Expert Systems With Applications 186 (2021) 115738

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

TrafficBERT: Pre-trained model with large-scale data for long-range traffic flow forecasting

KyoHoon Jin [a], JeongA Wi [b], EunJu Lee [a], ShinJin Kang [c], SooKyun Kim [d], YoungBin Kim [a,*]

[a] Department of Image Science and Arts, Chung-Ang University, Dongjak, Seoul 06974, South Korea
[b] Motion AI Team, Game AI Lab, NCSOFT, Seongnam-si, Gyeonggi-do 13494, South Korea
[c] School of Games, Hongik University, Sejong-si 30016, South Korea
[d] Department of Computer Engineering, Jeju National University, Jeju-si, Jeju-do 63243, South Korea

ARTICLE INFO

Keywords:
Traffic flow
Big data
Pre-trained model
BERT

ABSTRACT

Traffic flow prediction has various applications such as in traffic systems and autonomous driving. Road conditions have become increasingly complex, and this, in turn, has increased the demand for effective traffic volume predictions. Statistical models and conventional machine-learning models have been employed for this purpose more recently, deep learning has been widely used. However, most deep learning-based models require data additional to traffic information, such as information on adjacent roads or road weather conditions. Therefore, the effectiveness of these models is typically restricted to certain roads. Even if such information were available, there is a possibility of bias toward a specific road. To overcome this limitation, based on the bidirectional encoder representations from transformers (BERT), we propose trafficBERT, a model that is suitable for use on various roads because it is pre-trained with large-scale traffic data. Our model captures time-series information by employing multi-head self-attention in place of the commonly used recurrent neural network. In addition, the autocorrelation between the states before and after each time step is determined more efficiently via factorized embedding parameterization. Our results indicate that trafficBERT outperforms models trained using data for specific roads, as well as commonly used statistical and deep learning models, such as Stacked Autoencoder, and models based on long short-term memory, in terms of accuracy.

1. Introduction

The ease of collecting data has resulted in the accumulation of large volumes of traffic data that are utilized for different purposes in intelligent transportation systems (ITSs). Furthermore, traffic flow forecasting has become more important as road traffic conditions have become more complex. For instance, traffic conditions can be adjusted to optimize freight transport routes and increase transportation efficiency to enable the transportation of more goods in less time. Similarly, traffic forecasting for road sections is necessary for autonomous vehicles to operate efficiently. In many parts of the world, traffic forecasting is crucial for mitigating traffic congestion.

However, conventional traffic flow forecasting methods are based on limited data sources and traffic data. Another disadvantage of these methods is that a considerable amount of information is derived from a small volume of data. Furthermore, the computational performance of these methods is insufficient to support real-time operations; hence, statistical models or early machine-learning models have been extensively used. Conventional statistical time-series models, such as the autoregressive integrated moving average (ARIMA) and autoregressive conditional heteroskedasticity (ARCH) models, and machine learning models, such as the k-nearest neighbor (KNN) and support vector machine (SVM) models, are often used.

The relative ease of traffic data collection and advances in computing performance have facilitated data accumulation and accelerated computing operations. Furthermore, deep learning is increasingly applied to improve the performance of traffic flow forecasting beyond that of conventional statistical time-series models or early machine learning models. Initially, simple models, such as the multilayer perceptron model, were used. Recently, models based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been widely investigated.

Most researchers of natural language processing (NLP) have focused on developing natural language understanding models for diverse tasks. One such model is the bidirectional encoder representations from transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018),

* Corresponding author.
E-mail addresses: fhzh123@cau.ac.kr (K. Jin), jaywi@ncsoft.com (J. Wi), dmswn5829@cau.ac.kr (E. Lee), directx@hongik.ac.kr (S. Kang), kimsk@jejunu.ac.kr (S. Kim), ybkim85@cau.ac.kr (Y. Kim).

https://doi.org/10.1016/j.eswa.2021.115738
Received 12 July 2020; Received in revised form 6 June 2021; Accepted 5 August 2021
Available online 22 August 2021

a high-performance model used for training and fine-tuning, and is characterized by its ease of use in diverse applications. Owing to its exceptional features, it is used for a range of NLP tasks including text summarization (Miller, 2019) and sentence embedding (Reimers & Gurevych, 2019). In addition to NLP subdomains, the BERT is used in recommendation systems (Sun et al., 2019) as a substitute for RNN models, such as the gated recurrent unit (GRU) (Cho et al., 2014) model, and in other scientific applications. Its potential applicability in bioresearch has been demonstrated by the performance of BioBERT (Lee et al., 2019) and ClinicalBERT (Huang, Altosaar, & Ranganath, 2019). Despite its broad applicability across different domains, BERT has not yet been deployed in ITSs.

In this study, we aimed at utilizing various BERT characteristics for traffic flow forecasting. First, we used a bidirectional transformer structure similar to the BERT to predict the overall flow rather than each time step. In addition, unlike existing models that must be separately trained for each road, we increased the generalizability of the model by pre-training it using data from various road.

We herein propose trafficBERT, which integrates traffic flow data from an ITS with BERT by utilizing the aforementioned characteristics. Traffic flow data are time-series data used for tasks such as forecasting subsequent steps or sequences. Our decision to utilize the transformer-based BERT was motivated by a recent transformer-based model (Vaswani et al., 2017) the outperformed an RNN-based model regarding time-series prediction. This approach was expected to overcome the shortcomings of existing statistical methods, that are limited in their ability to process large-scale data. Furthermore, whereas existing models need to be separately trained for each road, the proposed model can successfully forecast traffic flows on various roads by fine-tuning a single model. We conducted experiments wherein we trained our model using existing traffic flow data and successfully verified its capability to forecast traffic flow using diverse traffic data as inputs. The proposed model was observed to be capable of transfer learning and delivered outstanding performance.

2. Related work

2.1. Research on deep learning in ITS

In previous studies on traffic flow forecasting, statistical time-series models or machine learning models have often been used. Time-series models can learn the characteristics of a time series from a given set of data and provide predictions, as in the case of the ARIMA algorithm. Early ARIMA models typically forecast extremely short sections (Ahmed & Cook, 1979). Subsequently, ARIMA models were diversified into seasonal ARIMA (Williams & Hoel, 2003) and ARIMA with exogenous variables (ARIMAX) (Williams, 2001), which encountered problems when large volumes of data were involved (Zhao et al., 2017). Models that utilize both the ARIMA and Kalman filter (KF) (Stathopoulos & Karlaftis, 2003; Xie, Zhang, & Ye, 2007) and those that employ historical averages (Kaysi, Ben-Akiva, & Koutsopoulos, 1993) have also been proposed. These models typically overestimate when large volumes of data are used (Miglani & Kumar, 2019).

Recently, deep learning models have been used to overcome this vulnerability (Liu, Li, Wu, & Li, 2018). In several subdomains of computer vision, superior prediction performance has been achieved in spite of large volumes of data by using CNNs (Krizhevsky, Sutskever, & Hinton, 2012). CNNs have also been used for traffic flow forecasting and are primarily characterized by their capability to detect topological localities. Relevant CNN-based models are fitted to forecast traffic flows with the traffic condition data of adjacent roads at approximately the same time (Liao, Chen, Hou, Xiong, & Wen, 2018; Yang et al., 2019; Zhang, Yu, Qi, Shu and Wang, 2019; Zhao et al., 2019). Another approach is to capture temporal and spatial correlations using Fully Convolutional Networks (FCN) (Zhang, Zheng, Sun and Qi, 2019). Apart from CNN-based models, those that incorporate long short-term memory (LSTM) or gated recurrent units (GRUs) have been extensively investigated. Some researchers have used LSTM to identify temporal–spatial correlations (Zhao, Chen, Wu, Chen and Liu, 2017) and others have used bidirectional LSTM for performance enhancement (Wang & Wang, 2018). Long-term and short-term forecasting have been attempted (Toqué, Khouadjia, Come, Trepanier, & Oukhellou, 2017).

Furthermore, models that can make predictions by combining the aforementioned methods to capture both spatial and temporal information have been studied (Wu, Tan, Qin, Ran, & Jiang, 2018; Zheng, Yang, Liu, Dai, & Zhang, 2019). To boost the traffic flow forecasting performance, researchers have adopted a range of techniques including graph convolution (Yu, Yin, & Zhu, 2019), Bayesian networks (Gu et al., 2019), wavenets (Wu, Pan, Long, Jiang, & Zhang, 2019), CNNs and RNNs (Lu, Rui, Yi, Ran, & Gu, 2020). Although deep learning models have been widely studied, the issue of their time-consuming training is yet to be addressed. Moreover, existing deep learning models are relatively inefficient because they require comprehensive information and separate training on each road for forecasting.

Therefore, we propose a model that efficiently deals with time-series data by using a self-attention mechanism-based transformer. Furthermore, we widened the scope of the proposed model by simplifying transfer learning.

2.2. Research on applications of BERT

We surveyed applications related to other time-series analyses to identify an appropriate technique for processing traffic flow time-series data. One of the techniques identified was NLP, wherein every word is analyzed. Many studies have been conducted to identify the correlations between words and their embeddings in vector spaces for large-scale datasets. In some studies, Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013) or GloVe (Pennington, Socher, & Manning, 2014) were used to determine the correlation between independent words and to train a model to place similar words closer together and dissimilar words farther apart in the vector space. Nevertheless, the aforementioned approach of embedding each word perceived as independent proved to be less applicable in traffic flow forecasting, where prior and subsequent road traffic flows must be observed.

Owing to advances in computer performance, it is now possible to learn the meaning of sentence units beyond word units and the overall contexts beyond the sentences. In one study, Embeddings from Language Models (ELMo) (Peters et al., 2018) was used to identify the characteristics of each sentence using bidirectional LSTM. This process had the advantage of using a pretrained model and the study led to increased use of the RNN-based model and further transformer-based studies. In OpenAI GPT2 (Radford et al., 2019), the characteristics of sentences were extracted using a unidirectional transformer, similar to ELMo. Two noteworthy points emerged from these two studies. The model does not merely learn words or sentences; it is effective at various NLP tasks when trained with contextual clues; pretraining the deep learning model using various types of data yields superior performance at various tasks.

In comparison with these studies that involved learning contextualized word representations, BERT delivered state-of-the-art performance and was pretrained using considerably more data. BERT uses a bidirectional transformer and outperforms many other models at various NLP tasks. Thus, we used various BERT characteristics for traffic flow forecasting. First, we used a bidirectional transformer structure similar to BERT to predict the overall flow rather than each time step. In addition, we increased the generalizability of the model by pretraining it using data for various roads, in contrast with existing models that have to be separately trained for each road.

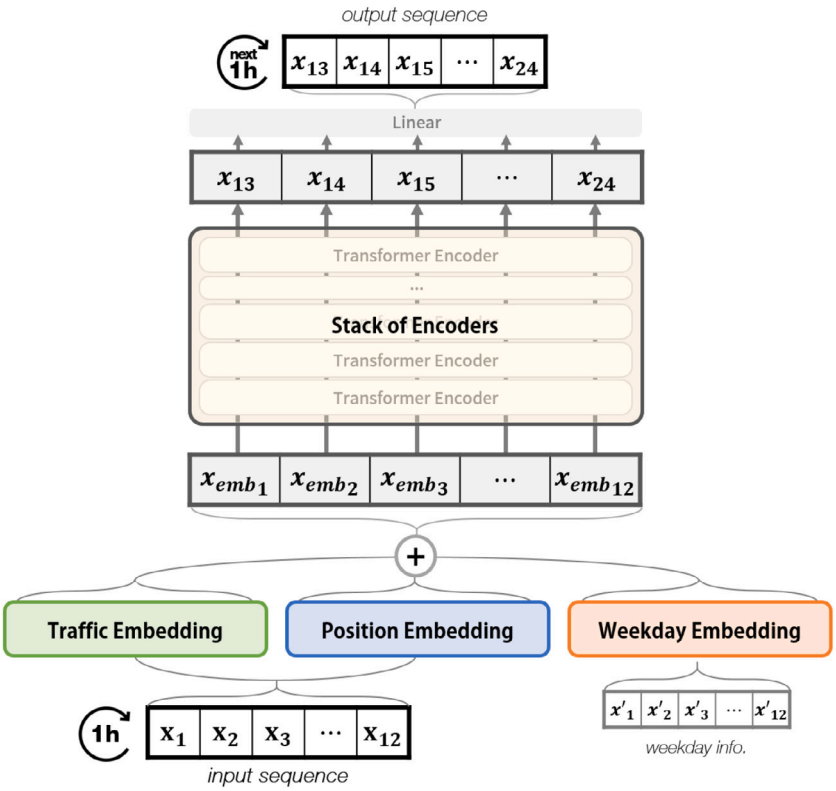

Fig. 1. Overall procedure for trafficBERT. The input embeddings were the sum of the traffic embedding, weekday embedding, and position embedding. The output time-step size was the same as that of the input time step. In this study, the features of 12 time steps (1 h) were input, and those in the subsequent 12 time steps (1 h) were predicted.

3. Methodology

To develop the proposed trafficBERT model and make it suitable for traffic flow forecasting tasks, we added and altered certain parts of BERT's structure. The structure of BERT, the basis of trafficBERT, is described in this section. The distinct parts of BERT are optimized for the given task. An overview of trafficBERT is presented in Fig. 1.

3.1. Weekday embedding & factorized embedding parameterization

The data contained in the PeMS and METR datasets represent the average vehicle speed $S = \{s_1, \ldots, s_N\}$, where the speed of the sensor on each road was measured at 5 min intervals. In every batch, we randomly selected a time step t, which became the first time step of the input data. The input was the specified time step T after t. Thus, the input data became $s_t, \ldots, s_{t+T}$. The output produced by BERT was of the same size as the input because has the same structure as Autoencoder (Yang et al., 2019). Furthermore, because BERT makes long-term rather than short-term predictions, the output consisted of time steps equal to the value of T after the specified t that is, $s_{t+T+1}, \ldots, s_{t+2T}$. In this study, we specified a value of 12 for the time step T. Thus, the input and output data are $s_t, \ldots, s_{t+12}$ and $s_{t+13}, \ldots, s_{t+24}$, respectively.

The embedding step in conventional BERT consists of token embedding, segment embedding, and position embedding, and the sum of these embeddings is obtained. However, given that these embeddings are intended for NLP tasks, we adopted a different approach in this study. First, conventional token embedding involves embedding techniques such as Word2Vec (Mikolov et al., 2013), whereas traffic flow data are continuous variables. Hence, we used a linear function, which we referred to as traffic embedding, for vectorization. Second, we focused on the general purposes of the proposed model and adopted 'weekdays' as the date-related information instead of segment embedding. We used weekdays, instead of weather or other date-related information, for weekday information to be used at any time without constraints. A linear function was used to fit the data for each day of the week to determine the embedding dimensions. Furthermore, three types of embeddings, including position embedding, were used. As in BERT, the embedded values were summed before use.

In the conventional BERT, the size of the input token embedding equals the size of the hidden layer, which is the second-best solution in A-Lite BERT (ALBERT) (Lan et al., 2019) from a modeling perspective. In modeling, WordPiece embedding is a context-independent representation, whereas the hidden-layer embedding is a context-dependent representation. The high performance of BERT is attributable to the context-dependent representation. As the objective of this study objective is to detect the overall flow rather than to increase the speed of individual sensors, we assigned more importance to the hidden-layer embedding. Accordingly, we scaled up the size of the hidden layer H rather than embedding the size of each sensor E, to focus on context identification. Each embedding method with the input sequence $S = \{s_1, \ldots, s_t\}$ and weekday information $X = \{x_1, \ldots, x_t\}$ is represented as follows:

$$\begin{aligned} E_{traffic}(S) &= W^M(W^T S) \\ E_{position}(S) &= W^M(W^P S) \\ E_{weekday}(X) &= W^M(W^W X) \\ E_{total} &= E_{traffic} + E_{position} + E_{weekday} \end{aligned} \tag{1}$$

In this equation, $E_{traffic}(S)$ represents the traffic embedding function, and $W^T \in \mathbb{R}^{1\times d_e}$ is the weight of traffic embedding with embedding dimension d_e. Furthermore, $E_{position}(S)$ represents the position embedding function, and $W^P \in \mathbb{R}^{1\times d_{max}}$ is the weight of the position embedding with the maximum position length d_{max}. $E_{weekday}(X)$ denotes the weekday embedding function, and $W^W \in \mathbb{R}^{7\times d_e}$ is the weight of the weekday embedding. A week contains seven days; hence, the initial dimension of W^W is seven. $W^M \in \mathbb{R}^{d_e\times d_{model}}$ is the weight that fits each embedding value to the dimension of the model d_{model}.

3.2. Multi-head self-attention

Current deep learning techniques for time-series prediction extensively use RNN-based deep learning models. To improve the performance of RNN models, various techniques have been developed, one of which is the attention mechanism (Bahdanau, Cho, & Bengio, 2014). The current attention mechanism applies attention only once to the entire dimension. However, in the multi-head attention employed in the transformer, attention is applied to the entire dimension multiple times by dividing the entire dimension by h and applying attention h times. Following the division and application of attention, the h results are concatenated to obtain a single vector. This is subsequently multiplied by a matrix to equalize the dimensions of themodel and the vector:

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(Head_1,\dots,Head_h)W^O \\ Head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model}\times d_k}$, and $W^0 \in \mathbb{R}^{hd_v\times d_{model}}$, respectively. In addition, the attention function is scaled dot-product attention, that is,

$$Attention(Q,K,V) = softmax(\frac{QK^\top}{\sqrt{d/h}})V \tag{3}$$

where h is introduced to avoid extremely small gradients and produce softer attention (Hinton, Vinyals, & Dean, 2015).

The attention layer is developed in the process, and the antecedent attention output is entered as the input for the following attention. This constitutes self-attention. As the process depends only on self-attention and excludes RNN structure operations, each layer involves fewer operations and as reported in transformer-related studies, the process can be parallelized to save time. Prior studies (Devlin et al., 2018; Vaswani et al., 2017) proved that this approach is effective for detecting the dependence of long-range data in a time series. Therefore, in this study, we used a multi-head self-attention transformer structure to forecast traffic flow. This structure is characterized by large volumes of data, a large number of operations, and high correlations in each time window.

3.3. Structure of BERT

The structure that we adopted for BERT is similar to that of transformer models (Vaswani et al., 2017) based on multi-head attention. The bidirectional encoder differentiates BERT from existing pre-trained language models (PLMs). Notably, BERT is distinct from other transformer models in terms of its embedding and pre-training methods.

Even before the emergence of BERT, language model pre-training was known to be effective for improving the performance of NLP tasks (Dai & Le, 2015). BERT is demonstrably superior to many other PLMs in two respects. First, it uses an encoder of bidirectional transformers. Although existing PLMs such as ELMo also use bidirectional models, these models are LSTM models the sequential dependency results in high operating loads and reduced operating speeds. Furthermore, although OpenAI GPT2 also uses a transformer, it is a unidirectional model and is inefficient for detecting traffic flows. BERT delivers improved performance with a bidirectional transformer. Second, BERT is pretrained with a larger and more diverse dataset.

To retain the features of BERT, we developed a model by stacking multiple layers of encoders for the transformer. We then induced the model to understand the overall traffic flow by integrating all the data. That is, we trained the model in a single step and refrained from separately using the data for each road. This enabled us to generate robust models capable of forecasting a variety of data.

The pre-training method also differed from that of the existing BERT, which is pre-trained using masked language modeling (MLM). This involves masking a random portion of the sequence data and predicting the value of the masked portion. Although this method may be useful in NLP, it has limited use in traffic flow prediction in which every time zone is equally important. Therefore, we calculated the output with the same time-step size as the input to make a long-range time prediction. In this study, sectioning was conducted in 12 steps. Each time step was 5-min long; therefore, when the model received input for one hour, it predicted the traffic flow for the next hour.

4. Experiment

4.1. Data

Similar to the approach using BERT, we pre-trained the model using diverse data to enhance its understanding of road traffic flows. Furthermore, we did not use any special data, but attempted to determine the performance of the model using the following three widely used datasets as general benchmarks for traffic flow forecasting.

- METR-LA: METR-LA comprises traffic information data collected in Los Angeles County using 207 loop detectors installed on highways in and around Los Angeles. The data were collected over four months (122 days), from 1 March to 30 June 2012, and the 30-s data samples in the dataset were aggregated into 5-min intervals.
- PeMS-L: A part of the performance measurement system (PeMS) data collected by various California transportation agencies (CalTrans). As aforementioned, 30-s data samples were aggregated into 5-min intervals. The data were measured at 1,026 stations in CalTrans District 7, in California, for more than two months (61 days), from 1 May to 30 June 2012.
- PeMS-Bay: A part of the PeMS data gathered by CalTrans. Using the same method as that used for PeMS-L, we obtained the PeMS-Bay data from 325 sensors installed in the Bay Area for five months (153 days), from 1 January to 31 May 2017.

In addition to the traffic speed, the PeMS and METR-LA datasets contain various other types of traffic-related data. We extracted the traffic speed and weekday information. In all the datasets, the traffic records were aggregated into 5-min intervals. The readings of each sensor were binned into 5-min chunks and subsequently averaged.

4.2. Experimental setting

In this study, we combined the three aforementioned datasets (METR-LA, PeMS-L, and PeMS-Bay) into a single dataset, which was then randomized to avoid bias from any dataset. We used 12 time steps (of 5 min each) to predict the traffic flow over 1 h increments; for example, the traffic flow from 14:00 to 15:00 was predicted by taking the traffic flow from 13:00 to 14:00 as input. Values were missing from all the data batches, and as these were time-series data, missing middle values were expected to affect the prediction output. For example, in the case of a continuous series of data (e.g., 87, 88, and 89), time-series information is available (i.e., valid data were present in increments of one step). However, in the event of missing values in a sequence (e.g., 87, 0, and 89), extracting time-series information was difficult. To solve this problem, extracted batches with more than one missing value were excluded from the training process. Finally, 20% of the data were randomly selected for exclusion, and the remaining data were combined and used to train the model. Half of the randomly excluded data were used as validation data for hyperparameter tuning and the remaining data were used for the performance test.

We used an RTX-2080 graphics processing unit (GPU) and a GTX-1080ti GPU with 8 and 11 GB of frame buffer memory, respectively, to train the models. To ensure that the structure of trafficBERT was identical to that of the BERT base, we used 12 layers in the experiment. We also used the Adam optimizer (Kingma & Ba, 2014). The learning rate was initially set to $1e{-}5$, which subsequently decreased by a factor of 0.5 at the third, sixth, and ninth epochs. Moreover, in deep

Table 1
RMSE, MAE, and MAPE for different models. We used the same hyperparameters as those used in the relevant studies. For all experiments, 12 observed data points were used to forecast the subsequent 12 time steps.

	RMSE	MAE	MASE	MAPE
ARIMA	15.50	9.49	4.08	23.28
SAE	13.27	12.58	1.26	21.2
LSTM	12.01	5.99	2.47	15.6
FC_LSTM	10.89	5.62	0.95	14.23
FC_GRU	10.72	5.73	0.96	14.53
trafficBERT	5.72	3.53	0.49	7.72

neural network learning, multiplication by large weights may lead to an excessive update step, as a result of which the algorithm may diverge inappropriately. To avoid such divergence, we used gradient clipping (Pascanu, Mikolov, & Bengio, 2013) and set the maximum norm to 5. To prevent overfitting, we used dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) at a rate of 0.2. In all the tests conducted, as mentioned earlier, a 60-min historical time window was used. In other words, 12 observed data points were used to forecast traffic flow over the subsequent 60 min.

4.3. Evaluation metrics

We evaluated the regression prediction results in terms of the root mean square error (RMSE), mean absolute error (MAE), mean absolute scaled error (MASE), and mean absolute percentage error (MAPE), all of which are commonly used metrics in traffic flow forecasting tasks. In the equations, the predicted and actual values are represented as $\hat{y}$ and y, respectively.

The RMSE is the square root of the ratio of the sum of squares of the deviations between the observed and true values to the number of observations.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}-y)^2} \tag{4}$$

The MAE is the average absolute error, which has the same unit as the data being measured.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}-y| \tag{5}$$

The MASE is the average absolute scaled error, a dimensionless quantity used as a scale-free error metric expressing each error as a ratio of the average error to a baseline. The MASE is an indicator of the extent to which the error changes over time.

$$MASE = \frac{MAE}{\frac{N}{N-1}\sum_{n=2}^{N}|y_n - y_{n-1}|} \tag{6}$$

Finally, the MAPE compensates for the drawbacks of scale-dependent errors such as the RMSE and MAE and facilitates the comparison of error proportions by dividing the fluctuations in the errors by the actual values.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{|\hat{y}-y|}{\hat{y}} \tag{7}$$

4.4. Results of traffic forecasting

The performance of trafficBERT was validated based on comparisons with the conventional statistical model, ARIMA (Kaysi et al., 1993) and the conventional deep learning models, LSTM and GRU, which are widely used to compare time-series data. In addition, we compared the proposed model with an SAE (Stacked AutoEncoder) model.

The SAE structure we used was proposed in a previous study (Jin, Xu, Wang, & Yan, 2018). We used only the 12-layer bidirectional version of the LSTM model. Instead of the simple LSTM, we used the seq2seq (Sutskever, Vinyals, & Le, 2014) structure comprising encoders and decoders for the FC_LSTM and FC_GRU, and employed the same number of layers for training. Specifically, 12 layers were used in total (i.e., six encoder layers and six decoder layers). In addition, we used the same embedding and hidden sizes as those used in trafficBERT. All the values were predicted based on the mean loss every 60 min (12-step value). The results are presented in Table 1.

The experimental results indicated that trafficBERT outperformed the other baseline models. It outperformed the ARIMA model by approximately 10, in terms of the RMSE. This observation is consistent with previous reports, which stated that ARIMA models are prone to overestimation when large-scale datasets are considered (Miglani & Kumar, 2019). In addition, compared to the deep learning models, the differences in the RMSE ranged from 5 to 7, indicating that the transformer-based model was able to detect long-range features more effectively than the other deep learning models. Furthermore, the conventional RNN-based models, LSTM and GRU, exhibited no significant differences in RMSE, suggesting that trafficBERT was likely to outperform other RNN-based models, including those with the simple recurrent units (Lei, Zhang, Wang, Dai, & Artzi, 2017).

We randomly selected sensors and dates from each dataset and schematized the values forecasted by trafficBERT to evaluate its performance quantitatively and qualitatively. As evident from Fig. 2, the values predicted by trafficBERT and the reference values exhibit similar trends, indicating that the distribution characteristics of the model's predictions are similar to those of the actual data. In addition, in certain time windows, trafficBERT accurately forecasted the road sections where traffic was abruptly blocked. Although trafficBERT was not provided with time information, it detected signs of gradual slowdowns prior to traffic jams, which mostly occurred during rush hours. However, sudden traffic jams, such as those shown in Fig. 2b or c, were attributable to by unexpected or less predictable problems such as traffic accidents and natural disasters. Although the decline was expected because it was well-supported by some trends in the data, the model did not predict the decline perfectly and exhibited a slight decrease in prediction accuracy . The prediction of these special situations is believed to require additional information, as noted in other studies.

4.5. Effects of transfer learning

Deep learning models are commonly trained for single tasks. When deep learning is used to forecast traffic flows, a model is typically trained to predict traffic conditions along a particular road. The methods of applying a single language model to various tasks have been continuously studied in the field of NLP, of which BERT is a representative model. Although BERT was designed to be pre-trained on various datasets and then fine-tuned for each task, it outperforms models trained for single tasks.

We conducted transfer learning experiments to determine the effects of pre-training trafficBERT with large-scale data similar to existing NLP approaches. In this experiment, we compared trafficBERT with group-constrained convolutional recurrent neural network (GCRNN) (Lin & Runger, 2017) models. First, we divided 10% of randomly selected data from the METR-LA dataset into testing datasets and used the remaining 90% of the METR-LA data to train the GCRNN models. In this experiment, we used two versions of trafficBERT: one trained on the entire combined dataset (PeMS-L, METR-LA, and PeMS-Bay) and the other trained only on the METR-LA data. The METR-LA data mentioned here refers to the data remaining after approximately 10% of the test data had been extracted, as mentioned earlier. Table 2 lists the results of the transfer learning experiments. A qualitative assessment of each method is shown in Fig. 3.

In the experiment, the GCRNN model outperformed trafficBERT trained using only the METR-LA data in the META-LA data, as indicated

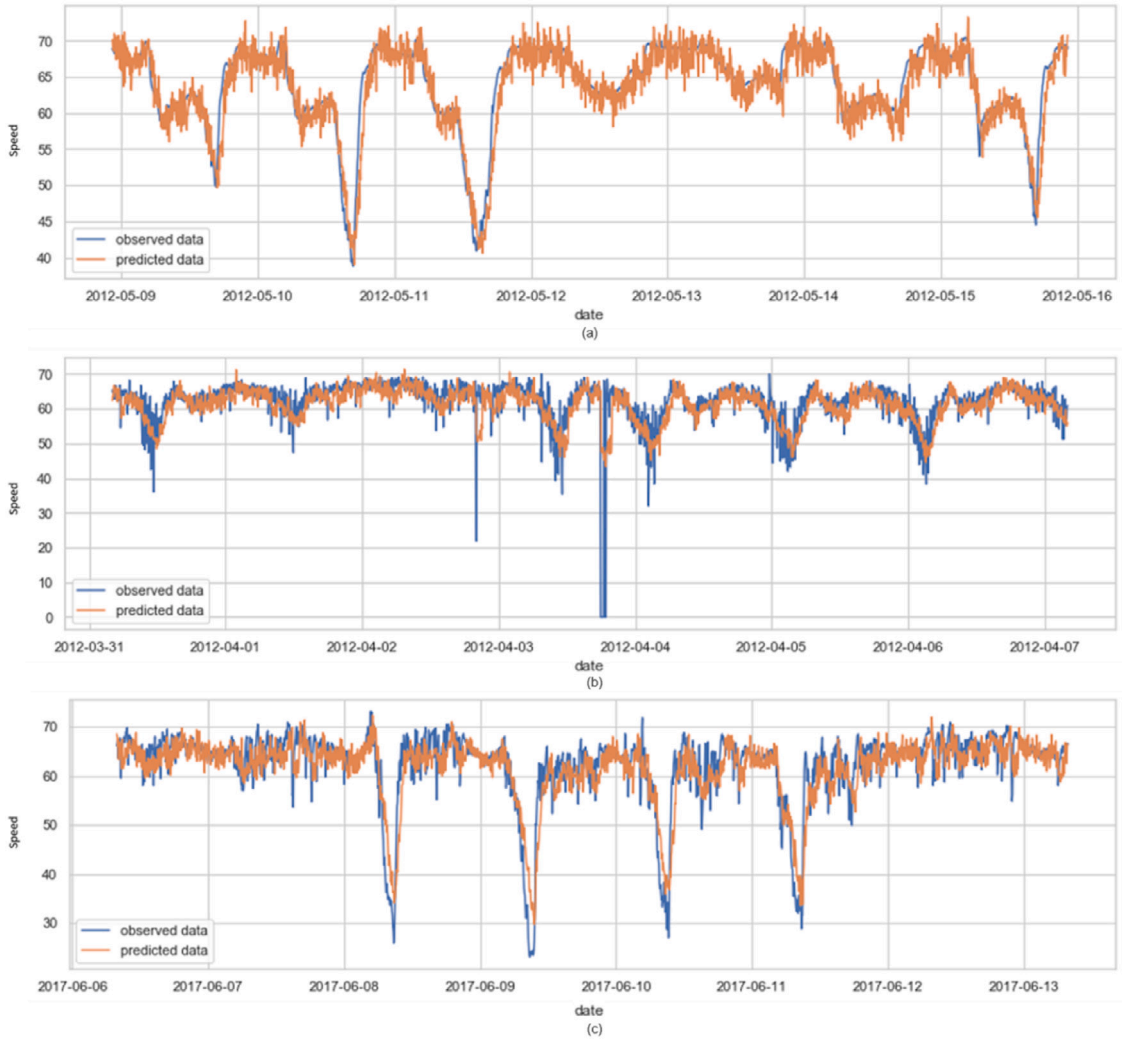

Fig. 2. Forecast of trafficBERT for randomly selected sensors and dates. From the top, (a) PeMS-L, (b) METR-LA, and (c) PeMS-Bay data. We predicted the values by entering the input data for an hour from a given time into the model. Then, we obtained a prediction value for the subsequent hour, entered the data for the next hour, and obtained another prediction value. This process was repeated, and the predicted values were plotted. In (b), the 0 toward the middle of the time period indicates missing data, rather than actual data.

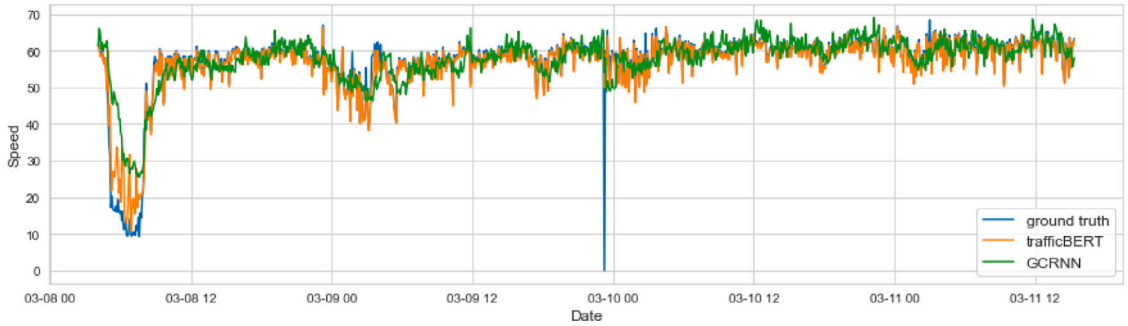

Fig. 3. Forecast by trafficBERT (full) and GCRNN for randomly selected sensors and dates from the METR-LA dataset. The 0 near the middle of the time period indicates missing data and not actual data.

by the RMSE values. However, when the METR and data from the other two datasets (PeMS-L, PeMS-Bay) were used for training, trafficBERT outperformed the GCRNN model. This observation indicates that pre-training with large-scale data was effective, not only for NLP but also for traffic flow forecasting. A model trained using only one type of data may be biased toward the data used for training because it has not been exposed to a sufficient variety of traffic situations. This limitation, which adversely affects the robustness of the model, can be mitigated by using more diverse data for pre-training such that the model encounters a wider variety of situations.

In the portion of data trending toward a decelerating period, as in the preceding section, the results confirmed that trafficBERT captured the trend of the data more effectively. Overall, the results showing deviation of predictions from correct values proved that the proposed model outperformed better than the other models. We attributed this factorized embedding parameterization. A close look at the flow of the

Table 2
RMSE, MAE, and MAPE for GCRNN and trafficBERT. The type of dataset used for training the models is stated in the parentheses next to each model.

	RMSE	MAE	MAPE
GCRNN (METR-LA) (Lin & Runger, 2017)	8.15	3.82	10.9
trafficBERT (METR-LA)	8.69	3.44	10.1
traffciBERT (FULL)	7.93	2.82	8.6

Table 3
Effect of layer number, model type, embedding type, and RMSE. Embed and Hidden represent the embedding size E and hidden layer size H, respectively.

#	Layer	Bidirectional	Embed	Hidden	Weekday embedding	RMSE
1	3	True	256	768	True	8.16
2	6	True	256	768	True	8.15
3	12	True	256	768	True	5.76
4	24	True	256	768	True	5.78
5	12	False	256	768	True	6.36
6	12	True	512	512	True	6.57
7	12	True	256	768	False	6.61
8	12	True	256	768	False (Weather)	6.62

real data shows severe fluctuations. To better capture this trend, we used a factorized embedding parameterization that was not used in the original BERT. This was interpreted as large deviations because our model captured rapidly changing trends better and, thus, performed slightly better in terms of accuracy.

4.6. Ablation studies

We conducted an ablation study to determine the effects of various modifications on the proposed trafficBERT model and other techniques and to ascertain whether our method was appropriate for the intended model design. The following aspects were investigated.

- Performance of trafficBERT relative to the number of layers
- Differences in the performance of uni- and bi-directional models
- Differences in performance depending on the application of factorized embedding parameterization and embedding type such as Weekday

Table 3 presents the results of the ablation study. Rows 1–4 indicate that the performance improves as the number of layers increased. Furthermore, compared with six layers, using 12 layers improves the performance significantly. However, the performance does not improve further when the number of layers was increased to 24. This observation suggests that the number of layers should not exceed 12.

Rows 3 and 5 indicate that the bidirectional model outperformed the unidirectional model under the same conditions when directionality was excluded. This indicates that unidirectionality alone cannot detect overall flows in trafficBERT, where the outputs include 12 steps instead of a sequence of single values. As mentioned in the literature on BERT (Devlin et al., 2018), bidirectional models can forecast overall flows more effectively than unidirectional models.

We determined the performance difference based on the factorized embedding parameterization and embedding type. Regarding factorized embedding parameterization, our approach was to train the models by increasing the embedding size while decreasing the size of the hidden layer. This enabled us to compare models with similar computational complexities. Rows 3 and 6 show the difference between applying factorized embedding parameterization and not doing so; as can be observed, there was a difference of approximately 0/8 between the RMSE values.

Finally, we determined the performance depending on the embedding type. We confirmed that the performance declined when weekday embeddings were not used. Rows 3 and 7 indicate that the performance improved when additional data, such as weekday information, were included in the model. We also conducted an experiment to determine if utilizing external data, such as adjacent road information or weather information, affected performance. Rows 3 and 8 indicate that weather information was added, instead of weekday embedding. The weather data were obtained from the National Oceanic and Atmospheric Administration and the National Weather Service Forecast Office. The weather information was used as the input, instead of the weekday embedding. Subsequently, the linear function was used to match the dimensions of the other embedding. This experiment shows that weather information, which is widely used as external data, does not improve trafficBERT.

5. Conclusion

We developed a BERT-based trafficBERT model and demonstrated its superiority over existing deep learning models by using experimental results.

Many researchers have sought to enhance the performance of their models by increasing their speed and incorporating diverse information. However, this approach for traffic flow forecasting requires separate training for each road, which degrades the efficiency of the models. The proposed trafficBERT model facilitates transfer learning and requires information only on the traffic speed and days of the week. Information on traffic flows on adjacent roads is not required. In addition to scalability, the proposed trafficBERT model facilitates easy use of additional features in different settings.

The proposed trafficBERT model demonstrated that pretraining a model with large volumes of data can improve traffic flow forecasting, even when there is no adjacent road or weather information. Although we employed widely used benchmark data for unbiased comparisons in this study, we observed that pretraining models using other available data, such as METR and PeMS, which have been recently used for traffic flow forecasting, improved the performance of the model on diverse roads. As aforementioned, although trafficBERT was outperformed by a specific model when it was trained using only METR-LA data, trafficBERT trained using two more datasets outperformed the GCRNN. From these results, it can be inferred that traffic flow prediction was improved when pretraining using large-scale data, similar to the commonly used PLM models in NLP. This demonstrates that the proposed model has diverse potential applicability. In addition, PLM models whose characteristics are different from those of BERT such as ERNIE (Zhang et al., 2019) and XLNet (Yang, Dai et al., 2019), are emerging. These models tend to be more prone to overfitting than BERT because they have a larger number of parameters to learn. However, different normalization techniques have been studied recently (Jang, Jin, An, & Kim, 2020; Verma et al., 2018; Yun et al., 2019). In combination with appropriate regularization methods, they may perform as well as the proposed trafficBERT model. Therefore, we expect that pretrained models with various characteristics can be applied to forecast traffic flow in the future.

CRediT authorship contribution statement

KyoHoon Jin: Conceptualization, Methodology, Software, Data curation, Validation, Writing - original draft, Writing - review & editing. **JeongA Wi:** Software, Writing - review & editing, Visualization. **EunJu Lee:** Software, Writing - review & editing. **ShinJin Kang:** Conceptualization, Writing - review & editing. **SooKyun Kim:** Conceptualization. **YoungBin Kim:** Conceptualization, Methodology, Writing - original draft, Supervision.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.eswa.2021.115738. JeongA Wi is an employee of NCSof.

Acknowledgments

This work was supported in part by Institute for Information & communications Technology Planning & Evaluation (IITP) through the Korea government (MSIT) under Grant No. 2021-0-01341 (Artificial Intelligence Graduate School Program (Chung-Ang University)), the National Research Foundation of Korea (NRF) through the Korea government (MSIT) under Grant No. NRF-2021R1F1A1061722 and Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020040186).

References

Ahmed, M. S., & Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using box-jenkins techniques, 722*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In *Advances in neural information processing systems* (pp. 3079–3087).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Gu, Y., Lu, W., Xu, X., Qin, L., Shao, Z., & Zhang, H. (2019). An improved Bayesian combination model for short-term traffic prediction with deep learning. *IEEE Transactions on Intelligent Transportation Systems*.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Huang, K., Altosaar, J., & Ranganath, R. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342.

Jang, S., Jin, K., An, J., & Kim, Y. (2020). Regional patch-based feature interpolation method for effective regularization. *IEEE Access*, *8*, 33658–33665.

Jin, Y., Xu, W., Wang, P., & Yan, J. (2018). SAE network: a deep learning method for traffic flow prediction. In *2018 5th international conference on information, cybernetics, and computational social systems (ICCSS)* (pp. 241–246). IEEE.

Kaysi, I., Ben-Akiva, M. E., & Koutsopoulos, H. (1993). *An integrated approach to vehicle routing and congestion prediction for real-time driver guidance, Vol. 1408*. Transportation Research Board.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2019). BioBERT: pre-trained biomedical language representation model for biomedical text mining. arXiv preprint arXiv:1901.08746.

Lei, T., Zhang, Y., Wang, S. I., Dai, H., & Artzi, Y. (2017). Simple recurrent units for highly parallelizable recurrence. arXiv preprint arXiv:1709.02755.

Liao, S., Chen, J., Hou, J., Xiong, Q., & Wen, J. (2018). Deep convolutional neural networks with random subspace learning for short-term traffic flow prediction with incomplete data. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1–6). IEEE.

Lin, S., & Runger, G. C. (2017). GCRNN: Group-constrained convolutional recurrent neural network. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(10), 4709–4718.

Liu, Z., Li, Z., Wu, K., & Li, M. (2018). Urban traffic prediction from mobility data using deep learning. *IEEE Network*, *32*(4), 40–46.

Lu, W., Rui, Y., Yi, Z., Ran, B., & Gu, Y. (2020). A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting. *IEEE Access*, *8*, 42042–42054.

Miglani, A., & Kumar, N. (2019). Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, *20*, Article 100184.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Miller, D. (2019). Leveraging BERT for extractive text summarization on lectures. arXiv preprint arXiv:1906.04165.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Stathopoulos, A., & Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C (Emerging Technologies)*, *11*(2), 121–135.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., et al. (2019). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1441–1450).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Toqué, F., Khouadjia, M., Come, E., Trepanier, M., & Oukhellou, L. (2017). Short & long term forecasting of multimodal transport passenger flows with machine learning methods. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)* (pp. 560–566). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., et al. (2018). Manifold mixup: Better representations by interpolating hidden states. arXiv preprint arXiv:1806.05236.

Wang, J., & Wang, H. (2018). One-step fabrication of coating-free mesh with underwater superoleophobicity for highly efficient oil/water separation. *Surface and Coatings Technology*, *340*, 1–7.

Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. *Transportation Research Record*, *1776*(1), 194–200.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, *129*(6), 664–672.

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121.

Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C (Emerging Technologies)*, *90*, 166–180.

Xie, Y., Zhang, Y., & Ye, Z. (2007). Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *22*(5), 326–334.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (pp. 5754–5764).

Yang, D., Li, S., Peng, Z., Wang, P., Wang, J., & Yang, H. (2019). MF-CNN: Traffic flow prediction using convolutional neural network and multi-features fusion. *IEICE Transactions on Information and Systems*, *102*(8), 1526–1536.

Yu, B., Yin, H., & Zhu, Z. (2019). ST-UNet: A spatio-temporal U-network for graph-structured time series modeling. arXiv preprint arXiv:1903.05631.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE international conference on computer vision* (pp. 6023–6032).

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129.

Zhang, W., Yu, Y., Qi, Y., Shu, F., & Wang, Y. (2019). Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transportmetrica A: Transport Science*, *15*(2), 1688–1711.

Zhang, J., Zheng, Y., Sun, J., & Qi, D. (2019). Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhao, Z.-Z., Chen, H.-P., Huang, Y., Zhang, S.-B., Li, Z.-H., Feng, T., et al. (2017). Bioactive polyketides and 8, 14-seco-ergosterol from fruiting bodies of the ascomycete Daldinia childiae. *Phytochemistry*, *142*, 68–75.

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75.

Zhao, W., Gao, Y., Ji, T., Wan, X., Ye, F., & Bai, G. (2019). Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access*, *7*, 114496–114507.

Zheng, Z., Yang, Y., Liu, J., Dai, H.-N., & Zhang, Y. (2019). Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3927–3939.