Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.048955 Published Online: 11 April 2024

ARTICLE

Traffic Flow Prediction with Heterogeneous Spatiotemporal Data Based on a Hybrid Deep Learning Model Using Attention-Mechanism

Jing-Doo Wang[1] and Chayadi Oktomy Noto Susanto[1,2,*]

[1]Departement of Computer Science and Information Engineering, Asia University, Taichung, Taiwan

[2]Departement of Information Technology, Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia

*Corresponding Author: Chayadi Oktomy Noto Susanto. Email: chayadions@ft.umy.ac.id

Received: 22 December 2023 Accepted: 08 March 2024

ABSTRACT

A significant obstacle in intelligent transportation systems (ITS) is the capacity to predict traffic flow. Recent advancements in deep neural networks have enabled the development of models to represent traffic flow accurately. However, accurately predicting traffic flow at the individual road level is extremely difficult due to the complex interplay of spatial and temporal factors. This paper proposes a technique for predicting short-term traffic flow data using an architecture that utilizes convolutional bidirectional long short-term memory (Conv-BiLSTM) with attention mechanisms. Prior studies neglected to include data pertaining to factors such as holidays, weather conditions, and vehicle types, which are interconnected and significantly impact the accuracy of forecast outcomes. In addition, this research incorporates recurring monthly periodic pattern data that significantly enhances the accuracy of forecast outcomes. The experimental findings demonstrate a performance improvement of 21.68% when incorporating the vehicle type feature.

KEYWORDS

Traffic flow prediction; sptiotemporal data; heterogeneous data; Conv-BiLSTM; data-centric; intra-data

1 Introduction

The performance of traffic flow prediction is the foundation for dynamic strategies and applications in intelligent transportation systems. This issue holds immense practical importance for enhancing traffic safety and alleviating road congestion. This predicting capability enables effective decision-making for traffic management, encompassing adjustments to traffic signals and the implementation of temporary traffic control measures. Hence, it has progressively garnered the interest of numerous researchers [1–4], with extensive usage in detecting transportation anomalies [5,6], optimizing resource allocation [7], managing logistics supply chains [8], and overseeing urban administration [9,10].

Traffic flow typically has inherent patterns, suggesting it is generally possible to predict it accurately. Over the past few decades, significant research has been conducted on predicting traffic flow. Various methods have been explored, including the autoregressive integrated moving average (ARIMA) approach [11], support vector regression (SVR), and K-nearest neighbors (KNN) [12]. ARIMA is a parametric model that relies on rigorous theoretical assumptions [13,14]. It works in a basic linear model. Most conventional parameter models are characterized by simplicity and efficiency in computation. However, they exhibit limited robustness and are better suited for road sections with consistent traffic circumstances. Consequently, it cannot reliably predict traffic flow, which is complex and nonlinear [15,16]. As a result of the previously described shortcomings of conventional statistical models, researchers gravitated towards machine learning models.

Machine learning models such as SVR and KNN provide adaptability as they can acquire knowledge from the data. Nevertheless, the prediction performance of these techniques remains unsatisfactory as they solely account for the temporal fluctuations of traffic flow, disregarding its stochastic and nonlinear characteristics. Moreover, the conventional machine learning approaches rely on manually designed parameters to seize the properties of traffic flow. Another limitation of classical machine learning is the laborious task of manually extracting features [15]. This condition creates a high dependency on domain experts in certain fields and causes insufficient for achieving precise prediction performance.

Presently, two distinct perspectives exist on enhancing performance in deep learning (DL), specifically the model-centric and the data-centric approaches. In a model-centric context, the researcher iteratively enhances the designed model (algorithm/code) while keeping the quantity and kind of acquired data constant. On the other hand, researchers of the data-centric approach adhere to static models while consistently enhancing the quality of the data [17]. Deep learning has made significant advancements in recent years, demonstrating exceptional achievements in diverse fields such as speech recognition and computer vision. In contrast to conventional artificial neural network (ANN) models, deep learning models employ multi-layer structures to retrieve intrinsic features from extensive raw data sets automatically. Due to the influence of deep learning, there has been a notable increase in enthusiasm for transportation research in recent years. Numerous deep-learning techniques for predicting traffic flow have been proposed [18–21]. In recent times, there has been a notable improvement in the predictive performance of deep models, including the variance of long short-term memory network (LSTM) [22] and convolutional neural network (CNN) [23], which can be attributed to their robust ability to effectively capture temporal or spatial dependencies, surpassing the performance of shallower models.

Nonetheless, current studies that rely on neural network models for traffic flow prediction encounter the following limitations. Certain studies utilize basic neural network models like stacked autoencoders (SAE), LSTM, or CNN, failing to capture traffic flow's intricate characteristics adequately; as a result, these models offer only marginal improvements in prediction performance. Typically, LSTM captures temporal characteristics, while CNN extracts spatial features. LSTMs are inherently unidirectional, which implies that they can handle information sequentially, moving from the past to the future. This condition can provide a constraint when dealing with worldwide scope and interdependencies in both directions. Addressing this issue, bidirectional long short-term memory (BiLSTM) was designed to process information from the past and future simultaneously, enabling the model to gain a more comprehensive knowledge of the whole context of the data series [24]. CNN excels at obtaining highly effective spatial information. However, it has difficulties when it comes to extracting temporal aspects. The two models mentioned above are often utilized independently for each distinct situation. By combining the advantages of both models, it is possible to overcome existing challenges associated with the complex and nonlinear spatiotemporal properties of traffic flow data.

Moreover, existing studies do not fully exploit the complex structure present in traffic flow data. They solely employ attention methods on a single network layer, neglecting to allocate attention to the remaining layers [25]. Several variables impact the performance of traffic flow prediction. They disregard the significance of conditions or occurrences at a specific location or during specific time intervals in previous traffic patterns, which are crucial in making precise predictions about future traffic patterns [26,27]. Here, we propose a unique hybrid deep learning Conv-BiLSTM method incorporating an attention mechanism to tackle the abovementioned issues. This methodology utilizes heterogeneous multi-periodic intra-spatiotemporal data to improve the performance of traffic flow prediction. The main contributions of this paper are as follows:

- In this study, we proposed a novel hybrid deep learning model incorporating Conv-BiLSTM networks and BiLSTM using an attention mechanism to leverage traffic flow's spatiotemporal and periodicity characteristics effectively. In contrast to the current hybrid model utilized for traffic flow prediction, the Conv-BiLSTM model demonstrates enhanced efficiency in capturing spatiotemporal data. The efficiency is achieved by processing spatial and temporal features together, improving predictive performance.
- Attention mechanisms are developed for Conv-BiLSTM and BiLSTM modules to dynamically assign varying levels of attention to a sequence of traffic flows at distinct temporal instances. The suggested system can autonomously differentiate the significance of each flow sequence's contribution to the ultimate prediction performance outcome.
- This study combines two methodologies: the model-centric approach and the data-centric approach. In the context of the data-centric approach, we include intra-data to segregate traffic flow into distinct categories depending on five vehicle types rather than aggregating them into a single count of vehicles.
- In this study, our approach collects heterogeneous spatiotemporal data features (holidays, weather, and vehicle type) at current, daily, weekly, and monthly periodicities that previous studies have not implemented to improve prediction performance.

The subsequent sections of this work are structured in the following manner. In Section 2, the data representation is introduced. Section 3 introduces a unique deep-learning methodology for predicting traffic flow. In Section 4, we undertake tests on the dataset and evaluate the predictive performance compared to many current approaches. Section 5 presents the conclusion and future research.

2 Research Data

The traffic flow dataset used in this study was obtained from the Taiwan Ministry of Transportation, which can be accessed publicly in the Traffic Data Collection System (TDCS) (https://tisvcloud.freeway.gov.tw/history/TDCS/M06A/). The dataset presented in this research encompasses pertinent information on the number of cars observed on the Taiwan National Freeway. This study focused on observing the traffic flow at eight gantries along Taiwan National Freeway No. 3 from November 2016 to October 2019 (3 years). The input of this paper is from the output of frequency distributions of repeats extracted from vehicle trips via previous approaches developed in [28,29]. On the other hand, there was another similar work to have maximal repeat extraction from bus passenger's trips [30]. Apart from data on the traffic flow, this research also involves other features such as weather (https://openweathermap.org/) and holiday (https://timeanddate.com) data. Detailed feature information can be found in Table 1.

Table 1: Feature information [31]

Traffic flow	Holidays	Weather	
		Numerical	Categorical
Sedan (VT-31)	Weekday	Wind speed	Clear
Pickup (VT-32)	Weekend	Humidity	Clouds
Bus (VT-41)	Cont_holiday		Drizzle
Truck (VT-42)			Fog
Trailler (VT-5)			Haze
			Mist
			Rain
			Thunderstorm

The weather feature is classified into two distinct categories: numerical and categorical. The wind speed measurements consist of decimal values spanning from 0 to 15.95, while the humidity readings are whole numbers ranging between 0 and 100. Conversely, various weather conditions are distinguished by boolean values. Holidays fall into three categories: weekday, weekend, and cont_holiday. The weekday category pertains to holidays occurring on any day from Monday to Friday. The weekend condition refers to holidays that occur on Saturdays or Sundays. The cont_holiday condition encompasses holidays that may occur either prior to or following weekdays or weekends. Meanwhile, for vehicle types, we divide traffic flow into five categories (sedan, pickup, bus, truck, trailer). Detailed data preprocessing can be seen in the previous research [31].

Successful traffic flow prediction models must accurately capture the random and nonlinear characteristics of transportation traffic conditions. Numerous traffic models based on statistics or machine learning techniques have been developed to enhance the accuracy of predictions. One of the crucial processes of machine learning is feature learning. The process entails extracting and selecting the most significant aspects from past traffic flow data.

The characteristics of traffic flow commonly display spatiotemporal correlation and periodic patterns. Specifically, the traffic flow at the observation area is influenced by the traffic conditions of nearby locations and affected by previous time intervals. Traffic flow also demonstrates periodic patterns on a current, daily, weekly, and monthly basis. For example, the traffic flow variation on the same day for two consecutive weeks exhibits remarkable similarity. What is rarely considered is the monthly periodicity, even though monthly periodicity is the most pronounced pattern compared to daily and weekly periodicity. For instance, it can be related to weather or holiday patterns. Generally, weather conditions and holidays within a country tend to remain relatively consistent, unlike daily and weekly periodicity, which are more random. This research paper introduces a deep learning model that utilizes spatiotemporal correlation and periodic characteristics to enhance short-term traffic flow prediction.

Predicting traffic flow aims to enhance transportation efficiency by delivering precise and timely information regarding upcoming traffic conditions. The issue of predicting traffic flow can be stated in the following manner. Consider that X_T^p represents the traffic flow throughout the T th time interval. The observed feature can refer to vehicle types, weather, or type of holiday. The observed prediction target in this research is vehicle-type sedan (VT-31) traffic flow. Given the historical traffic flow sequence of observed feature, the objective at the current time t is to forecast the traffic flow at the time interval $(t + h\ \Delta)$, where Δ represents the prediction horizon $\left\{X_T^p\right\}(T) = t - n\Delta,\ \ldots,\ t - \Delta,\ t$, and $p \in P$, where P is the set of observation features. This study considers the following parameters: $\Delta = 1$ h, $n = 12$, and $h = 1$. n indicates that we utilize a historical dataset spanning 12 h to forecast the traffic flow for the upcoming one hour. For simplicity in explanation, we represent $t - n$ as $t - n\Delta$ by excluding the Δ symbol in this paper.

Before presenting our traffic flow prediction model, we explain the process of creating a historical dataset for this research. Here, we represent time series numerical data from the dataset as an image, adopting the approach carried out in previous research [31]. Let f_t^p represent the value of each observation feature p at time t. Each feature in Table 1 undergoes a normalization step to ensure that its data value falls within the range of [0,1]. The representation of the historical observation feature p can be expressed as $X_T^p = [f_0^p, \ldots, f_T^p]$.

Next, we combine all features (p features) to generate a matrix representing the spatiotemporal traffic flow. Where $\acute{Ç}_t^p = [f_t^1, f_t^2, \ldots, f_t^p]$ designates the observation feature of the prediction-based current periodic (c) at time t.

$$\acute{Ç}_t^p = \begin{bmatrix} f_{t-n}^1 & f_{t-(n+1)}^1 & \cdots & f_t^1 & \cdots & f_{t+n}^1 \\ f_{t-n}^2 & f_{t-(n+1)}^2 & \cdots & f_t^2 & \cdots & f_{t+n}^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{t-n}^p & f_{t-(n+1),}^p & \cdots & f_t^p & \cdots & f_{t+n}^p \end{bmatrix} \tag{1}$$

Furthermore, we contemplate the periodic characteristics of the traffic flow and other features. We consider daily (d), weekly ($\ddot{\mathrm{W}}$), and monthly ($\dot{\mathrm{m}}$) patterns to create historical data and incorporate all features with periodicity. The traffic data that shows a daily pattern can be obtained by considering the n time intervals before and after the precise moment t on the previous day. This can be stated as:

$$D_t^p = \begin{bmatrix} f_{t-d-n}^1 & f_{t-d-n+1}^1 & \cdots & f_{t-d}^1 & f_t^1 & \cdots & f_{t+n}^1 \\ f_{t-d-n}^2 & f_{t-d-n+1}^2 & \cdots & f_{t-d}^2 & f_t^2 & \cdots & f_{t+n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{t-d-n}^p & f_{t-d-n+1}^p & \cdots & f_{t-d}^p & f_t^p & \cdots & f_{t+n}^p \end{bmatrix} \tag{2}$$

where d designates the exact instant as time t on the last day. Likewise, we obtain historical traffic flow data that exhibits a weekly pattern by analyzing the time intervals before and after the same instant in the previous week, denoted as time t. We employ the following method:

$$\ddot{\mathrm{W}}_t^p = \begin{bmatrix} f_{t-\ddot{\mathrm{w}}-n}^1 & f_{t-\ddot{\mathrm{w}}-n+1}^1 & \cdots & f_{t-\ddot{\mathrm{w}}}^1 & f_t^1 & \cdots & f_{t+n}^1 \\ f_{t-\ddot{\mathrm{w}}-n}^2 & f_{t-\ddot{\mathrm{w}}-n+1}^2 & \cdots & f_{t-\ddot{\mathrm{w}}}^2 & f_t^2 & \cdots & f_{t+n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{t-\ddot{\mathrm{w}}-n}^p & f_{t-\ddot{\mathrm{w}}-n+1}^p & \cdots & f_{t-\ddot{\mathrm{w}}}^p & f_t^p & \cdots & f_{t+n}^p \end{bmatrix} \tag{3}$$

where $\ddot{\mathrm{W}}$ indicates the identical point as time t in the last week, month patterns are denoted as combined data from the current month and the previous month.

$$\dot{\mathrm{M}}_t^p = \begin{bmatrix} f_{t-\dot{\mathrm{m}}-n}^1 & f_{t-\dot{\mathrm{m}}-n+1}^1 & \cdots & f_{t-\dot{\mathrm{m}}}^1 & f_t^1 & \cdots & f_{t+n}^1 \\ f_{t-\dot{\mathrm{m}}-n}^2 & f_{t-\dot{\mathrm{m}}-n+1}^2 & \cdots & f_{t-\dot{\mathrm{m}}}^2 & f_t^2 & \cdots & f_{t+n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{t-\dot{\mathrm{m}}-n}^p & f_{t-\dot{\mathrm{m}}-n+1}^p & \cdots & f_{t-\dot{\mathrm{m}}}^p & f_t^p & \cdots & f_{t+n}^p \end{bmatrix} \tag{4}$$

The visual depiction of the merging process for each data by period can be seen in Fig. 1. The dataset marked as "blue" encompasses historical data over the preceding 12-h period. In contrast, the dataset marked as "orange" pertains to historical data, specifically from the corresponding time of the current day. The "purple" data represents the 12-h historical records of the preceding week, whereas the "brown" segment encompasses the identical dataset from the preceding month. The combination of each data is represented as current, daily, weekly, and monthly periodic data, as mentioned below.

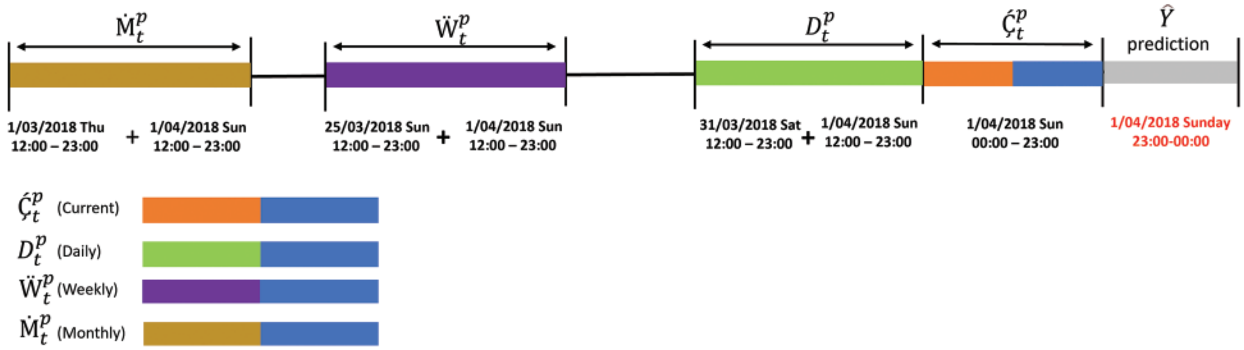

Figure 1: Data input representation

3 Proposed Method

This section explains the proposed hybrid model for predicting traffic flows. The proposed model consists of a Conv-BiLSTM module and three BiLSTM modules. This part of the proposed method can be seen in Fig. 2. In the first step, the primary objective of the convolutional network is to extract spatiotemporal information. The data set used in this research consists of heterogeneous data covering three aspects: traffic flow, holidays, and weather with different periodicity patterns. Convolutional models are most suitable for managing this type of data. In the second step, the primary purpose of BiLSTM is to extract information from the temporal characteristics of traffic flows. This model will extract crucial data based on periodic patterns for each feature. In prior studies, the authors [25] employed a comparable methodology. One notable distinction is how the input data is structured for the Conv-BiLSTM and BiLSTM layers.

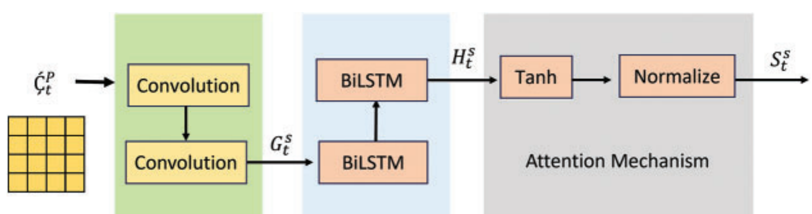

Figure 2: Conv-BiLSTM with attention mechanism

Previous studies have utilized homogenous inter-data traffic flow from two distinct geographical areas [25]. The data on homogeneous traffic flow does not provide distinctions among various types of vehicles. In this investigation, we employed heterogeneous intra-data [27], which refers to data collected from a specific area (eight gantries in Taiwan National Freeway No.3). This research categorizes traffic flow data based on five different types of vehicles (sedan, pickup, bus, truck, trailer). Furthermore, we have incorporated an attention mechanism that operates in the Conv-BiLSTM layer and across all model layers. Running this strategy enables the model to dynamically assess the varying significance levels of flow sequences at different periodic instances. The subsequent subsections will provide a comprehensive explanation of each module.

3.1 Conv-BiLSTM

The Conv-BiLSTM module serves as the primary constituent of the model that has been suggested, seeking to derive spatiotemporal characteristics from the traffic flow. The Conv-BiLSTM module integrates a convolutional neural network with a Bidirectional Long Short-Term Memory (BiLSTM) network, as depicted in Fig. 2. The architecture in the first stage consists of two convolutional layers. In the next stage, it passes through two BiLSTM layers.

The Conv-BiLSTM model takes as input spatiotemporal data derived from three distinct features, which are denoted as the current periodic matrix $\acute{Ç}_t^p$. This matrix, as denoted in Eq. (1), represents the historical data that is to be forecasted. To acquire the spatiotemporal feature, a one-dimensional (1-D) convolution operation is conducted on the traffic flow data $\acute{Ç}_t^p$ at each time step t. Pooling is not performed in the convolution layer. The output of the last convolution layer (layer 2) within this particular layer is represented as G_t^s. Furthermore, the output of the convolutional layer is subsequently utilized as the input for the BiLSTM layer. In this study, to improve the performance of traffic flow prediction, we leverage the BiLSTM models, which can capture temporal characteristics of traffic data by employing contextual information from two-directional. The bidirectional character of BiLSTM models enables them to capture long-term dependencies present in the data more efficiently. Individuals can retain and recall information from preceding and subsequent sequence segments. This cognitive function is important in activities necessitating comprehension of the whole context.

The initial BiLSTM layer is tasked with processing the sequential output derived from the final convolution layer, $G_T^s = G_{t-n}^s, \ldots, G_{t-1}^s, G_t^s$, spanning from the beginning to the last. This section computes the hidden state value for each time step $H_{1,T}^s = H_{1,t-n}^s, \ldots, H_{1,t-1}^s, H_{1,t}^s$. Next, the concealed state sequence $H_{1,T}^s$ is fed into the second BiLSTM layer to compute the hidden state $H_{2,T}^s$ at time step t, which serves as the output of the complete BiLSTM network H_T^s. The output of this last BiLSTM layer subsequently serves as the input for the attention mechanism, which will be elaborated on in the following section.

3.2 Bi-Directional LSTM for Temporal Dependency

Acquiring temporal dependence is another crucial challenge in traffic flow prediction. Recurrent neural network (RNN) is frequently employed to process data that exhibits sequential properties. The Elman Network, introduced by Elman in 1990, is considered the most typical and fundamental version of the standard RNN frequently utilized [32]. Nevertheless, the conventional RNN often encounters gradient explosion and gradient disappearance issues while handling lengthy time series data. The LSTM cell incorporates three control gates: the input, forget, and output. These gates utilize three techniques to regulate the flow of information inside the network, enabling the implementation of long-term memory. Fig. 3 illustrates the conventional configuration of the LSTM cell.

Fig. 3 depicts $\boldsymbol{X}_t$ as the parameter input value of the LSTM cell at a specific moment t, $\boldsymbol{C}_t$ as the state value of the memory cell and $\boldsymbol{h}_t$ as the hidden value output at time t. $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_g$, $\boldsymbol{W}_o$, and b represents the weights matrix and bias of each threshold layer. The term *tanh* refers to the tanh activation function, while σ denotes the sigmoid activation functions. The internal computation method of LSTM can be elucidated through Eqs. (5) to (10):

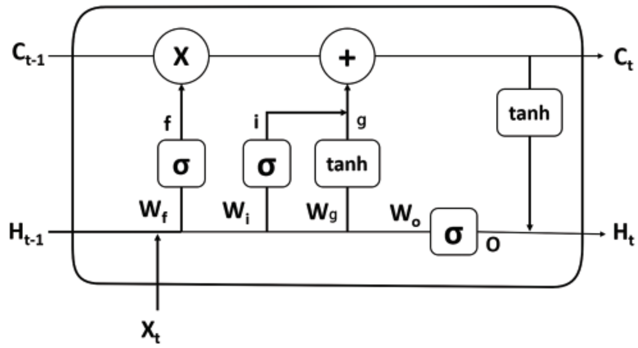

Figure 3: The LSTM network structure [31]

Step 1: Compute the activation value f_t of the forget gate at time t using the following formula:

$$f_t = \sigma\left(W_f\left[X_t; H_{t-1}\right] + b_f\right) \tag{5}$$

Step 2: The next step is to calculate the numerical value of the input gate i_t and the candidate's state g_t of the cell at time t. The precise calculating formulas are as follows:

$$i_t = \sigma\left(W_i\left[X_t; H_{t-1}\right] + b_i\right) \tag{6}$$

$$g_t = tanh\left(W_g\left[X_t; H_{t-1}\right] + b_g\right) \tag{7}$$

Step 3: Compute the value C_t for updating the cell state at time t using the following formula:

$$C_t = f_t \times C_{t-1} + i_t \times g_t \tag{8}$$

Step 4: Compute the value of the output variable o_t of the output gate at time t using the following formula:

$$o_t = \sigma\left(W_o\left[X_t; H_{t-1}\right] + b_o\right) \tag{9}$$

$$H_t = o_t \tanh(C_t) \tag{10}$$

Fig. 4 depicts the bidirectional LSTM network, which includes forward and backward Long Short-Term Memory (LSTM) models. The forward LSTM processes information in one way, while the reverse LSTM processes information in the opposite direction. The input sequence undergoes processing by the forward LSTM layer, given the output $H_T^{forward}(H_T^f)$, while the reverse form of the input sequence is inputted into the backward LSTM layer, given the output $H_T^{backward}(H_T^b)$.

Ultimately, the concealed states of the forward and backward layers are combined to form the output. The basic LSTM's limitation of just utilizing prior information is resolved, and the prediction performance is enhanced by implementing two unidirectional LSTMs. We utilize the BiLSTM to capture the temporal correlation of traffic flow in our study. Fig. 5 depicts the general structure of the BiLSTM module utilized in the proposed model. In this module, D_t^p, $\ddot{\mathrm{W}}_t^p$, and $\dot{\mathrm{M}}_t^p$ represent the input of the LSTM. $H_t^{d,f}$, $H_t^{w,f}$ and $H_t^{m,f}$ represent the output of the forward LSTM while $H_t^{d,b}$, $H_t^{w,b}$ and $H_t^{m,b}$ represent the backward LSTM output when the inputs are D_t^p, $\ddot{\mathrm{W}}_t^p$, and $\dot{\mathrm{M}}_t^p$, respectively. To account for the regularity of traffic data, we employ numerous BiLSTM layers to extract recurring characteristics from past traffic data.

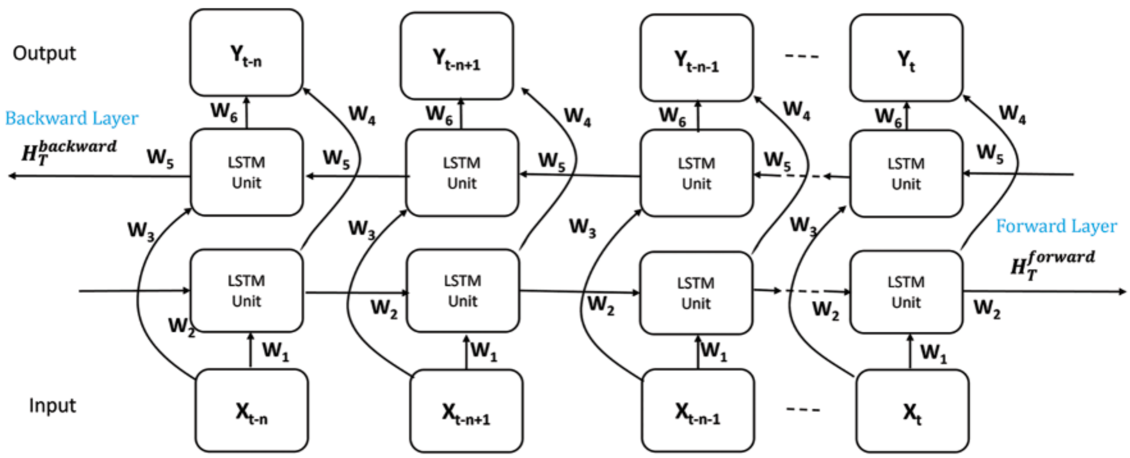

Figure 4: The structure of the BiLSTM network

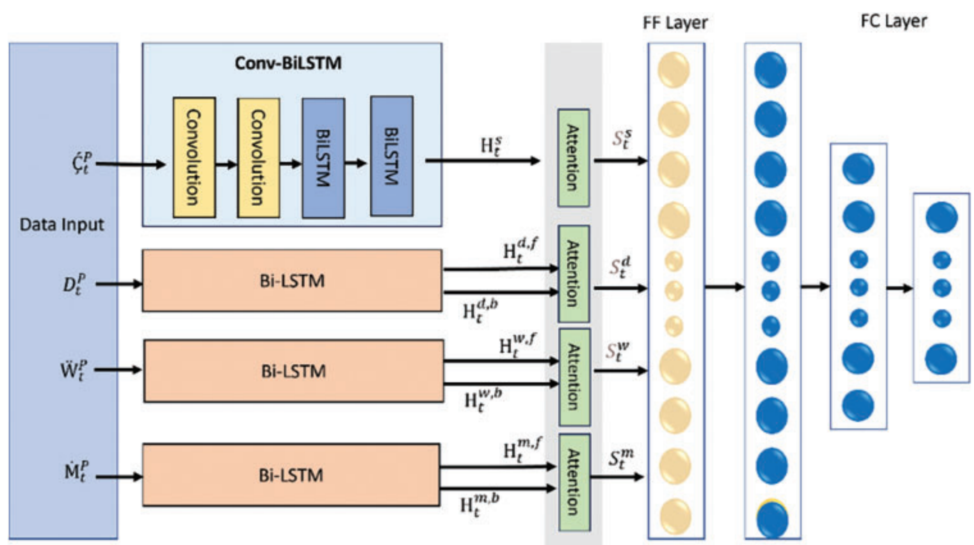

Figure 5: The proposed model

3.3 Attention Mechanism Based on Temporal Dependency

The attention mechanism approach has been extensively used across various domains, including natural language processing, image processing, and speech recognition. For example, the utilization of attention mechanisms to enhance translation accuracy initially emerged in the context of translation machines [1,33]. In short, the attention mechanism directs its emphasis on information that significantly influences the results and reduces the weight of unimportant information during the feature extraction process. The relative significance of traffic flow data at various time intervals may vary concerning the forecasting objective. In the domain of traffic flow prediction, a similar phenomenon is observed, where the influence of traffic flow varies at different periods, affecting the relevance of prediction performance [25]. The variability of traffic conditions at a given observation site can exhibit temporal fluctuations when forecasting traffic flow at the site of an observation area. In instances of congestion, the anticipated outcome may be more significantly impacted by the traffic conditions observed at a remote point instead of those observed at a closer point.

However, the conventional BiLSTM model cannot determine which segments of a traffic flow sequence are essential or significant. We have devised a dedicated attention mechanism tailored for the Conv-BiLSTM module to tackle this problem. This mechanism enables automatic identification and utilization of varying importance's level within a traffic flow sequence at different time points. We divide the temporal dependency into four categories based on the characteristics of the traffic data: current, daily, weekly, and monthly period. The link between multiple recent time intervals and the desired one is referred to as the current pattern; for example, traffic conditions at 10:00 am will influence the situation at 11:00 am. The current, daily, weekly, and monthly patterns reference the recurring character of human behavior. For instance, weekday traffic patterns vary similarly, with distinct morning and evening rush hours. Additionally, the morning rush hour periods may be postponed due to later weekend wake-up times.

An attention method is employed to dynamically modify the weighting of the output from the BiLSTM module. The expression of the attention mechanism's implementation can be formulated as follows:

$$\mu_{it} = \tanh\left(W_\omega h_{it} + b_\omega\right) \tag{11}$$

$$\beta_{it} = \frac{exp\left(\mu_{it}^T \mu_\omega\right)}{\sum_t exp\left(\mu_{it}^T \mu_\omega\right)} \tag{12}$$

$$S_i = \sum_t \beta_{it} h_{it} \tag{13}$$

The learnable parameters in this context are denoted as W_ω, b_ω, and μ_ω. The attention score is represented by β_{it}, and the output of the attention layer is denoted as S_i.

3.4 Output Layer

Following the attention layer, the spatiotemporal and periodicity features derived from the three network components are consolidated into a feature vector via a feature fusion layer. Assuming that $X \in R^{N\times C}$ represents the input to the output layer, a two-layer fully connected neural network is employed to predict a single timestep. T two-layer fully connected neural networks are employed to predict T future timesteps. The final forecast is derived by aggregating the prediction results from each timestep. The specific process can be outlined as follows:

$$\hat{y}^{(i)} = ReLU\left(XW_1^{(i)} + b_1^{(i)}\right) \times W_2^{(i)} + b_2^{(i)} \in R^{N\times 1} \tag{14}$$

$$\hat{Y} = \left[\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots \hat{y}^{(T)}\right] \in R^{N\times T} \tag{15}$$

where the variable $\hat{y}^{(i)}$ represents the timestep used for making predictions at time i, $W_1^{(i)} \in R^{N\times 1}, b_1^{(i)} \in R^{C'}$, $W_2^{(i)} \in R^{C'\times 1}$, and $b_2^{(i)} \in R$ represents the parameters that can be learned. The dimension of the output of the first fully linked layer is denoted as C'. We use $\hat{Y}$ and $\hat{y}$ to designate the predicted and ground truth values. Table 2 shows a training algorithm for a Conv-BiLSTM model.

Table 2: Training algorithm of Conv-BiLSTM model

Algorithm 1 Training algorithm of Conv-BiLSTM model

Input: Historical observation: X_T^p; n, p; d; $\ddot{\mathrm{w}}$, $\dot{\mathrm{m}}$;
Output: Learned Conv-BiLSTM model
model $= \varnothing$
for *all available time interval* t $(0 \leq t \leq T-1)$ ***do***
  for *all features* p $(1 \leq p \leq P)$ ***do***
    $\acute{Ç}_t^p = \left[f_{t-n}^p, f_{t-(n+1),}^p, \ldots, f_t^p, \ldots, f_{t+n}^p\right];$
    $D_t^p = \left[f_{t-d-n}^p, f_{t-d-n+1}^p, \ldots, f_{t-d}^p, f_t^p, \ldots, f_{t+n}^p\right];$
    $\ddot{\mathrm{W}}_t^p = \left[f_{t-\ddot{\mathrm{w}}-n}^p, f_{t-\ddot{\mathrm{w}}-n+1}^p, \ldots, f_{t-\ddot{\mathrm{w}}}^p, f_t^p, \ldots, f_{t+n}^p\right];$
    $\dot{\mathrm{M}}_t^p = \left[f_{t-\dot{\mathrm{m}}-n}^p, f_{t-\dot{\mathrm{m}}-n+1}^p, \ldots, f_{t-\dot{\mathrm{m}}}^p, f_t^p, \ldots, f_{t+n}^p\right];$
    //f_{t+n+1}^p is ground truth for feature p at time t+n+1
    $\left(\left\{\acute{Ç}_t^p, D_t^p, \ddot{\mathrm{W}}_t^p, \dot{\mathrm{M}}_t^p\right\}, f_{t+n+1}^p\right)$ *model*
Set up all trainable parameters $\ominus$ in Conv-BiLSTM.
// $\ominus$ is all learnable parameters in Conv-BiLSTM
repeat
  randomly take a batch of parameters from the *model;*
  finding the best value of $\ominus$ with the smallest error;
until *stopping condition is achieved;*

4 Experimental Result

4.1 Metaparameter Settings

The proposed model was constructed utilizing the TensorFlow framework, and the experiments were carried out on a *Google Collaboration Pro Plus* platform equipped with a T4 GPU. The experiment involved a model that consisted of two convolution processes with three filters and 256 hidden units of BiLSTM. Using our dataset as a case study, we varied the size of the kernel convolution layer from 2 to 11. We employed a two-layered BiLSTM architecture to capture the periodic patterns in the traffic data effectively. Assign the optimization technique used was Adam, with an initial learning rate of 0.001, a batch size of up to 128, and an epoch size of 500 for this model. This Adam optimization technique was chosen since it has the capability to modify the learning rate adaptively. We adopt the Conv-LSTM model as a benchmark model, following the specifications specified in the research [19]. To determine how to assess the efficacy of the suggested model, we employed the mean absolute error (MAE) metric [34]. These parameters include a filter of size 10, a kernel size of 3, and a batch size of 128. The evaluation results using this model produced an MAE value of 21.041.

4.2 Proposed Model Performance

We performed comparison experiments using the following short-term traffic flow prediction methodologies to assess the prediction performance of the proposed model: Conv-LSTM, CNN-LSTM, and CNN-BiLSTM. We employ two scenarios, with the first scenario including applying the attention mechanism solely to the first layer of Conv-BiLSTM. Furthermore, every layer employs an attention technique. Fig. 6 shows the best prediction model performance based on their feature and kernel size. The selection of kernel size is essential in determining the performance of convolutional neural networks. The kernel is a compact window that traverses the input data to extract distinctive characteristics. The choice of kernel size directly impacts the network's capacity to capture spatial information in the input. Greater kernel sizes efficiently capture comprehensive patterns and advanced characteristics, although they can result in heightened computing intricacy and necessitate a larger number of parameters. Conversely, smaller kernel sizes excel in capturing intricate details and specific traits, improving parameter utilization. Ensuring the appropriate equilibrium of kernel sizes is crucial for attaining peak performance in a convolutional network, as it dictates the network's ability to acquire hierarchical representations and effectively adapt to various inputs. Evaluating the kernel size and other architectural decisions is crucial for creating convolutional networks that perform exceptionally well in different computer vision tasks.

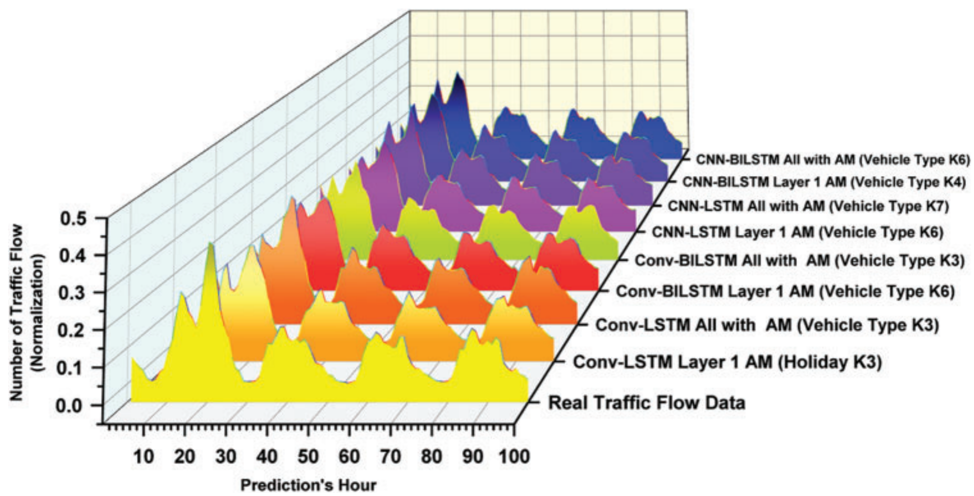

Figure 6: The best performance of all the prediction models with its features and kernel size

Fig. 7 indicates that the Conv-BiLSTM model, with vehicle type feature as input, generates the greatest performance among other models when employing a kernel size of 3 and applying an attention mechanism to each layer.

Tables 3 and 4 display the performance of various methods in predicting traffic flow over the next hour. Based on the data presented in the table, it is evident that the proposed model (Conv-BiLSTM with all layers using attention mechanism) outperformed all other models regarding the evaluation metrics with a mean absolute error (MAE) value of 16.478. There was a performance increase of 21,68%. Across several prediction models, the vehicle type feature consistently exhibits the lowest loss value. These findings indicate that the vehicle type attribute has the greatest influence on the performance of the results for prediction.

Fig. 8 illustrates the impact of features and kernel sizes on each model when all layers use an attention mechanism. Fig. 8a, by utilizing weather data as input, the Conv-LSTM model's performance is subpar while employing kernel sizes of 2 and 3. Conv-LSTM's performance improves when the kernel size exceeds 3, comparable to other models. The Conv-BiLSTM model achieves optimal performance with a kernel size of 5, as depicted in Fig. 8a. Fig. 8c demonstrates the input data related to the vehicle type of various periodicities significantly influences the model's performance. Employing kernel sizes between 2 and 6 demonstrates a propensity for achieving favorable performance outcomes. Meanwhile, kernels with a value larger than 6 have negligible effect.

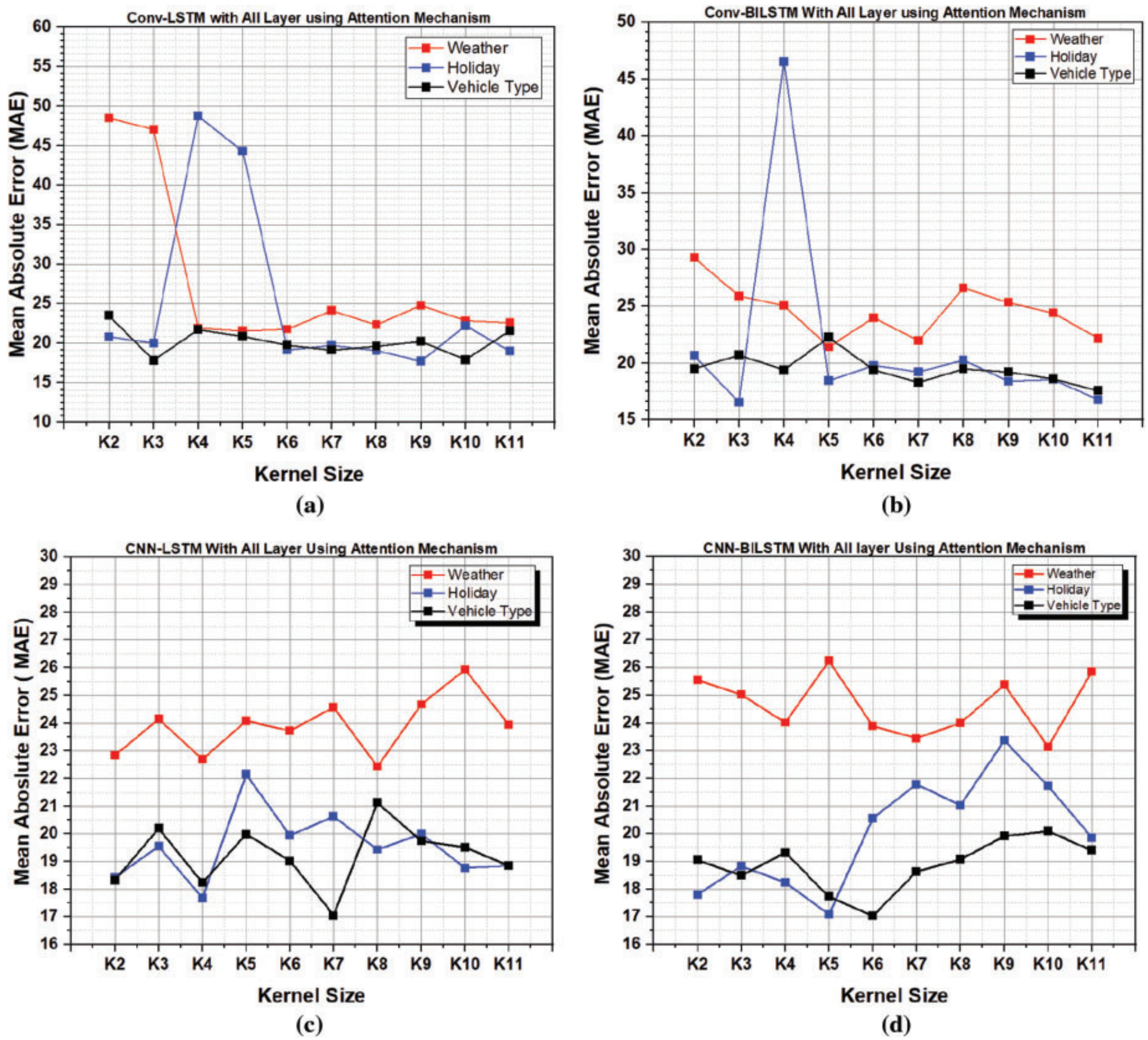

Figure 7: The effect of convolution kernel size on the traffic flow prediction model performance. (a) Conv-LSTM with all layer attention mechanism, (b) Conv-BiLSTM with all layer attention mechanism, (c) CNN-LSTM with all layer attention mechanism, (d) CNN-BiLSTM with all layer attention mechanism

Table 3: MAE value by the effect of convolution kernel size on the model with layer 1 using attention mechanism

Kernel size	Conv-LSTM			Conv-BiLSTM			CNN-LSTM			CNN-BiLSTM		
	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type
K2	46.470	20.645	20.691	23.049	19.115	18.640	24.612	18.483	18.975	24.578	21.722	19.671

(Continued)

Table 3 (continued)

Kernel size	Conv-LSTM			Conv-BiLSTM			CNN-LSTM			CNN-BiLSTM		
	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type
K3	21.028	**16.543**	19.407	21.915	19.987	18.493	24.127	20.824	22.651	25.217	18.952	19.179
K4	21.676	46.547	22.259	21.921	18.850	20.202	25.349	19.227	19.841	23.890	20.985	**17.807**
K5	47.899	18.440	19.392	22.650	20.876	20.530	23.646	19.809	20.494	24.554	20.276	21.516
K6	50.364	19.789	18.292	22.227	17.913	**16.891**	23.995	20.078	17.458	21.663	18.103	20.306
K7	21.583	19.215	19.473	22.689	20.614	18.944	25.908	19.257	18.885	25.638	20.558	18.921
K8	22.525	20.248	19.217	22.172	19.509	18.151	24.597	20.195	**17.452**	25.455	20.149	18.975
K9	20.444	18.393	18.592	25.818	17.857	17.618	24.769	20.732	18.189	23.935	18.529	19.735
K10	25.847	18.535	17.550	22.699	21.226	18.241	23.794	20.341	19.699	22.745	21.264	19.424
K11	22.810	16.774	20.308	24.601	18.805	20.297	24.793	17.345	19.900	24.806	18.013	19.272

Table 4: MAE value by the effect of convolution kernel size on the model with all layer using attention mechanism

Kernel size	Conv-LSTM			Conv-BiLSTM			CNN-LSTM			CNN-BiLSTM		
	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type	Weather	Holiday	Vehicle type
K2	48.520	20.826	23.492	29.268	18.933	17.726	22.832	18.432	18.331	25.545	17.805	19.041
K3	47.083	20.009	17.833	25.901	18.705	**16.478**	24.147	19.547	20.194	25.022	18.823	18.487
K4	21.916	48.789	21.717	25.050	18.394	20.632	22.696	17.682	18.230	24.003	18.240	19.312
K5	21.544	44.302	20.870	21.422	18.603	18.195	24.080	22.144	19.990	26.236	17.086	17.726
K6	21.770	19.146	19.746	23.989	18.808	17.871	23.725	19.943	19.027	23.875	20.561	**17.025**
K7	24.124	19.740	19.132	21.954	18.814	17.957	24.568	20.627	**17.060**	23.448	21.785	18.633
K8	22.322	19.072	19.608	26.614	19.863	19.216	22.425	19.427	21.124	23.986	21.025	19.065
K9	24.806	**17.727**	20.234	25.322	18.106	18.355	24.679	19.997	19.741	25.365	23.358	19.910
K10	22.861	22.222	17.904	24.403	17.195	19.438	25.931	18.768	19.512	23.139	21.716	20.092
K11	22.589	18.997	21.601	22.177	17.769	22.768	23.950	18.847	18.225	25.849	19.863	19.386

Based on the conducted studies, incorporating vehicle-type variables had the most significant effect on enhancing prediction performance, resulting in a 21.68% increase when utilizing the Conv-BiLSTM model. Table 4 demonstrates the fluctuation in error reduction when comparing various kernel sizes for each traffic flow prediction model using a model that incorporates attention techniques in all layers. The Conv-LSTM model has optimal performance when using a kernel size of 9 in combination with the holiday feature. Meanwhile, the Conv-BiLSTM model, which yields the most significant performance improvement when utilizing a kernel size of 3, incorporates the vehicle type feature. On the other hand, the CNN-LSTM and CNN-BiLSTM models, which have kernel sizes of 7 and 6, respectively, demonstrate the best performance when considering the vehicle type feature.

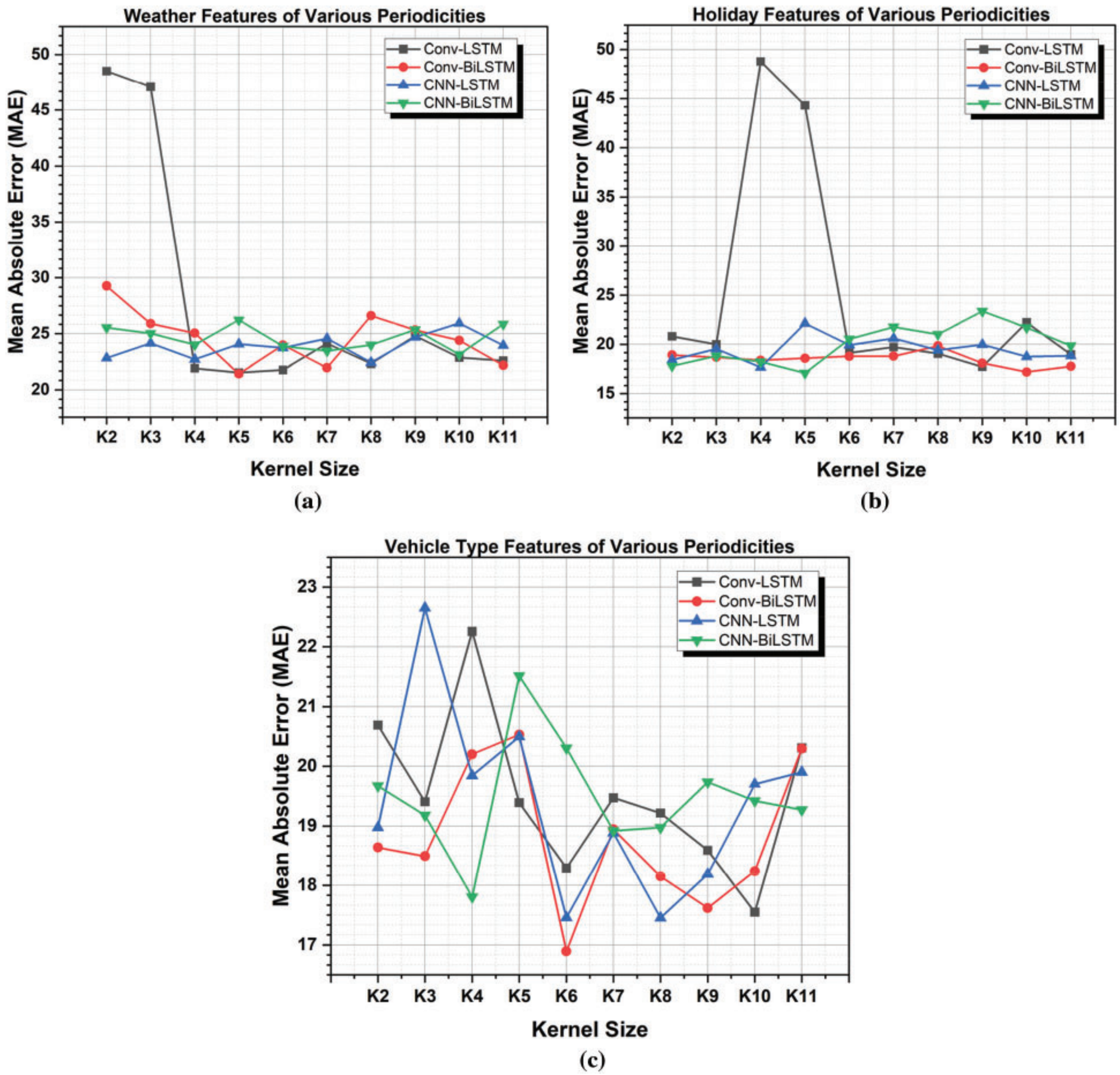

Figure 8: The effect of convolution kernel size on the traffic flow prediction model performance based on input data feature to proposed model using all layers with attention-mechanism. (a) Weather features of various periodicities, (b) holiday features of various periodicities, (c) vehicle type features of various periodicities

5 Conclusion and Future Research

This study examines the short-term traffic flow prediction by utilizing the dataset provided by the Taiwan Ministry of Transportation, specifically focusing on the number of cars on Taiwan National Freeway No. 3. The hybrid deep learning model that combines convolutional neural networks and BiLSTM networks was suggested as a way to deal with the complex and nonlinear features of traffic flow. The results of our study suggest that the Conv-BiLSTM model, which incorporates an attention mechanism, effectively captures spatiotemporal data. Furthermore, integrating the suggested attention mechanism throughout all layers amplifies the Conv-BiLSTM's efficacy in enhancing prediction performance. The traffic flow prediction model effectively catches repeating trends on a current, daily, weekly, and monthly basis, hence improving the performance of predictions. Integrating diverse features such as holidays, weather conditions, and vehicle types has benefited prediction models. The Conv-BiLSTM model, when combined with the vehicle type feature, enhances prediction performance by 21.68%. Empirical evidence shows that this strategy surpasses earlier methodologies in traffic flow prediction.

Current research only emphasizes capturing spatiotemporal correlations based on the nature and dynamics of traffic flow features. However, it has not paid attention to the Euclidean nature of the road structure, such as paying attention to the linkage of traffic flow information between road nodes [35]. In order to enhance prediction performance, it is necessary to enhance the present model by transforming it into an Euclidean grid that will enable the model to effectively capture the spatiotemporal correlation without sacrificing significant amounts of crucial information. Another challenge in this research is the potential for data bias and scalability issues. The observation area of this study specifically collects historical data based on traffic flow from eight gantries located on one section of Taiwan National Freeway No. 3. Future research is important to compare the results of traffic flow predictions involving observation areas from various points to test the model's robustness. However, many factors influence traffic flow, such as road conditions, events, and traffic flow in opposite directions. The experiment should consider more factors to improve prediction performance results in future work. The research would become more engaging if it possessed the capacity to predict traffic flow in real-time scenarios.

Acknowledgement: The authors would like to express their sincere gratitude to the anonymous referees for their insightful and constructive criticism. The author would also like to extend gratitude to Muhammadiyah University of Yogyakarta for providing ethical support and Asia University for supplying the practical laboratory where the research was conducted.

Funding Statement: The authors did not get any dedicated financial support for this study.

Author Contributions: The authors affirm their contribution to the paper in the following manner: Wang: drafting of the manuscript, analysis, data processing, and data collection. Susanto: description of the results, drafting of the manuscript, study conception, and analysis. Collaboratively, the writers analyzed the results and approved the final version of the manuscript.

Availability of Data and Materials: The raw material of the datasets can be obtained from https://tisvcloud.freeway.gov.tw/history/TDCS/M06A/. Data sharing was not carried out in this study.

Conflicts of Interest: The authors affirm that they do not have any conflicts of interest to disclose in relation to the current study.

References

1. Wang, L., Geng, X., Ma, X., Liu, F., Yang, Q. (2018). Cross-city transfer learning for deep spatio-temporal prediction. arXiv preprint arXiv:1802.00386.
2. Çetiner, B. G., Sari, M., Borat, O. (2010). A neural network based traffic-flow prediction model. *Mathematical and Computational Applications, 15(2),* 269–278.

3. Kamarianakis, Y., Shen, W., Wynter, L. (2012). Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. *Applied Stochastic Models in Business and Industry, 28(4),* 297–315.
4. Polson, N. G., Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies, 79,* 1–17.
5. Kong, X., Song, X., Xia, F., Guo, H., Wang, J. et al. (2018). LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web, 21,* 825–847.
6. Cook, A. A., Mısırlı, G., Fan, Z. (2019). Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal, 7(7),* 6481–6494.
7. Dafermos, S., Sparrow, F. T. (1971). Optimal resource allocation and toll patterns in user-optimised transport networks. *Journal of Transport Economics and Policy, 5(2),* 184–200.
8. Bhattacharya, A., Kumar, S. A., Tiwari, M., Talluri, S. (2014). An intermodal freight transport system for optimal supply chain logistics. *Transportation Research Part C: Emerging Technologies, 38,* 73–84.
9. Morioka, M., Kuramochi, K., Mishina, Y., Akiyama, T., Taniguchi, N. (2015). City management platform using big data from people and traffic flows. *Hitachi Review, 64(1),* 52–57.
10. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal, 1(1),* 22–32.
11. Wang, H., Liu, L., Dong, S., Qian, Z., Wei, H. (2016). A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD-ARIMA framework. *Transportmetrica B: Transport Dynamics, 4(3),* 159–186.
12. Hong, W. C., Dong, Y., Zheng, F., Wei, S. Y. (2011). Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. *Applied Mathematics and Computation, 217(15),* 6733–6747.
13. Li, X., Pan, G., Wu, Z., Qi, G., Li, S. et al. (2012). Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science, 6,* 111–121.
14. Lippi, M., Bertini, M., Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems, 14(2),* 871–882.
15. Karlaftis, M. G., Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies, 19(3),* 387–399.
16. Li, Y., Shahabi, C. (2018). A brief overview of machine learning methods for short-term traffic forecasting and future directions. *Sigspatial Special, 10(1),* 3–9.
17. Hamid, OH. editor (2022). From model-centric to data-centric AI: A paradigm shift or rather a complementary approach? *2022 8th International Conference on Information Technology Trends (ITT)*, Dubai, UEA, IEEE.
18. Fitters, W., Cuzzocrea, A., Hassani, M. editors (2021). Enhancing LSTM prediction of vehicle traffic flow data via outlier correlations. *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, IEEE.
19. Awan, N., Ali, A., Khan, F., Zakarya, M., Alturki, R. et al. (2021). Modeling dynamic spatio-temporal correlations for urban traffic flows prediction. *IEEE Access, 9,* 26502–26511.
20. Yao, R., Zhang, W., Zhang, D. (2020). Period division-based Markov models for short-term traffic flow prediction. *IEEE Access, 8,* 178345–178359.
21. Jing, Y., Hu, H., Guo, S., Wang, X., Chen, F. (2020). Short-term prediction of urban rail transit passenger flow in external passenger transport hub based on LSTM-LGB-DRS. *IEEE Transactions on Intelligent Transportation Systems, 22(7),* 4611–4621.
22. Graves, A., Graves, A. (2012). *Supervised sequence labelling*. Heidelberg: Springer.

23. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y. et al. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors, 17(4),* 818.
24. Subramanian, M., Lakshmi, S. S., Rajalakshmi, V. R. (2023). Deep learning approaches for melody generation: An evaluation using LSTM, BILSTM and GRU models. *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, New Delhi, India, IEEE.
25. Zheng, H., Lin, F., Feng, X., Chen, Y. (2020). A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems, 22(11),* 6910–6920.
26. Rejeb, IB., Ouni, S., Zagrouba, E. editors (2019). Intra and inter spatial color descriptor for content based image retrieval. *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, Abu Dhabi, UEA, IEEE.
27. Zhao, Y., Lin, Y., Zhang, Y., Wen, H., Liu, Y. et al. (2022). Traffic inflow and outflow forecasting by modeling intra-and inter-relationship between flows. *IEEE Transactions on Intelligent Transportation Systems, 23(11),* 20202–20216.
28. Wang, J. D. (2016). Extracting significant pattern histories from timestamped texts using MapReduce. *The Journal of Supercomputing, 72,* 3236–3260.
29. Wang, C. T. (2019). Method for extracting maximal repeat patterns and computing frequency distribution tables. U.S. Patent.
30. Wang, J. D., Pan, S. H., Ho, C. Y., Lien, Y. N., Liao, S. C. et al. (2020). Online Web query system for various frequency distributions of bus passengers in Taichung city of Taiwan. *IET Smart Cities, 2(3),* 135–145.
31. Wang, J. D., Susanto, C. O. N. (2023). Traffic flow prediction with heterogenous data using a hybrid CNN-LSTM model. *Computers, Materials & Continua, 76(3),* 3097–3112. https://doi.org/10.32604/cmc.2023.040914
32. Rodriguez, P., Wiles, J., Elman, J. L. (1999). A recurrent neural network that learns to count. *Connection Science, 11(1),* 5–40.
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30,* 6000–6010.
34. Hussain, B., Afzal, M. K., Ahmad, S., Mostafa, A. M. (2021). Intelligent traffic flow prediction using optimized GRU model. *IEEE Access, 9,* 100736–100746.
35. Yan, H., Ma, X., Pu, Z. (2021). Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems, 23(11),* 22386–22399.