Journal of Database Management
Volume 34 • Issue 3

TransFusion Model Fusion Mechanism Based on Transformer for Traffic Flow Prediction

Xintong Song, Harbin Institute of Technology, China

Donghua Yang, Harbin Institute of Technology, China

Yutong Wang, Harbin Institute of Technology, China

Hongzhi Wang, Harbin Institute of Technology, China*

https://orcid.org/0000-0002-7521-2871

Jinbao Wang, Harbin Institute of Technology, China

Bo Zheng, CnosDB, China

ABSTRACT

In recent years, the problem of traffic congestion has become a hot topic. Accurate traffic flow prediction methods have received extensive attention from many researchers all over the world. Although many methods proposed at present have achieved good results in the field of traffic flow prediction, most of them only consider the static characteristic of traffic data, but do not consider the dynamic characteristic of traffic data. The factors that affect traffic flow prediction are changeable, and they will change over time. In response to this dynamic characteristic, the authors propose a model fusion mechanism based on transformer (TransFusion). The authors adopt two basic forecasting models (TCN and LSTM) as the underlying architectures. In view of the performance of different models on the traffic data at different times, the authors design a model fusion mechanism to assign dynamic weights to basic models at different times. Experiments on three datasets have proved that TransFusion has a significant improvement compared with basic models.

KEYWORDS

LSTM, Model Fusion Mechanism, TCN, Transformer

1. INTRODUCTION

The explosive growth of urban population and the increase of vehicles are likely to cause traffic congestion. Traffic accidents have become another turbulence factor in people's lives. At the same time, traffic congestion has caused a great burden on the environment. The traffic problem has received

DOI: 10.4018/JDM.325353

*Corresponding Author

considerable attention all around the world. There are many factors influencing traffic conditions. Because the traffic state is gathered by human activities, traffic conditions are different at different times and regions (such as the regular congestion the morning and evening traffic peaks, and the more vehicles in the center of the city). In addition to basic factors such as time and region, weather and solar terms are also important factors that affect traffic conditions. Complex factors make it difficult to alleviate traffic congestion. In order to solve the problem of traffic congestion, researchers from various countries have put forward the Intelligent Transportation System (ITS).

Intelligent transportation system (ITS) is a non-linear and time-variant system. Its technical core is traffic prediction. Traffic prediction is to predict the current or next traffic flow through the traffic flow at the previous moments. The status information mainly includes traffic flow, road structure, average vehicle speed, etc. The traffic flow information is particularly critical. Traffic flow refers to the number of vehicles passing through the current section of a highway in a unit period. The traffic flow can clearly see the congestion degree of the current section of a highway. In order to make more accurate prediction of traffic flow in the next time, predicting traffic flow requires not only knowing the traffic flow in the area at historical moments, but also combining local time, region, weather, solar terms and other factors.

The traffic prediction problem is a complex time varying problem. To tackle this problem, an increasing number of traffic conditions forecasting models have been proposed in relevant literature during the past several years, like ARIMA28, SVR9, SAE21, LSTM34, Conv-LSTM20. The spatial-temporal characteristic of traffic data is dynamically related. For example, during the peak period of traffic flow, the central area of the city is inconvenient for vehicles to enter and exit due to traffic congestion. There is little difference in traffic flow between the previous moment and the next moment. The temporal characteristic of traffic data has a greater impact on the predictive effect. At midnight, because the traffic flow of the entire city is small, in a certain period of time, vehicles can reach a farther location from the current location. Therefore, the traffic flow of the area is greatly affected by the traffic flow of the surrounding areas. It is difficult to find a single model with good performance to predict traffic flow at all times, which hampers the improvement of performance. Although some hybrid models simultaneously mine the spatial-temporal relevance of traffic data, the features they get are static, not dynamic. In order to better capture dynamic spatio-temporal characteristic, in this paper we propose a model fusion mechanism based on Transformer, referred to as TransFusion. We select some models that are general in spatio-temporal network data prediction as baseline models. The Transformer layer is used to assign different weights to different models at different times. Dynamic weight distribution can make full use of the advantages of each baseline model at different times to consistently outperforms other baselines. Besides, TransFusion is data-driven and don't need external information(e.g. location information of sensors, topological map of the road). The contribution of this paper can be summarized as follows:

- We propose a model fusion mechanism TransFusion to dynamically combine some simple models. TransFusion can extract the dynamic spatio-temporal relevance of traffic data by assigning appropriate weights to different models. TransFusion is data-driven and don't need external information(e.g. location information of sensors, topological map of the road).
- Experiments on three datasets demonstrate the superior performance of TransFusion for Traffic Flow Prediction.

2. RELATED WORK

Traffic flow forecasting can predict the traffic flow at the next moment by capturing the current flow information of the highway and combining regional time and geographical factors. It is of great significance for guiding traffic travel and solving the problem of heavy traffic. It has aroused extensive attention at home and abroad. There are many methods for traffic flow prediction, which can be divided into four categories. The first one is linear model (e.g., autoregressive moving average model ARMA, seasonal ARIMA, Kalman filter methods). The second category is nonlinear model (e.g., wavelet models [Yang & Hu, 2016] and chaotic theory models [Xue & Shi, 2008]). The third category is machine learning models (e.g., k-nearest neighbor models [Cai et al., 2016] and support vector regression [Castroneto et al., 2009; Cheng et al., 2017; Sun et al., 2015; Xiao et al., 2018]).

The fourth category is deep learning model. With the advent of deep learning, researchers have found that deep learning models have powerful capabilities that can greatly compensate for the shortcomings of traditional methods, and have achieved significant results in natural language processing (Li et al., 2019), image recognition (Wang et al., 2020) and other aspects. Some researchers use it for traffic flow prediction research. Lv et al. (2015) adopt stacked autoencoders to predict traffic flow, and find that the stacked autoencoder model is superior to the BP neural network model and the support vector machine (SVM) model. Huang et al. (2013) establish a traffic flow prediction model using deep belief networks and get good prediction results. Due to the powerful ability of RNN on time series data, many researchers use RNN's variant LSTM (LSTM solves the problem of RNN gradient disappearance and gradient explosion) to predict traffic flow. Zhao et al. (2017) establish LSTM-based traffic prediction model. Fu et al. (2016) use LSTM model and GRU model to establish a traffic flow prediction model.

Because the traffic flow prediction problem is affected by both temporal and spatial characteristic, it is difficult to achieve better prediction results using a single model. Some researchers use deep learning hybrid models to extract features from both time and space for prediction. Lin et al. (2019) proposed the SpAELSTM model and Wu et al. (2016) adopted a hybrid model of CNN and LSTM to predict traffic flow. Traffic data is extremely dynamic (at this moment it may be more affected by time factors and at another moment it may be more restricted by space factors), therefore, models with poor overall performance will also be better than splendid overall performance at some moments. Based on this dynamic influence, this paper proposes a dynamic model fusion mechanism (TransFusion) based on Transformer. TransFusion is composed of two layers from a bottom-up perspective: The first layer is Basic Layer. It consists of some basic models. The second layer is Transformer Layer. It assigns appropriate weights to basic models. The advantages of the model fusion mechanism are as follows:

- The network architecture makes full use of the dynamic characteristic of traffic data. It can make full use of the advantages shown by different models at different moments, thereby improving the accuracy of prediction.
- The architectures of the underlying models are quite different, which can fully complement each other's advantages. The model fusion mechanism has good flexibility and scalability. The basic models can be replaced.
- Transformer Layer can capture long-term dependence and performs well in hour-level traffic flow prediction.

3. METHODOLOGY

In this section, we introduce the design of Model Fusion Mechanism based on Transformer (TransFusion). Specifically, we first give the preliminaries of traffic flow prediction. Then, we briefly introduce the structure of several basic models. Finally, we elaborate the technical details of TransFusion.

3.1 Preliminaries

Traffic flow refers to the number of vehicles passing by a road section in a small interval. Traffic flow prediction refers to a certain area. According to its historical traffic flow data, it predicts the traffic flow of the area in the future for a period of time. The mathematical description of the traffic flow forecasting problem is as follows:

The traffic flow of location p at the current time t is $x_p\left(t\right)$, and the traffic flow at the next time is $x_p\left(t+1\right)$. Using the traffic flow data of n adjacent locations $\{p\_i \mid i = 1,2,3,\ldots,n\}$ (including location p) in the past N time periods $\left(t-N+1,t\right)$ to predict the traffic flow data at the next moment of location p is the traffic flow prediction. The input of the problem can be expressed as a two-dimensional space-time matrix:

$$x_{n\times N}\begin{bmatrix} x_1\left(t-N+1\right) & x_1\left(t-N+2\right) & \cdots & x_1\left(t-1\right) & x_1\left(t\right) \\ x_2\left(t-N+1\right) & x_2\left(t-N+2\right) & \cdots & x_2\left(t-1\right) & x_2\left(t\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n-1}\left(t-N+1\right) & x_{n-1}\left(t-N+2\right) & \cdots & x_{n-1}\left(t-1\right) & x_{n-1}\left(t\right) \\ x_n\left(t-N+1\right) & x_n\left(t-N+2\right) & \cdots & x_n\left(t-1\right) & x_n\left(t\right) \end{bmatrix} \tag{1}$$

where n represents the spot within the region and N represents the time. A simplified description of the problem is as follows(ρ is the predictive model):

$$x_p\left(t+1\right) = \rho\left(X_{n\times N}\right) \tag{2}$$

3.2 Basic Models

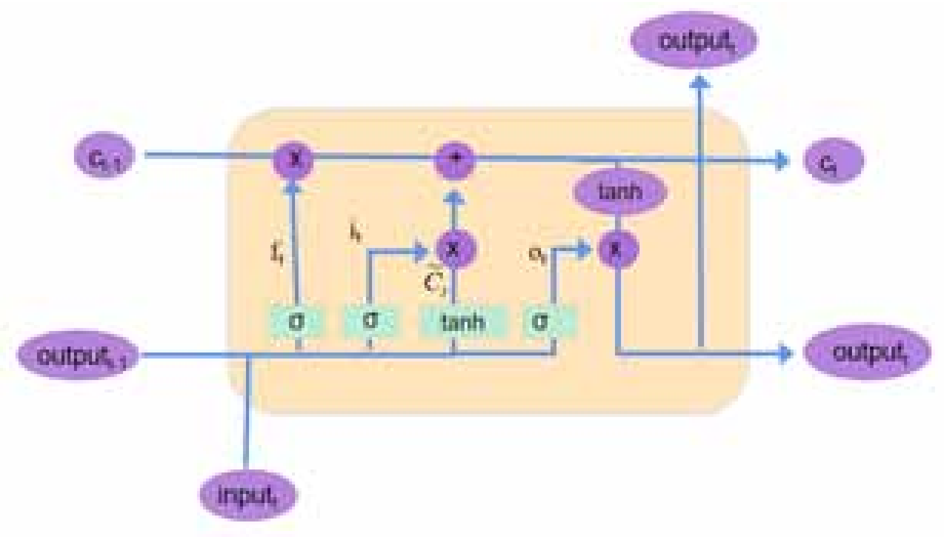
We choose two models (LSTM) that perform well in time series forecasting tasks as the basic models.

LSTM Due to the fully connected mode between layer and the layer (input layer, hidden layer, output layer) in the traditional neural network model, it is difficult to achieve better results for contextual sequence. The emergence of recurrent neural networks has improved this dilemma. The recurrent neural network stores the previous content in the current cell for calculation. The output of the previous cell and the output of the next cell are related. At the same time, the connection mode is changed between the nodes of the hidden layer. The output of the hidden layer is not only related to the input of the current layer, but also affected by the output of the hidden layer at the previous moment. Therefore, recurrent neural network (RNN) can effectively extract time series features.

Due to the chain derivation rule of RNN, if there is a minimum value in the weight matrix, the gradient will shrink exponentially after the matrix is multiplied N times. After a period of time, the gradient becomes 0, which is the disappearance of the gradient. If the value of the weight matrix is large, after N times of multiplication, the gradient explosion phenomenon will appear. The existence of gradient disappearance and gradient explosion restricts the development of RNN, making it difficult to learn long-term dependence. With the continuous advancement of deep learning research, a new type of neural network is proposed, namely the long and short-term memory network (LSTM). LSTM can solve the problem of gradient disappearance and gradient explosion. It has been widely used in time series feature extraction (Sulo et al., 2019) and natural language processing (Cai et al., 2019).

The structure of LSTM is shown in Figure 1. LSTM uses cell state to store long-term memory and uses three gates (forget gate, input gate, output gate) to control cell state. The output of the gate is a real vector between 0 and 1. When the output is 0, any vector multiplied by it will result in zero vector, which is equivalent to nothing passing. When the output is 1, there will be no change in any vector multiplied by it, which is equivalent to everything passing. Forget gate determines how much of the cell state C_{t-1} at the previous moment is retained to the current moment C_t. Input gate

Figure 1. The flamework of LSTM



determines how much of the input of node at the current moment is saved in the cell state C_t. Output gate controls how much of the cell state C_t is output to the current output value h_t of LSTM.

TCN Research has indicated that certain convolutional architectures can reach state-of-the-art accuracy in multiple sequence modeling tasks (Dauphin et al., 2017; Gehring, Auli, Grangier, & Dauphin, 2017; Gehring, Auli, Grangier, Yarats, et al., 2017), a generic temporal convolutional network (TCN)(Bai et al., 2018) was proposed. The TCN is based upon two principles: the fact that the network produces an output of the same length as the input, and the fact that there can be no leakage from the future into the past. The architecture of TCN can be expressed as:

$$\text{TCN} = \text{1DFCN} + \text{causalconvolutions}$$

. The TCN uses a 1D fully-convolutional network (FCN) architecture to achieve the first point. The TCN uses causal convolutions, convolutions where an output at time t is convolved only with elements from time t and earlier in the previous layer to achieve the second point.

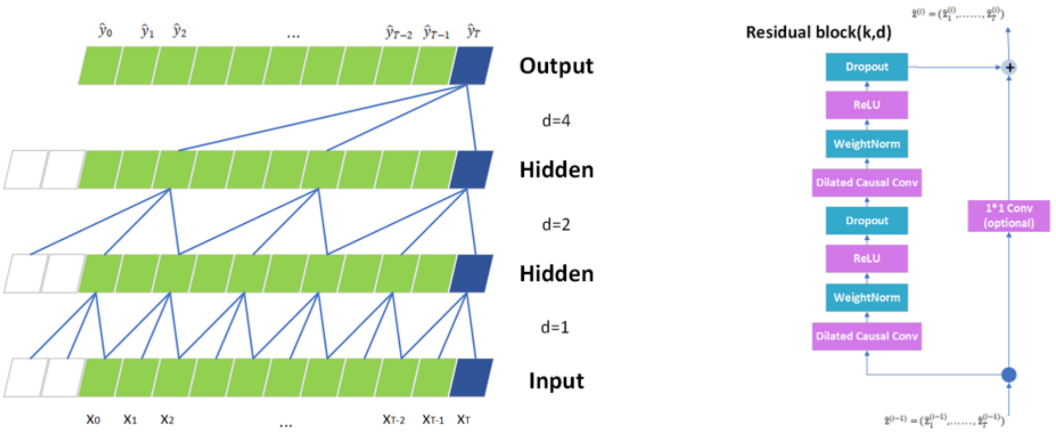
In order to solve the long-term dependence problem in sequence modeling tasks, TCN employ dilated convolutions to expand the receptive field. The convolution operation F on element s of the sequence is defined as

$$F\left(s\right) = \sum_{i=0}^{k-1} f\left(i\right) \cdot x_{s-d \cdot i} \tag{3}$$

where d is the dilation factor and k is the filter size. choosing larger filter sizes k and increasing the dilation factor d can expand the receptive field of TCN. The struture of dilated causal convolution is shown in Figure 2(a).

To maintain the stability of deeper and larger TCNs, a residual module is employed is adopted instead of a convolutional layer. A residual block add the input to outputs of the block. The residual block of TCN is shown in Figure 2(b). Within a residual block, the TCN has two layers of dilated causal convolution and the rectified linear unit (ReLU). Weight normalization and spatial dropout

Figure 2. The architecture of TCN


$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_{T-2}$ $\hat{y}_{T-1}$ $\hat{y}_T$
Output
d=4
Hidden
d=2
Hidden
d=1
Input
x_0 x_1 x_2 ... x_{T-2} x_{T-1} x_T
Residual block(k,d)
$\hat{z}^{(i)} = (\hat{z}_1^{(i)}, \ldots\ldots, \hat{z}_T^{(i)})$
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1*1 Conv (optional)
$\hat{z}^{(i-1)} = (\hat{z}_1^{(i-1)}, \ldots\ldots, \hat{z}_T^{(i-1)})$

(a) Dilated Convolutions

(b) Residual Block

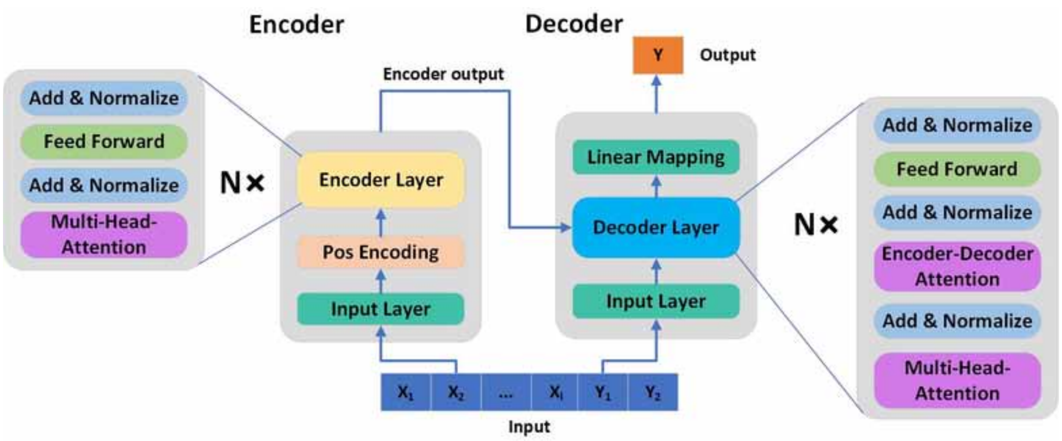
are adopted to normalization and regularization. Besides, the TCN adopts 1 ×1 convolution to ensure consistent input and output width.

3.3 TransFusion

Since the appearance of Transformer (Vaswani et al., 2017), it has been widely used in NLP (Devlin et al., 2018; Raffel et al., 2019; Zhou et al., 2018) and CV (Carion et al., 2020, Zheng et al., 2020, Zhu et al., 2020) tasks. Transformer does not process data in an ordered sequence manner. It processes entire sequence of data and uses self-attention mechanisms to learn dependencies in the sequence. Therefore, Transformer-based models can model complex dynamics of time series data. In order to extract dynamic spatialtemporal characteristic of traffic data, we add a dynamic model fusion mechanism to the basic models. Transfusion has made some changes on the basis of Transformer's architecture. It is shown in Figure 3.

TransFusion consists of encoder and decoder layers. The input of TransFusion is defined as X. X is composed of $\left[X1, X2, \ldots, Xl\right]$ ($Xi = \left[Xi1, Xi2, \ldots, Xij\right]$, j is the number of stations) and

Figure 3. Architecture of TransFusion



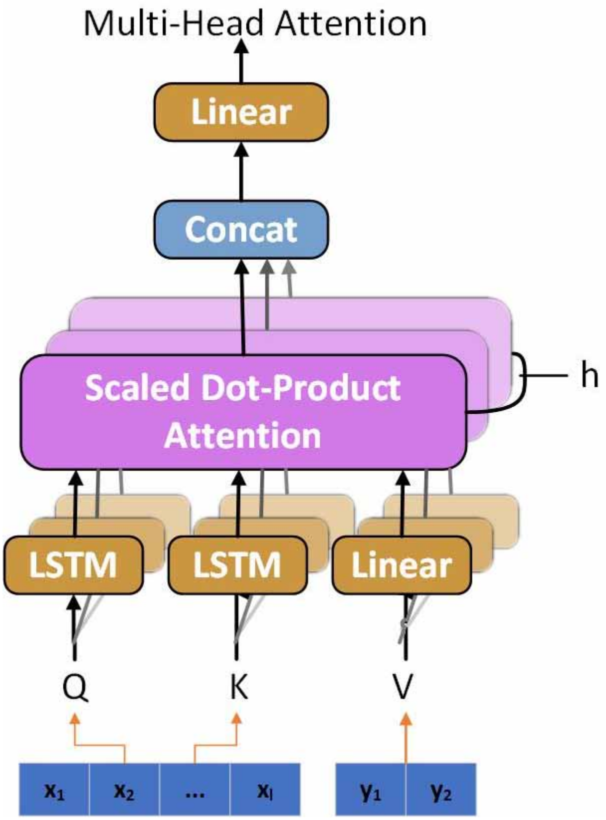
$\left[Y1, Y2, \ldots, Ys\right]$ ($Yi = \left[Yi1, Yi2, \ldots, Yij\right]$). $\left[X1, X2, \ldots, Xl\right]$ is spatio-temporal matrices of traffic data. $\left[Y1, Y2, \ldots, Ys\right]$ is the output of basic models. l is the length of sequence. s is the number of basic model. We concatenate $\left[X1, X2, \ldots, Xl\right]$ and $\left[Y1, Y2, \ldots, Ys\right]$ as the input of the TransFusion. First, we map the high-dimensional input X to a low-dimensional vector of dimension $dmodel$ through a fully-connected network. Second, we use positional encoding with sine and cosine functions to encode sequential information in the time series data by element-wise addition of the input vector with a positional encoding vector. Then, we input the resulting vector to N encoder layers. The output of the encoder layers and the input X of model are used as the input of the decoder layers. The output of the decoder layers becomes the final output of the model after dimension transformation.

Encoder: The encoder is composed of a stack of N identical layers. Each layer has two sub-layers: multi-head self-attention mechanism and fully connected feed-forward network. A residual connection is used between each layer. Each sub-layer is followed by a normalization layer.

Decoder: The encoder is also composed of a stack of N identical layers. Each layer has three sub-layers. The first one and the second one are same as the encoder. The decoder inserts a third sub-layer to apply self-attention mechanisms over the encoder output. A residual connection and a normalization layer follow each of sub-layers.

Multi-Head Attention: We employ the attention mechanism to assign dynamic weights to the prediction results of different basic models. Multi-Head Attention and Encoder-Decoder Attention are depicted in Figure 4. In the first encoder layer and decoder layer, $\left[x1, x2, \ldots, xl\right]$ is the input vector

Figure 4. Architecture of attention



after position encoding. In the other encoder layers and decoder layers, $\left[x1,x2,...,xl\right]$ is the output of the previous encoder layer and decoder layer. $\left[y1,y2\right]$ is the variant of the output of basic models $\left[Y1,Y2\right]$. They perform the same transformation as $\left[x1,x2,...,xl\right]$. In Multi-Head Attention, we employ spatio-temporal matrix as the queries and keys. In Encoder-Decoder Attention, we employ the output of encoder layers as the queries and keys. Combined with the spatio-temporal characteristic of traffic data, attention mechanism dynamically assigns weights to values. Besides, we extract temporal dependency with LSTM cells. We linearly project the values h times with different projections.

$$MultiHead\left(Q,K,V\right)=Concat\left(Att_{i=1,2,...,h}\left(LSTM_{i_q}\left(Q\right),LSTM_{i_k}\left(K\right),VW_i^V\right)\right)W^O \tag{4}$$

$$Att_i\left(Q,K,V\right)=soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where the projections are matrices $W_i^V \in R^{d_{model}\times d_v}$ and $W^O \in R^{hd_v\times d_{model}}$

4. EXPERIMENT

In this section, we will perform experiments to validate the performance of TransFusion. First, we will introduce the datasets used in experiments. Then, we will analyze the parameter sensitivities through experiments and prove the necessity of model fusion mechanism. Finally, we select several experiments with incremental and ensemble algorithms as the baselines for comparison in a series of experiments.

4.1 Datasets

We carried out comparative experiments on three real-world highway traffic datasets (I105-E, PEMSD4, PEMSD8). The datasets are collected from the Caltrans Performance Measurement Systems (PeMS) (Chao et al., 2000). The system has more than 39,000 detectors deployed on the highway in the major metropolitan areas in California. The traffic data are aggregated into every 5-minute interval from the raw data, which means there are 12 points in the flow data for each hour. We use traffic flow data from the past hour to predict the flow for the next hour. We apply max-min normalization. 60% of data is used for training, 20% are used for testing while the remaining 20% for validation.

I105-E: This traffic dataset contains traffic information collected from 24 detectors in the highway I105-E. The time span of this dataset is from May to June in 2014. The locations of the detectors in the highway I105-E are illustrated in Figure 5.

PEMSD4: It is the traffic data in San Francisco Bay Area, containing 3848 detectors on 29 roads. The time span of this dataset is from January to February in 2018.

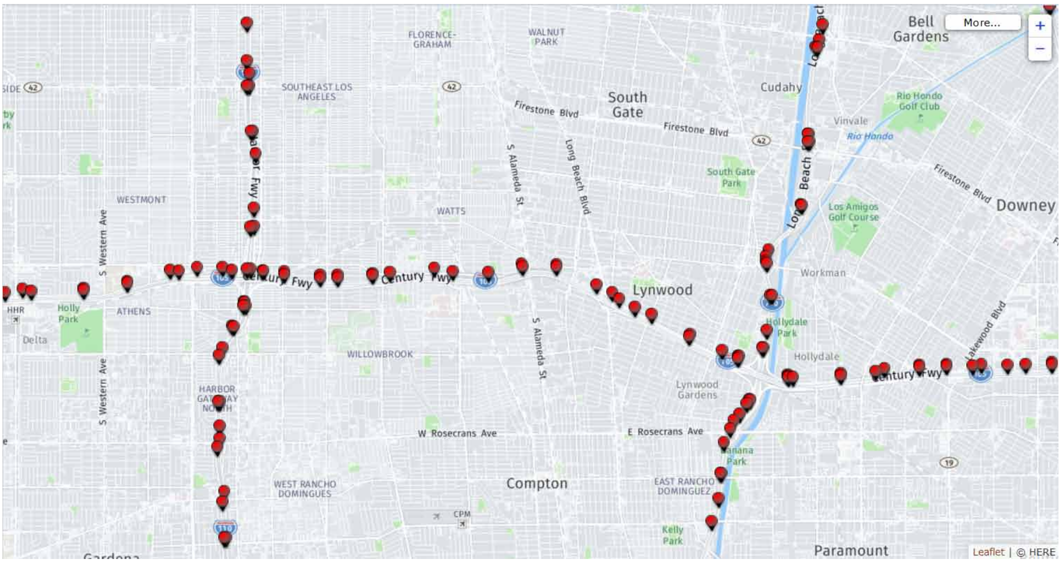
PEMSD8: This traffic dataset contains the traffic data of 1979 detectors on 8 roads in San Bernardino. The time span of this dataset is from July to August in 2016.

4.2 Evaluation Metrics

To evaluate the prediction results of the traffic flow prediction model, three evaluation functions are used, i.e., root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The evaluation criterion function is defined as follows:

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i-\widehat{y_i}\right)^2} \tag{6}$$

Figure 5. The locations of the detectors in the highway I105-E



$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \widehat{y_i}\right| \tag{7}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \tag{8}$$

where N denotes the indices of observed samples; x_i represents the ground truth, $\widehat{y_i}$ represents the predicted values. Missing values are excluded in calculating these metrics.

4.3 Baselines

To fully evaluate the performance of TransFusion mechanism, we compare our model with the following seven baseline methods:

- VAR (Zivot, 2020): Vector Auto-Regressive can capture the pairwise relationships among all traffic flow series. The number of lags is set to 12.
- SVR (1997): Support Vector Regression uses a linear support vector machine for regression tasks. It uses kernel trick which can map inputs to a feature space with high dimensional. Then, a non-linear regression could be converted into a linear function. The penalty term C is set to 0.1.
- LSTM (Hochreiter & Schmidhuber, 1997): Long Short-Term Memory Network (LSTM) has been widely applied in the time series forecasting area. Because of its memorial ability, LSTM performs well in the forecasting studies.
- SAE (Lv et al., 2015): Stacked AutoEncoder is composed of several auto-encoders stacked in series. It can convert complex input data into a series of simple high-order features. We use fully connected layers to construct the encoder and decoder.
- TCN (Bai et al., 2018): Temporal Convolutional Network uses causal convolution and residual module to model time series forecasting task. Because of the larger receptive field, it performs well in multiple fields.
- Seq2Seq (Sutskever et al., 2014): Sequence-to-sequence is also an encoder-decoder architecture, where the encoder is composed of a fully connected dense layer and a LSTM layer. It learns

from input and returns a sequence of encoded outputs and a final hidden state. The decoder is of the same structure as encoder.

- Conv-LSTM (Liu et al., 2017): Uses Convolutional Neural Networks(CNN) to extract spatial characteristic and uses Long Short-Term Memory Network(LSTM) to extract temporal characteristic.

4.4 Experiment Result

We split all datasets with ratio 6:2:2 into training sets, validation sets and test sets. One hour historical data is used to predict the next hour's data. It means that we use the past 12 continuous time steps to predict the future 12 continuous time steps. We compared TransFusion with seven baseline methods. Table 1 summarizes RMSEs, MAEs and MAPEs for each method, as well as relative performance gain with respect to VAR method.

In the experiment, we find that SAE(FC) based on fully connected layers has the problem of underfitting on complex datasets(PEMSD4 and PEMSD8). We adopt Seq2Seq(based on LSTM) on PEMSD4 and PEMSD8. It can be seen from Table 1 that TransFusion achieves the best performance in both two datasets in terms of RMSE and MAE. The comparison suggests that deep learning models overall outperform VAR and SVR for three evaluation metrics. Within the deep learning approaches, the performance of Seq2Seq is higher than TCN, ConvLSTM and LSTM. Seq2Seq can better extract the long-term dependence of feature. It is also applicable to hourly traffic flow forecasting and shows good performance on it. Conv-LSTM performs well on small datasets, but not on large datasets. In terms of RMSE, the TransFusion model outperforms Seq2Seq, with relative RMSE decrease of 3.8% on I105-E, 2.8% on PEMS4 and 8.5% on PEMS8, respectively. It also outperforms Conv-LSTM, with relative RMSE decrease of 4.9% on I105-E, 9.5% on PEMS4 and 17.2% on PEMS8. It suggests that TransFusion's attention mechanism can better capture dynamical spatiotemporal characteristic in the

Table 1. Traffic flow prediction performance comparison by three evaluation metrics

Model	I105-E			PEMSD4			PEMSD8		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
VAR	53.09	39.98	19.38	55.84	38.30	33.11	45.80	31.46	26.23
	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%
SVR	50.64	37.47	17.50	56.81	37.61	32.78	45.99	30.39	24.87
	-4.6%	-6.3%	-9.7%	+1.7%	-1.8%	-1.0%	+0.4%	-3.4%	-5.2%
LSTM	38.96	28.33	12.54	40.05	26.49	25.44	38.91	25.21	21.21
	-26.6%	-29.1%	-35.3%	-28.3%	-30.8%	-23.2%	-15.0%	-19.9%	-19.1%
TCN	40.15	28.74	13.14	42.10	28.09	23.70	42.15	28.06	23.44
	-24.4%	-28.1%	-32.2%	-24.6%	-26.7%	-28.4%	-8.0%	-10.8%	-10.6%
SAE(FC)	45.84	34.91	13.98	-	-	-	-	-	-
	-13.7%	-12.7%	-27.9%	-	-	-	-	-	-
Seq2 Seq	38.28	27.69	12.49	39.91	26.16	23.78	39.14	25.66	20.81
	-27.9%	-30.7%	-35.6%	-28.5%	-31.7%	-28.2%	-14.5%	-18.4%	-20.7%
Conv-LSTM	38.72	27.18	11.90	42.87	28.40	25.67	43.27	27.88	23.50
	-27.1%	-32.0%	-38.6%	-23.2%	-25.8%	-22.5%	-5.5%	-11.4%	-10.4%
TransFusion	36.82	27.10	12.04	38.79	25.81	26.37	35.81	23.74	20.46
	-30.6%	-32.2%	-37.9%	-30.5%	-32.6%	-20.4%	-21.8%	-24.5%	-22.0%

traffic data compared to the static characteristic extractd by Conv-LSTM. Because TransFusion don't use external road information, we don't compare it with methods based on graph neural network.

4.5 Influence of TransFusion

Transformer-based models have good effect on time series data (Wu et al., 2020). In order to prove the effectiveness of proposed TransFusion, we also compare TransFusion with Transformer-based models on same datasets.

As Table 2 shows, TransFusion has comprehensive performance improvement compared with Transformer-based model. It can be seen from Table 2 that Transformer does not extract the spatio-temporal features of traffic data well. With the aid of the underlying model, TransFusion can better solve this problem, achieving 5.5% 13% improvement than Transformer-based model.

Besides, we conduct experiments on PEMSD8 dataset to research the influence of different experiment settings(the number of encoder layers N and the number of heads h). As Figure 6 shows, N and h have a big impact on the performance of TransFusion. Although searching suitable settings is important for TransFusion, the performance of TransFusion is better than the basic models under many settings.

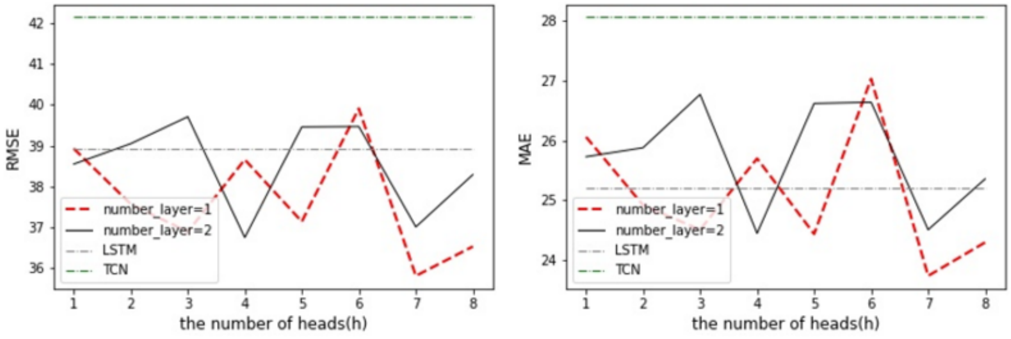
5. CONCLUSION

In this paper, we propose a novel Transformer-based model fusion mechanism called TransFusion for traffic flow prediction. We employ two traditional models with good performance (TCN and LSTM) as basic models. Besides, we use Transformer-based structure to assign dynamic weights at different times. Finally, we conduct extensive experiments on three real-world datasets and the results show that our proposed method is superior to several baseline models.

Table 2. TransFusion and transformer-based model

Model	I105-E			PEMSD4			PEMSD8		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Transformer	38.98	28.78	13.40	41.46	28.21	30.07	38.33	26.31	23.52
	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%	-0.0%
TransFusion	36.82	27.10	12.04	38.79	25.81	26.37	35.81	23.74	20.46
	-5.5%	-6.2%	-10.1%	-6.4%	-8.5%	-12.3%	-6.6%	-9.8%	-13.0%

Figure 6. The influence of different settings



(a) RMSE comparison of PEMSD8

(b) MAE comparison of PEMSD8

REFERENCES

Bai, S., Kolter, J.Z., & Koltun, V. (2018). *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*. Semantic Scholar.

Bo, Z. (2002). Arma-based traffic prediction and overload detection of network. *Journal of Computer Research and Development.*

Cai, L., Zhou, S., Yan, X., & Yuan, R. (2019). A stacked bilstm neural network based on coattention mechanism for question answering. *Computational Intelligence and Neuroscience*, *2019*(9), 1–12. doi:10.1155/2019/9543490 PMID:31531011

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C, Emerging Technologies*, *62*, 21–34. doi:10.1016/j.trc.2015.11.002

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *Endto-end object detection with transformers*. arXiv.

Castroneto, M., Jeong, Y., Jeong, M., & Han, L. D. (2009). Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications*, *36*(3), 6164–6173. doi:10.1016/j.eswa.2008.07.069

Chao, C., Petty, K., & Skabardonis, A. (2000). Freeway performance measurement: Mining loop detector data. Transportation Research Record Journal of the Transportation Research Board (1748).

Cheng, A., Jiang, X., Li, Y., Zhang, C., & Zhu, H. (2017). Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method. *Physica A*, *466*, 422–434. doi:10.1016/j.physa.2016.09.041

Dauphin, Y.N., Fan, A., Auli, M., & Grangier, D. (2017). *Language modeling with gated convolutional networks*. MLR Press.

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

Fu, R., Zhang, Z., & Li, L. (2016). *Using lstm and gru neural network methods for traffic flow prediction*. IEEE.

Gehring, J., Auli, M., Grangier, D., & Dauphin, Y.N. (2017). *A convolutional encoder model for neural machine translation*. Springer.

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y.N. (2017). *Convolutional sequence to sequence learning*. Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735 PMID:9377276

Huang, W., Hong, H., Li, M., Hu, W., Song, G., & Xie, K. (2013). *Deep architecture for traffic flow prediction*. Springer.

Li, P., Li, X., & Pan, H. (2019). Text-based indoor place recognition with deep neural network. *Neurocomputing*.

Lin, F., Xu, Y., Yang, Y., & Ma, H. (2019). A spatial-temporal hybrid model for short-term traffic prediction. *Mathematical Problems in Engineering*, 1–12. doi:10.1155/2019/4858546

Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017). Short-term traffic flow prediction with conv-lstm. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1–6). IEEE. doi:10.1109/WCSP.2017.8171119

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research.*

Sulo, I., Keskin, S. R., Dogan, G., & Brown, T. (2019). Energy efficient smart buildings: Lstm neural networks for time series prediction. In *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*. Research Gate.

Sun, Y., Leng, B., & Guan, W. (2015). A novel wavelet-svm short-time passenger flow prediction in beijing subway system. *Neurocomputing*, *166*, 109–121. doi:10.1016/j.neucom.2015.03.085

. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, *28*(7), 779–784.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems* (pp. 3104–3112). MIT Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need.* NIPS.

Wang, X., Huang, G., Ma, C., Tian, W., & Gao, J. (2020). Convolutional neural network applied to specific emitter identification based on pulse waveform images. *IET Radar, Sonar & Navigation*, *14*(5), 728–735. doi:10.1049/iet-rsn.2019.0456

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. doi:10.1061/(ASCE)0733-947X(2003)129:6(664)

Wu, N., Green, B., Xue, B., & O'Banion, S. (2020). *Deep transformer models for time series forecasting: The influenza prevalence case*. Semantic Scholar.

Wu, Y., & Tan, H. (2016). *Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework*. Computer Vision and Pattern Recognition.

Xiao, J., Wei, C., & Liu, Y. (2018). Speed estimation of traffic flow using multiple kernel support vector regression. *Physica A*, *509*, 989–997. doi:10.1016/j.physa.2018.06.082

Xue, J., & Shi, Z. (2008). Short-time traffic flow prediction based on chaos time series theory. *Journal of Transportation Systems Engineering and Information Technology*, *8*(5), 68–72. doi:10.1016/S1570-6672(08)60040-9

Yang, H., & Hu, X. (2016). Wavelet neural network with improved genetic algorithm for traffic flow time series prediction. *Optik (Stuttgart)*, *127*(19), 8103–8110. doi:10.1016/j.ijleo.2016.06.017

Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). Lstm network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75. doi:10.1049/iet-its.2016.0208

Zheng, S., Lu, J., Zhao, H., Zhu, X., & Zhang, L. (2020). *Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers.* ACL Anthology.

Zhou, T., Jiang, D., Lin, Z., Han, G., Xu, X., & Qin, J. (2019). Hybrid dual kalman filtering model for short-term traffic flow forecasting. *IET Intelligent Transport Systems*, *13*(6), 1023–1032. doi:10.1049/iet-its.2018.5385

Zhou, X., Li, L., Dong, D., Yi, L., & Wu, H. (2018). Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (*Volume 1*: Long Papers).* ACL Anthology. doi:10.18653/v1/P18-1103

Zivot, E. (2020). *Vector Autoregressive Models for Multivariate Time Series*. University of Washington.

Zhu, X., Su, W., Lu, L., Li, B., & Dai, J. (2020). *Deformable detr: Deformable transformers for end-to-end object detection*. arXiv.

Donghua Yang received his B.S., M.S., and Ph.D. degrees from Harbin Institute of Technology in 1999, 2003 and 2008 respectively. He is a vice professor of the Faculty of Computing and the Center of Analysis, Measurement and Computing, Harbin Institute of Technology, Harbin. His research interests include big data management and analytics. Educational Background Ph.D (Jul. 2008) Department of Computer Science & Technology, Harbin Institute of Technology Dissertation: Research on Query Processing on XML Data. Apr. 2008. (this dissertation won the Outstanding PHD Dissertation of China Computer Federation) Awards: IBM PHD Fellowship, 2007 China's Excellent Database Engineer, 2006 Microsoft Fellowship Award, 2005 B.Se. (Jul. 2001) Department of Computer Science & Technology, Harbin Institute of Technology Research Interest Big data management, Data Quality, Graph Data Management Field Experiences 2015-present Professor, Department of Computer Science and Technology, Harbin Institute of Technology 2010-2015 Associate Professor, Department of Computer Science and Technology, Harbin Institute of Technology 2008-2010 Lecturer, Department of Computer Science and Technology, Harbin Institute of Technology 2004-2005 Research Associate, Department of Computer Science and Engineer, University of New South Wales 2006-2007 Intern, School of Computer, National University of Singapore Professional Services PC Member: KDD 2016, DASFAA 2016 PC Chair: ICYCSEE 2016.