study, we employed the dataset provided by PEMS, which is a public highway database in California, USA, for training evaluation. The database contains real-time and historical data of a total of 47,000 detection points in California. Among which, 15 detection points in the I10-E7 area of the highway in Los Angeles, California were chosen as the research targets, and the corresponding positions are presented in Figure 4. In particular, the data from May 1 to July 31, 2021 were regarded as the experimental data. The data were recorded in intervals of 5 min, producing a total of $288 \times 92 \times 15 \times 2$ pieces of sample data. The flow data of the last 1000 samples are displayed on the left-hand side of Figure 5, and the traffic flow chart from June 7 to June 11, 2021 is presented on the right-hand side. As observed, the characteristics of the traffic flow were strongly periodic, and the total traffic flow data described a similar trend.

![Detection point location map](image1)

**Figure 4.** Detection point location map.

![Traffic flow map](image2)

**Figure 5.** Traffic flow map.
4.4. Performance Index

To determine the prediction performance of the proposed model, we evaluated the prediction results based on three evaluation indicators, that is, mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE), which are expressed as below:

\[ MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i| \]  
\[ MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \]  
\[ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \]

where \( n \) represents the number of samples, \( y_i \) represents the true value of the data, and \( \hat{y}_i \) represents the predicted value of the data sample.

5. Results Analysis

The proposed model structure was trained on the experimental environment and parameter setting adopted in the experiment, as discussed in Section 4. The traffic flow and speed data were inputted into the CNN to pick up the spatial characteristics, and BILSTM was inputted to pick up the temporal characteristics. The predicted results are presented on the left-hand side of Figure 6. As observed from the figure, the actual traffic flow value fitted appropriately with the predicted value, and the overall flow trend was consistent with an evident prediction effect. Comparatively, the prediction error is charted on the right-hand side of the figure, which indicates an extremely small predicted flow error, which completely proves the accuracy and feasibility of the proposed model.

Figure 6. Forecast results and errors.

Owing to the strong periodicity of traffic data and to obtain a more comprehensive understanding of the model, we further processed the data and trained the model by categorizing the data into two major components: weekly and daily cycles. Since the traffic flow in the predicted time period was often strongly correlated with its adjacent instance or the same instance, but not with the time in a distant period, we selected the traffic flow at the same instance to forecast the traffic flow in the target time period.

First, we set the current time as \( t \) and the predicted time step as \( T_p = 2 \) h. These two time segments of the daily and weekly cycles were established as \( T_d \) and \( T_w \), respectively,
which were integer multiples of $T_p$. Correspondingly, the traffic fragments were intercepted to establish the correlation matrix, where the daily cycle segment can be stated as follows:

$$F_d = \left( f_{1-T_d/T_p} \times m+1 \cdots, f_{1-T_d/T_p} \times m+T_p, f_{1-T_d/T_p} \times m+1 \cdots, f_{1-m+1} \cdots, f_{1-m+T_p} \right)$$  \hspace{1cm} (17)

Similarly, the weekly cycle segment can be expressed as follows:

$$F_d = \left( f_{1-7 \times (T_d/T_p) \times m+1 \cdots}, f_{1-7 \times (T_d/T_p) \times m+T_p}, f_{1-7 \times (T_d/T_p) \times m+1 \cdots}, \cdots f_{1-7 \times m+1 \cdots}, f_{1-7 \times m+T_p} \right)$$  \hspace{1cm} (18)

Thereafter, the processed data were substituted into the model, and the data volume of 24 samples was predicted after two hours. The prediction results are illustrated in Figure 7, which further verifies the accuracy of the model and the periodicity of traffic flow.

![Weekly component test results and real values](image)

**Figure 7.** Weekly and daily forecasts.

To verify the effect of time on model prediction, this study further classified the time intervals. Upon assuming that $n$ denotes a time step, we set $n = 1, 3, \text{ and } 6$ for training, i.e., the corresponding time intervals were 5, 15, and 30 min. At the same instant, we selected three models, namely BILSTM, GRU, and SVR, for comparison, and independently executed each model on 30 instances according to the time interval, packaged the results, and presumed the average value for comparison. Overall, MAPE, RMSE, and other evaluation indicators implied that the proposed model delivered superior performance to other models at time intervals of 5 min. The results are shown in Figures 8–10.

![Daily Cycle Model Test Results and Real Values](image)

**Figure 8.** 5-minute forecast result graph.
Similarly, the results presented in the following Table 1 indicate that the SVR model produced the largest error in all predictions, because the SVR uses a kernel function to map numerous uncertain traffic flow data into high-dimensional space, and, consequently, fails to completely utilize the spatial and periodic characteristics for traffic prediction, thereby resulting in poor prediction performance. Upon considering the 5 min interval as a reference, the $MAE$, $MAPE$, and $RMSE$, and other indicators were reduced by 47.5%, 36.2%, and 47.2%, respectively. Compared to the SVR model, the GRU model more appropriately reflects the periodicity of traffic flow by predicting the time series of traffic flow. However, the GRU model is effective only in capturing the short-term characteristics of traffic flow and fails to capture the long-term features. Therefore, we introduced the BILSTM model to store additional information by forward and reverse information propagation through the bidirectional LSTM model in order to achieve the effect of long-term temporal feature sequences and improve the stability of the model. Compared to the GRU model, the BILSTM model reduced the performance indices of MAE, MAPE, and RMSE by 15%, 29.5%, and 10.6%, respectively. Both the GRU and BILSTM models provided predictions based on the time series characteristics of traffic flow, but they could not reflect the spatial characteristics of traffic flow. Therefore, in this paper, we propose the combined CNN-BILSTM model. As evaluated from the following Table 1, the CNN-BILSTM model was further superior in performance indicators, which were 30.4%, 32.2%, and 39.6% greater than the BILSTM model, respectively.
Table 1. Prediction performance of different models at different time intervals.

<table>
<thead>
<tr>
<th>T</th>
<th>Indices</th>
<th>SVR</th>
<th>GRU</th>
<th>BILSTM</th>
<th>CNN-BILSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 min</td>
<td>MAE</td>
<td>16.05</td>
<td>8.42</td>
<td>7.16</td>
<td>4.98</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>0.069</td>
<td>0.044</td>
<td>0.031</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>21.56</td>
<td>11.39</td>
<td>10.18</td>
<td>6.15</td>
</tr>
<tr>
<td>15 min</td>
<td>MAE</td>
<td>20.29</td>
<td>11.39</td>
<td>9.74</td>
<td>6.81</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>0.085</td>
<td>0.05</td>
<td>0.039</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>30.03</td>
<td>16.46</td>
<td>13.47</td>
<td>8.67</td>
</tr>
<tr>
<td>30 min</td>
<td>MAE</td>
<td>22.85</td>
<td>12.75</td>
<td>11.92</td>
<td>8.12</td>
</tr>
<tr>
<td></td>
<td>MAPE</td>
<td>0.071</td>
<td>0.038</td>
<td>0.032</td>
<td>0.022</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>35.22</td>
<td>18.75</td>
<td>18.96</td>
<td>10.11</td>
</tr>
</tbody>
</table>

6. Conclusions and Future Work

In this study, we proposed the combined CNN-BILSTM model for forecasting short-term traffic flow on highways. First, the spatial features of the traffic flow data were extracted through the CNN model, and the output data were inputted into the BILSTM model for temporal feature extraction and prediction. Subsequently, the results were compared with alternative forecasting models. The comparative analyses of the prediction performances of the models revealed the superior performance of the proposed model, including the cases of forecasting on overall data, categorizing data into daily and weekly cycles to account for the influence of periodicity, or considering the prediction effect of the model at various time intervals. The prediction precision of the model was greater than that of the comparison model, which established the enhanced prediction effect of the proposed model.

Although the proposed model offers several advantages for predicting traffic flow, it poses certain limitations, as well. In the future, we should take into account the influence of speed on traffic flow and take into account the influence of weather factors, occupancy rate, and road construction. In this context, we can further study multilane traffic prediction, which is not limited to the prediction of total traffic flow. This will further enhance the precision of the model.

Author Contributions: Writing—original draft, Y.C.; Writing—review & editing, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fujian Zhi-lian-yun Supply Chain Technology and Economy Integration Service Platform from the Fujian Association for Science and Technology, the Fujian-Kenya Silk Road Cloud Joint R&D Center (2021D021) from Fujian Provincial Department of Science and Technology, the Fujian Social Sciences Federation Planning Project (FJ2021Z006), General program of Fujian Natural Science Foundation (2022J01941) and the Development Fund of Scientific Research from Fujian University of Technology (GY-518109).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, and further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the reviewers and editors for improving this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References