Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-023-08672-1

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

A Deep Ensemble Approach for Long-Term Traffic Flow Prediction

Nevin Cini[1,2] · **Zafer Aydin**[2]

Received: 3 May 2023 / Accepted: 21 December 2023

Abstract

In the last 50 years, with the growth of cities and increase in the number of vehicles and mobility, traffic has become troublesome. As a result, traffic flow prediction started to attract attention as an important research area. However, despite the extensive literature, traffic flow prediction still remains as an open research problem, specifically for long-term traffic flow prediction. Compared to the models developed for short-term traffic flow prediction, the number of models developed for long-term traffic flow prediction is very few. Based on this shortcoming, in this study, we focus on long-term traffic flow prediction and propose a novel deep ensemble model (DEM). In order to build this ensemble model, first, we developed a convolutional neural network (CNN), a long short-term memory (LSTM) network and a gated recurrent unit (GRU) network as deep learning models, which formed the base learners. In the next step, we combine the output of these models according to their individual forecasting success. We use another deep learning model to determine the success of the individual models. Our proposed model is a flexible ensemble prediction model that can be updated based on traffic data. To evaluate the performance of the proposed model, we use a publicly available dataset. Experimental results show that the developed DEM model has a mean square error of 0.06 and a mean absolute error of 0.15 for single-step prediction; it shows that achieves a mean square error of 0.25 and a mean absolute error of 0.32 for multi-step prediction. We compared our proposed model with many models in different categories; individual deep learning models (i.e., LSTM, CNN, GRU), selected traditional machine learning models (i.e., linear regression, decision tree regression, k-nearest-neighbors regression) and other ensemble models such as random-forest regression. These results also support the claim that ensemble learning models perform better than individual models.

Keywords Deep learning · Traffic flow prediction · Ensemble learning · Long short-term memory · Convolutional neural networks · Gated recurrent unit

1 Introduction

Traffic congestion causes many problems that we can examine under different headings such as economic, environmental and social. Among them, the most emphasized is the increase in cost with the lengthening of the travel time. These two key issues lead to the emergence of other problems. For example, prolonged travel time causes social and psychological problems, environmental problems such as noise pollution and even accidents from time to time. Although increasing the cost of travel is an economic problem, increasing fuel consumption also leads to environmental problems such as air pollution [1].

Decision makers who reach information about when and where traffic congestion may occur with traffic flow forecasting can direct drivers to safer roads so that resources can be used more efficiently. With a more effective planning, it is possible to use public transportation more efficiently as well. In this way, the environmental impact caused by traffic can be reduced. For this reason, traffic flow forecasting is of key importance in controlling traffic congestion and solving many problems that may occur, and is an indispensable component for intelligent transportation systems [2].

That is why, in recent years, many studies in this area have focused on developing reliable and realistic traffic flow prediction models using the latest technologies [3–5]. However, most of these studies have presented short-term traffic flow

✉ Nevin Cini
nevin.cini@agu.edu.tr

Zafer Aydin
zafer.aydin@agu.edu.tr

1 Artificial Intelligence Research and Development Center, Şanlıurfa, Turkey

2 Department of Computer Engineering, Abdullah Gul University, Kayseri, Turkey

Published online: 27 January 2024

prediction models [6]. Few of the proposed forecasting models are capable of long-term forecasting. However, long-term forecasting is as important and useful as short-term forecasting [1, 7, 8]. Furthermore, long-term traffic flow forecasting is of practical importance for decision makers. An accurate forecast model will facilitate traffic management even during the rush hours, and will enable effective measures to be taken by informing in advance of possible negative events.

However, long-term forecasting is a challenging task. This is due to the stochastic nature of the dynamics that make up the traffic flow data, which is nonlinear and contains complex dependencies [9]. It is also not identical in both temporal and spatial dimensions. Modeling dynamic temporal and spatial dependencies for traffic flow prediction is very burdensome and arduous. These complex dependencies increase in number and become more and more complex in long-term predictions. As the forecast horizon increases, even in the best models, the prediction quality decreases and the average error increases [10].

As a result, reliable long-term prediction becomes a difficult task, and it is almost impossible to model long-term dependencies of traffic flow with simple and traditional prediction models [8, 11, 12].

In this study, we propose a deep learning-based ensemble framework for long-term traffic flow prediction. While deep learning (DL) models can learn dynamic and complex dependencies of traffic data better than traditional learning algorithms, ensemble learning (EL) provides flexibility by increasing the generalization ability of the final model. Because many different predictive models collaborate to solve the given problem in ensemble learning, it is often expected that the ensemble model will exceed the predictive success of a single model.

The most important feature of the proposed EL model is that we employ three different DL models (i.e., CNN, LSTM and GRU) as base learners. This increases model diversity so that a failure of one model can be compensated by another model. As shown in Fig. 1, the performances of all three models change as traffic conditions change. From this figure, it is clear that we cannot achieve the best prediction performance with a single model. Because each model has strengths and weaknesses, the contribution of the base models to the final prediction result cannot be equal. In a successful ensemble model, a base model with high predictive performance is expected to contribute more to the final result than less successful models. In our ensemble model, we have developed a meta-learner to provide this. Owing to this meta-learner, we have dynamically weighted the base models, that is, we have ensured that each model contribute in the final prediction result according to its current prediction performance. We leverage this capability of ensemble learning to improve long-term prediction accuracy. In order to assess the accuracy, we conducted several experiments, in

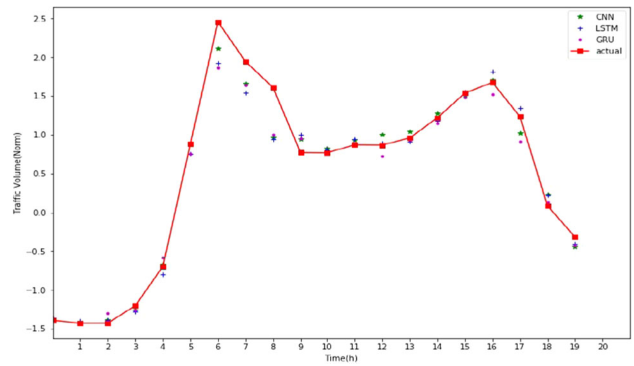

Fig. 1 Different performances of base learners

which we compared the proposed model with widely used prediction models.

There are three main contributions of this study:

- In this study, we proposed a fully DL-based ensemble learning framework for long-term traffic flow prediction.
- We used three different DL models as base learners. In the model we developed, we use LSTM and GRU together. We have not come across a model in the literature that uses these two techniques together. Since these two techniques are versions of recurrent neural networks, it is not preferred to use them together in a prediction model. However, although these two techniques are similar to each other, their performances are quite different as seen in Fig. 1. So where one fails, the other can be quite successful. For this reason, we preferred to use these two techniques together.
- We use deep learning architectures in our model, both as base learners and the meta-learner. Thanks to a feed forward neural network (FFNN), which is the most basic deep learning technique, we decide the weights of the base learners. We train a feed forward neural network as a meta-learner in order to obtain the final prediction result. In this way, we ensured that the base learners dynamically contribute in the final prediction result according to their prediction success (more successful ones contribute more, less successful ones contribute less).

The paper is organized as follows: In Sect. 2, we provide a background section. We explain the related technologies in Sect. 3. A brief overview of current literature on traffic flow prediction is provided in Sect. 4. In Sect. 5, we introduce the details of our deep learning-based ensemble framework. Then, we present dataset, preprocessing steps, experimental results and discussion in Sect. 6. Section 7 includes conclusion and future work.

2 Background

In this section, we will define the traffic flow prediction problem.

2.1 Traffic Flow Prediction

Traffic flow refers to the number of vehicles passing a certain road section per unit time. This data is collected automatically, usually with the help of sensors. Since vehicles can only move on the roads prepared for them, accurate estimation of the traffic flow in a certain area prevents possible congestion and ensures more efficient use of the roads [1, 2, 4].

2.2 Problem Formulation

Traffic flow forecasting models are often based on a simple assumption: the future depends on the past. In other words, data that generated traffic conditions in the past will affect current and future traffic situation. Therefore, continuity of data is important. Traffic flow prediction is a time series problem, and as with all time series problems, past values are used as target function parameters in the traffic flow estimation problem. In other words, the target/prediction value at time T_n becomes one of the target function parameters at time $T_{(n+1)}$. This is for single-step prediction. In multi-step prediction, more than one value at consecutive time steps participates in the process at the same time. To formulate this problem mathematically, we use the notation f_t^i to define traffic flow from station i at time t. In order to extract spatial and temporal features of traffic flow here, we construct spatial-temporal feature matrix as follows:

$$f_t^s = \begin{bmatrix} f_1^1 & f_2^1 & f_3^1 & \dots & f_t^1 \\ f_1^2 & f_2^2 & f_3^2 & \dots & f_t^2 \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ f_1^s & f_2^s & f_3^s & \dots & f_t^s \end{bmatrix} \tag{1}$$

Here, s denotes the number of stations. We construct this flow matrix with temporal information horizontally and spatial information vertically. In the next step, we can formulate the traffic flow prediction problem as follows:

$$\begin{bmatrix} f_{t-\beta}^d \\ f_{t-(\beta-1)}^d \\ f_{t-(\beta-2)}^d \\ . \\ . \\ . \\ f_{t-1}^d \end{bmatrix}^{\mathrm{T}} \xrightarrow{\theta} \begin{bmatrix} f_t^d \\ f_{t+1}^d \\ f_{t+2}^d \\ . \\ . \\ . \\ f_{t+(h-1)}^d \end{bmatrix}^{\mathrm{T}} \tag{2}$$

Here, since we use historical flow data to predict future flows, the matrix on the left hand side represents historical flow data and the matrix on the right hand side represents prediction values. The traffic flow prediction model is represented by a prediction function, which is represented by θ. f^d denotes traffic flow from station d. β is the looked-back steps, and h is the prediction horizon.

2.3 The Differences Between Short- and Long-Term Prediction

In fact, the difference between the short- and long-term forecast goes far beyond the period we determine with only the prediction horizon. In the literature, long-term forecasting is categorized as predicting an hour later or a few steps later (usually five steps or more), while short-term forecasting is defined as predicting one step or a few minutes later. Here, we can say that a categorization based on this definition is not reliable due to the lack of a consensus in terms of the prediction horizon. However, according to the assessments made taking into account the time interval of the data, it is reasonable in our opinion to consider five steps and beyond as a long-term forecast. However, we believe that it would be a more correct approach to classify forecast models that can make reliable forecasts not only for the specified time horizon but also beyond, as long-term forecast models, without taking into account the prediction horizon of the model. Therefore, in this study, we test our model with several time horizons and compare their performance.

3 Related Technologies

In this section, the related technologies used in the proposed model are described.

3.1 Recurrent Neural Network (RNN)

This model is one of the most important DL techniques particularly developed for time series problems. RNN has a simple feedback loop in order to learn dependencies among the different time intervals. However, the basic RNN architecture is insufficient to capture complex relationships in long time intervals, so two different versions have been proposed. The first of these versions is long short-term memory (LSTM) and the other is gated recurrent network (GRU). LSTM has three different gates (input, output, forget) while GRU has two different gates (update and reset) and by the agency of these gates, they remove unnecessary information from the model that comes from the past states, and allowing the model to focus on only useful information. In this way, the model can learn long-term dependencies with ease. Figure 2 presents the general structure of the RNN, LSTM and GRU.

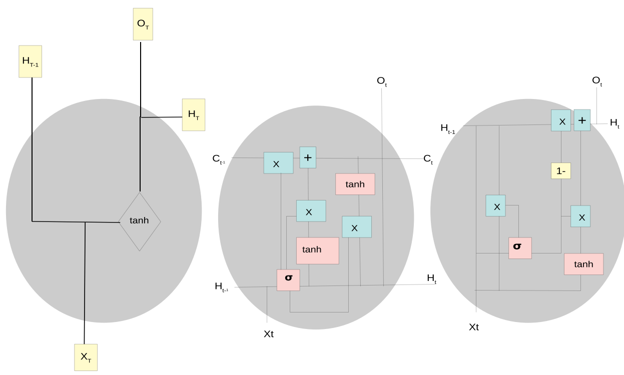

Fig. 2 Structure of a RNN (on the left), LSTM (in the middle) and GRU (on the right)

3.2 Convolutional Neural Network (CNN)

Since CNN was not developed for time series problems, it was not used for time series prediction for a long time. However, with the increase in the amount of data, the increase in computational load and the inability to parallelize the RNN algorithm efficiently led to new searches. CNN is promising for time series problems as it can be parallelized and produces faster results. In recent years, successful CNN-based time series forecasting models have been developed. Due to its architecture, CNN is used to reveal the relationships of different time series, especially in problems that need learning temporal dependencies and spatial dependencies together. A simple CNN model includes convolution layers, pooling layers, fully connected layers (FC) and an output layer. Filters are used in the convolution and pooling layers and the results are combined in the FC layer. In this way, learning is provided at each convolution layer.

3.3 Deep Ensemble Learning

Ensemble learning models combine several base or individual models with different strategies in order to provide better generalization and improve final prediction performance [13, 14]. Moreover, today, deep learning models with complex and layered architecture outperform traditional prediction models. Deep ensemble learning models, on the other hand, aim to build a more successful prediction model by combining the peculiar advantages of these two models. There are many models developed for traffic flow prediction in the literature, but few of them are ensemble learning-based. However, ensemble learning-based models provide higher accuracy and generalizability because they are constructed by combining either individual models developed with different combinations of the same method or individual models developed using different methods. Combining multiple models in this way for traffic flow forecasting can increase the final forecasting accuracy while preventing overfitting. Because each individual model deals with one aspect of the final model, as a result, the final model provides a more general representation and achieves a higher predictive accuracy compared to individual models. To this end, we focus on ensemble learning approaches in this study and propose a novel deep ensemble model for traffic flow prediction. The formula for an ensemble model is as follows:

$$\mathrm{FPM}(t) = \sum_{k=1}^{K} W_k \alpha_k(t). \tag{3}$$

where FPM is final prediction model, α_k is the kth individual model, W_k is the weight of the kth individual model, and K is the number of individual models.

According to this formula, the ensemble learning model gives weight to each individual model. The most common approach in the literature, for this purpose, is to give equal weight to each model. One issue of this approach is that each model conduces equally to the final prediction, without considering the prediction performance of single models. When we give a fixed weight to each single model, we limit the performance of the ensemble model due to a reduction in its generalization ability. Therefore to improve the prediction accuracy, we propose a flexible and robust deep ensemble model in this study. The proposed model assigns the weights based on the individual model performance and traffic situation change.

4 Related Work

The importance of traffic flow prediction in transportation engineering is increasing, and accordingly, we can say that there is a very large literature in this field. Most studies propose a model to predict traffic flow. We will examine these proposed models under two topics by following the tradition in the literature: parametric models and nonparametric models [3, 12, 15–19]. We summarize the related literature in Table 1.

4.1 Parametric Models

Models in this class can be explained by traffic flow theories of transportation engineering, statistics and probability. In a parametric model, traffic flow is represented as a function of random variables (e.g., accident, instantaneous decisions of drivers), time-dependent variables (e.g., time of day, day of the week or season) and auxiliary variables (e.g., weather, public holidays, sports or concert events). That is, traffic flow is defined as the total number of vehicles passing through a certain road segment at a certain time period under the influence of many dependent or independent variables, each

Table 1 Summary of the related literature

References	Horizon	Input data	Data size	Method/technologies used	Evaluation metrics
[9]	24 h	Highways agency network traffic flow data	15 min resolution from 1 Jul 2018 to 28 Jan 2020	Wavelet decomposition, convolutional neural network-long and short- term memory neural network	Root mean square error (RMSE) mean absolute error (MAE), R square
[7]	Up to 24 h	Caltrans performance measurement system (PeMS) dataset	5 min interval. Data size not mentioned	LSTM encoder–decoder	RMSE and symmetric mean absolute percentage error (SMAPE)
[1]	24 h	Dataset obtained from the DRIVENET	Between February 1, 2015 to March 31, 2016	Deep neural network (DNN)	Absolute percentage error (APE), mean absolute percentage error (MAPE)
[8]	Up to 4 h	GPS-data taken from the GAIA	The data contains trips from October to November 2016	Graph CNN-LSTM neural network	RMSE, MSE, MAE and MAPE
[12]	Up to 1 h	Urban corridor of 30 road segments with 24 intersections along Victoria Street (Melbourne)	One-year data of the year 2016	Convolutional GRU with attention network	Weighted mean absolute percentage error (WMAPE), RMSE, MAE
[16]	Single step (unspecified)	PeMS	The 5-min traffic flow data of District 5 named Central Coast in 2013	Ensemble learning, CNN	MAE, RMSE, mean relative error (MRE) and the standard deviations MAE, MRE and RMSE
[17]	Up to three-step	Data from the Portland-Vancouver Metropolitan region	During a 4-month period from March 4 to June 28, 2019	Ensemble learning, ensemble empirical mode decomposition, DBN (deep belief networks)	RMSE, MAPE
[22]	1-h	Princes highway, Victoria Road, Canterbury Road, and M1 in Sydney	Hourly traffic count from November 2017 to November 2018	Ensemble learning, ARIMA	RMSE, MAPE
[30]	5 h	Data from Hangzhou Integrated Transportation Research Center and PeMSD10	From 16th October, 2013 to 3rd October, 2014 and from 1st January, 2018 to 31st March, 2018 (15 min resolution)	Graph convolutional network, recurrent neural network	RMSE and MAPE
[33]	1 h (12 steps)	PeMSD4 and PeMSD8	From January to February in 2018 and from July to August in 2016 (5-min interval)	Encoder–decoder, attention network, LSTM	RMSE, MAE, MAPE, median absolute error (MdAE), mean absolute scaled error (MASE)

Table 1 continued

References	Horizon	Input data	Data size	Method/technologies used	Evaluation metrics
[38]	1 h (4 steps)	Arterial sensors in Arcadia, CA in 2015	15-min interval data	Ensemble learning, ARMAX, partial least squares, support vector regression, kernel ridge regression, Gaussian process regression	MAE and StdAE (standard deviation)
[39]	Up to 30 min	PeMS ("freeway segment located in San Diego")	From September 1, 2019 to September 30, 2019, and the sampling time interval is 5 min	Ensemble empirical mode decomposition, wavelet, LSTM	RMSE, MAE, MAPE
[47]	Up to 30 min	Data of Yuanda Road, Furong District, Changsha City	From September to October in 2013, excluding weekend data for a total of 40 days, time interval is 5 min	Optimized variational mode decomposition (OVMD) and combined long short-term memory network (LSTM)	RMSE, MAE

of which is dynamic in itself. Modeling with parametric approaches is relatively easy, but these models are suitable for uncomplicated small-sized datasets [20].

The most widely used parametric approaches in the literature are ARIMA, Kalman filtering and linear regression.

ARIMA is a time series modeling approach that explores the temporal relationship between data points of a time series. There are many traffic flow forecasting models developed using ARIMA and its advanced versions i.e., ARIMAX, SARIMA, SARIMAX in the literature [21, 22].

Kalman filtering is a widely used traffic flow prediction method. Its main idea is to predict future traffic flow using historical traffic flow data with a recursive or iterative process [23].

Linear regression is a pretty simple parametric approach. This method describes the traffic flow as a linear combination of the independent variables [24].

4.2 Nonparametric Models

Models in this category are more advanced than parametric models, and their performance varies according to the quality and size of the dataset. These models can achieve satisfactory prediction success with big data, but this requires quite a lot of computational capacity. K-nearest neighbor (KNN), support vector machine (SVM) and neural networks (NN) are the approaches we can count in this category.

KNN can be used for classification or regression. In this model, common patterns are tried to be extracted from historical traffic flow data. By using the best match with these defined patterns, future traffic patterns are tried to be predicted [25].

Another parametric model used in traffic flow prediction is SVM [15]. Although the estimation accuracy can be increased by using different "kernels," the computational load of model training is quite high, especially compared to KNN and NN. Therefore, it is not practical for large datasets. Indeed, KNN and SVM are not popular models developed for traffic flow forecast. The most popular models in this category are the NN-based models. And the reasons why NN-based models are so popular can be listed as follows: (1) they are suitable for big data, (2) they have fast convergence, and (3) they can achieve high prediction accuracy. A wide variety of NN models have been proposed for traffic flow prediction [1, 7, 12, 18].

4.2.1 Deep Learning-Based Models

Although deep learning models are also nonparametric models, we wanted to examine these studies separately since they have been very popular in this field in recent years and have a fairly wide literature. The simplest DL models that can be found in the literature in this field are multilayer-perceptron (MLP)-based models developed using multiple hidden layers [26]. However, the most widely used DL technique in solving the traffic flow prediction problem is recurrent neural networks (RNN). Especially, GRU and LSTM techniques, which are variants of RNN, are the most common methods since they are successful in capturing dependencies at different times. For example, [27] developed a two-layer

LSTM-based model. It used a fully connected layer as the extraction layer in the first layer, and the LSTM layer as the prediction layer in the second layer. The proposed other LSTM-based models are in [6]. A GRU-based model is proposed in [28]. In this study, weather data was used in addition to traffic data. Apart from RNN, CNN-based models also have been proposed for short-term traffic flow forecasting problems [29–31]. CNN-based models are especially preferred because they can produce results faster than other neural networks [32].

4.2.2 Hybrid and Ensemble Models

Understanding that it is not possible to model the complexity of traffic data with simple and traditional methods, many researchers have turned to hybrid models, especially in recent years. While in early studies we can see the combination of several parametric models, in recent studies, many of the hybrid models were built by combining two or more nonparametric methods [7, 8, 11, 33, 34]. Especially LSTM and CNN are used together in recently developed hybrid models [30, 35, 36]. There are also hybrid models developed by using parametric and nonparametric methods together [11, 37].

On the other hand, EL-based models emerge as a new trend [38–43]. There are only a limited number of EL-based prediction models in the literature [16, 17, 22, 44–46]. However, none of these studies focus on long-term forecasting. And this is a research gap that we want to fill in this study.

5 Methodology

5.1 The Proposed Model

The proposed model is a deep ensemble model which is capable of properly fusing the prediction results of multiple deep learning models. Our model learns the strengths and weaknesses of individual models and weights the predictions of single models according to their prediction performance. In addition, our model is flexible and performs well under different traffic conditions since our model receives actual data as well as prediction results from each model to obtain the final result.

Figure 3 demonstrates the details of our deep ensemble traffic flow prediction framework. Our proposed model consists of three stages. The first stage is the preprocessing and dataset preparation. We will explain this stage in detail in the next section. The second stage is base model selection. At this stage, we adjust the configurations of the three base models, namely LSTM, GRU and CNN. For this, we run models LSTM, GRU and CNN multiple times with different time lags, numbers of hidden layers and neurons. We optimize the internal parameters of each base model and select the best models with the highest accuracy. After selecting the base models, in the third stage, we decide how much each base model will contribute to the final model according to their performance. That is we develop a meta-learner to dynamically weight each base learner. For this, we first form new training, validation and test sets using each base model, then by using these new datasets we build a feed forward neural network-based (FFNN) model with deep architecture and the outputs of this final model (i.e., FFNN) or meta-learner are the weights of each base model. Thus, the weight of each base model is determined automatically and dynamically. Meanwhile, in order to capture the traffic condition changes, we use raw input data as well during the construction of the final model. Consequently, we separate the base models weighting step from the base models selection and tuning step so that the ensemble model can be dynamic and can change with the traffic conditions.

As we mentioned in Sect. 3.3, an ensemble model can be built in two different approaches: It can be built by combining either individual models developed with different combinations of the same method or individual models developed using different methods. The novelty of our model is that it combines these two approaches. Moreover, the base learners and meta-learner we use in our model all have deep architecture, and we do not use a fixed weight for each base learner, we introduce a meta-learner with the ability of dynamically weighting the base learners according to their predictive success.

6 Experiments

6.1 Dataset and Preprocessing

We conducted this study with a publicly available and a real-world dataset.[1] The dataset contains a total of 274 stations. The data was collected from January 1st, 2015 to December 31st, 2015, which contains both weekends and weekdays and aggregated 1 h intervals. Although the dataset contains 274 stations, some stations only have data for 3–4 months, for instance, station 116820 has data only for the 2nd, 9th, 11th and 12th months. That is, for some stations there are too many missing values, and this disrupts the continuity of the dataset. However, this is not desirable for time series and can significantly reduce the forecasting quality. Therefore, both because our computational resources are limited and because we want our model to produce more reliable predictive results, we have selected 100 stations with as few

[1] Source: www.transportation.gov/data, and it is available at: https://cloud.google.com/bigquery/public-data.

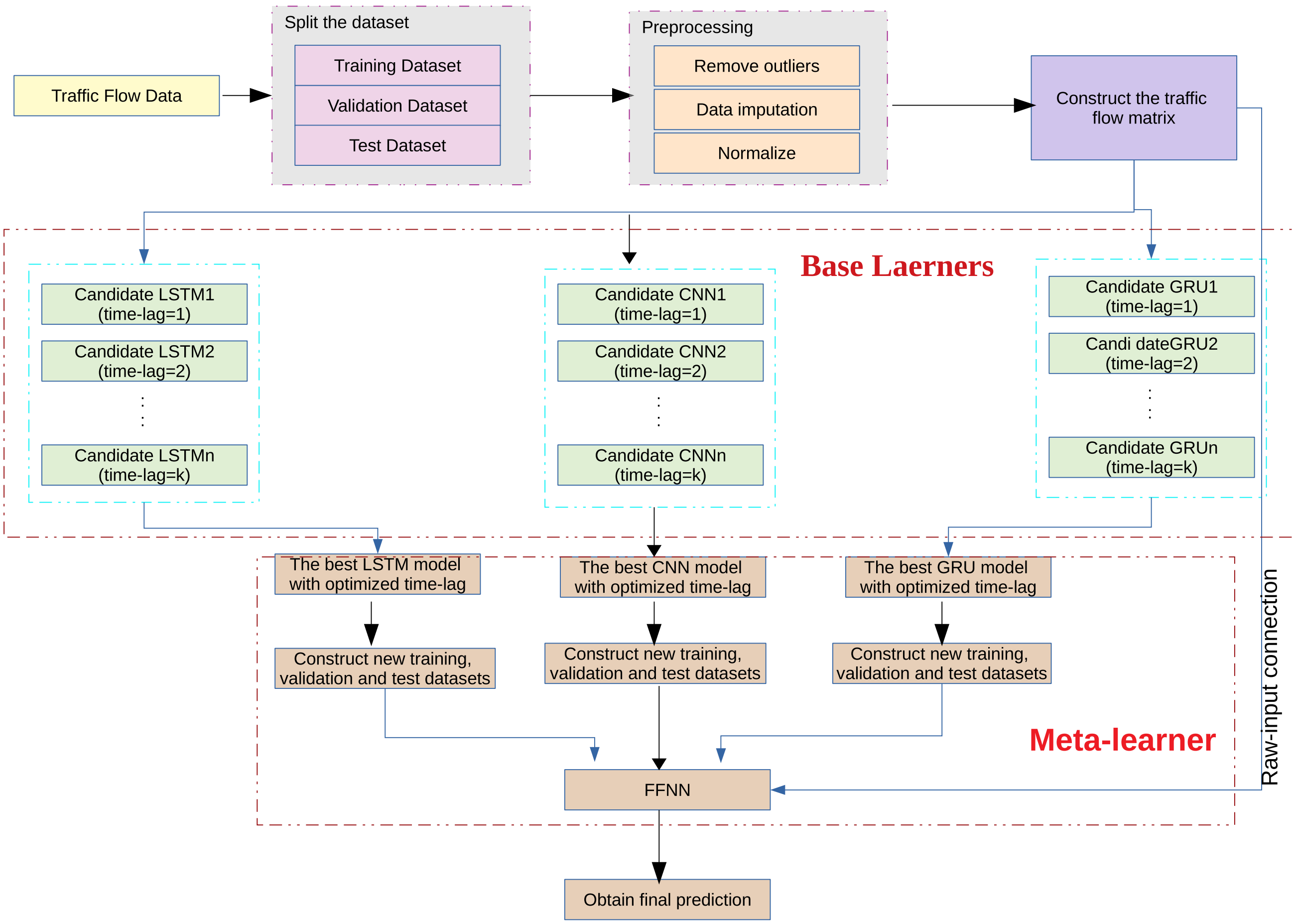

Fig. 3 General structure of the proposed model

missing values as possible and we tested all prediction models by using these 100 stations. Figure 4 shows the first 45 stations we have selected and Fig. 5 shows the locations of the selected stations.

We filled the missing values of the stations used in the experiments by averaging the data of the previous and the next hour. Thus, the total number of data samples is 100 $*$ 365. We chose this method to fill the missing data because data that is closer together, whether spatially or temporally, is more related to each other than data that are far apart. This idea is based on Tobler's first law. Tobler's first law says that things that are close together are more related to each other [22]. Inspired by this, we used this method to fill the missing data.

While choosing the stations we will use in our experiments, we also took into account the road type to which the station belongs, in addition to the amount of data because we wanted to show how robust and generalizable our model is for different road types. The road types we use are shown in Table 2.

We separated the dataset into three: We organized 65% of the dataset (about the first eight months) as the training set, the last two months as the validation set, and the remaining part as the test set. And we performed Z-Score normalization.

6.2 Constructing Traffic Flow Matrix

We tried to find the optimum time lag by running each deep learning model (base learner) many times with different time lags, i.e., the current traffic flow depends on how many steps in the past traffic flow. Thus, we have obtained an optimum time lag for each base learner. If we show the time-lag value with W; we set W hours as the time lag and added W new features, each of which indicates hourly traffic volumes in a W-h period. In this way, prediction models try to predict the traffic volume in the $(W + 1)$th hour by using previous W hours of data. We tested the proposed model for multiple horizon values: The prediction horizon h is specified as 1 for single-step prediction, and 2, 3, 4, 5, 9, 12, 24 for multi-step prediction (i.e., long-term prediction). That is, we used W hours historical data to predict the following h hour(s) traffic flow value. Accordingly, we constructed the traffic

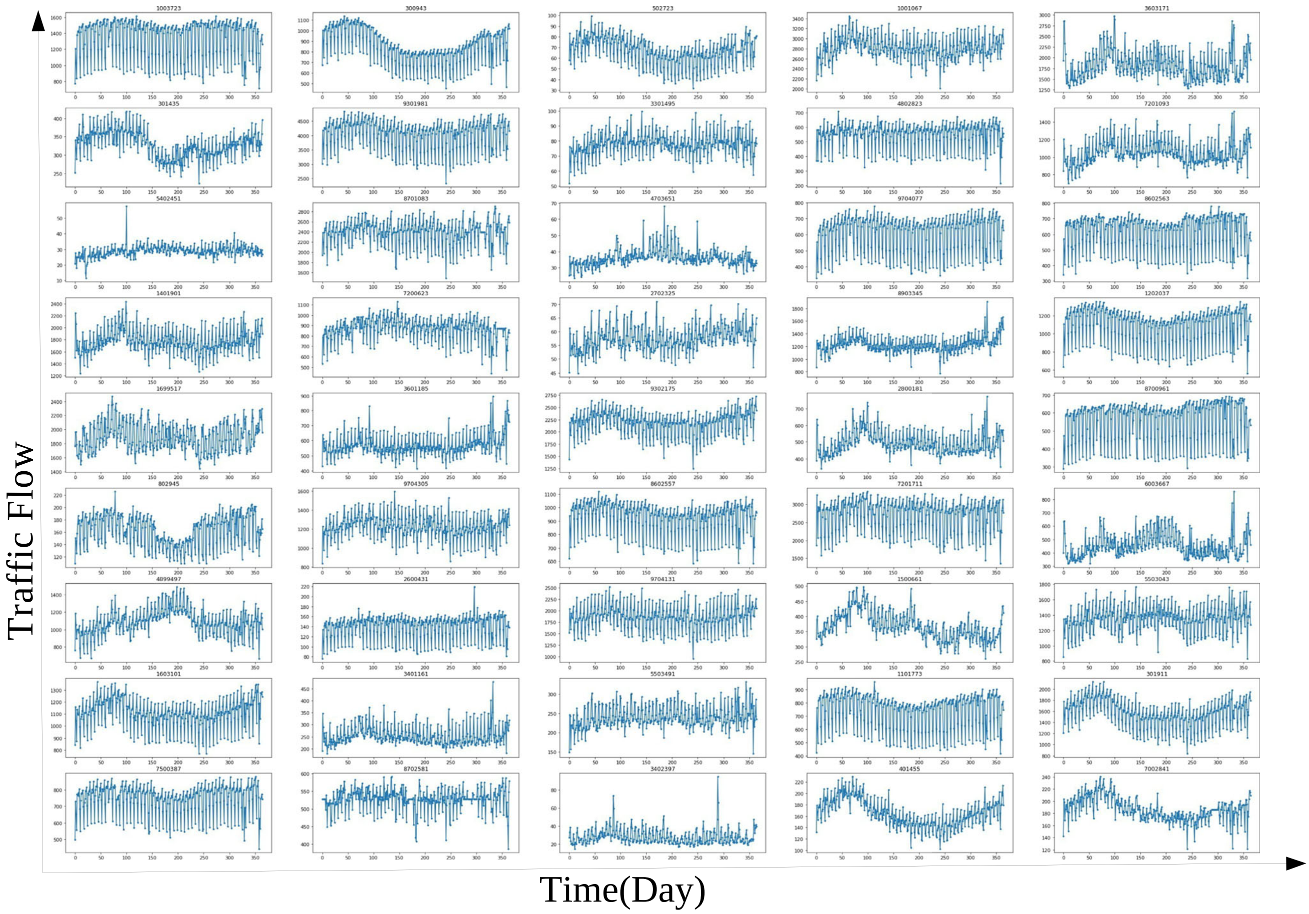

Fig. 4 Dataset (the first 45 stations)

flow matrix as input (X) and output (Y) matrix as follows:

$$X_h^{s1} = \begin{bmatrix} f_t^{s1} & f_{(t+1)}^{s1} & \cdots & f_{t(W-1)}^{s1} \\ f_{(t+d)}^{s1} & f_{(t+d+1)}^{s1} & \cdots & f_{t(W-1)+d}^{s1} \\ \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \quad (4)$$

$$Y_h^{s1} = \begin{bmatrix} f_{t(W-1)+1}^{s1} & f_{t(W-1)+2}^{s1} & \cdots & f_{t(W-1)+h}^{s1} \\ f_{t(W-1)+2}^{s1} & f_{t(W-1)+3}^{s1} & \cdots & f_{t(W-1)+h+1}^{s1} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (5)$$

In Eqs. 4 and 5, f_t^{s1} indicates the traffic flow of station 1 at time t. h represents prediction horizon, W denotes time lag (or time-window size), and d is the stride value.

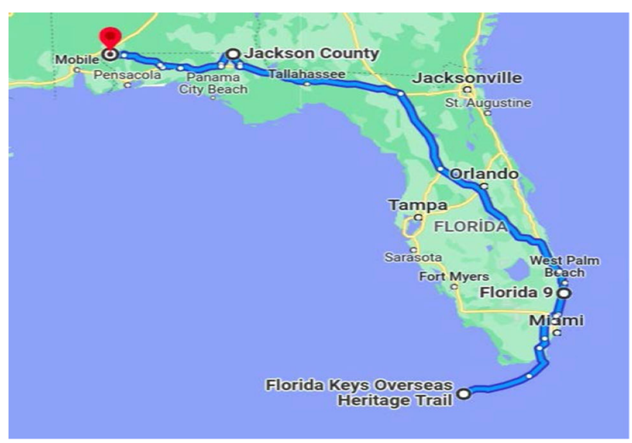

Fig. 5 Road network used for experiments

6.3 Experiments Settings

TensorFlow[2] and Keras,[3] which are open-source libraries of Python, were used to build the proposed deep ensemble model and other deep learning models. We used the scikit-learn[4] as the machine learning library to implement the linear regression (LR), KNN, DT, RF models.

[2] www.tensorflow.org.

[3] www.keras.io.

[4] www.scikit-learn.org.

Table 2 Road types

Urban: principal arterial—other
Urban: principal arterial—other freeways or expressways
Urban: principal arterial—interstate
Urban: minor arterial
Rural: principal arterial—other
Rural: minor arterial
Rural: principal arterial—interstate
Rural: major collector

We made a lot of trials to determine the best time lag. As a result of these trials, we found that the best time lag is 24 h for all models.

We optimized the hyper-parameters of each model separately. For deep learning models, the number of hidden neurons, activation function, dropout rate and learning rate were optimized by using 'Bayesian Search' algorithm. Table 3 shows the hyper-parameter values that we obtained as a result of optimization for each deep learning model. We used 'Random Search' algorithm for optimizing hyper-parameters of LR, KNN, DT, RF models. The Adam algorithm is used to optimize the loss function of all deep learning models and the ensemble model. The maximum number of epochs is set to 100; however due to early stopping, there was no model that reached 100 epochs.

6.4 Comparison Metrics

We use four metrics to measure the performance of the developed models, mean absolute error (MAE), mean squared error (MSE), mean squared logarithmic error (MSLE) and R-squared score which are the most frequently used metrics for traffic forecasting.

MAE, MSE, MSLE, R^2 are defined as:

$$\mathrm{MSE} = \frac{1}{T}\sum_{n=1}^{T}(t_n - p_n)^2. \tag{6}$$

$$\mathrm{MAE} = \frac{1}{T}\sum_{n=1}^{T}|t_n - p_n|. \tag{7}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{T}(t_n - p_n)^2}{\sum_{n=1}^{T}(t_n - v)^2}. \tag{8}$$

$$\mathrm{MSLE} = \frac{1}{T}\sum_{n=1}^{T}(\log(1 + t_n) - \log(1 + p_n))^2. \tag{9}$$

where t, p and T indicate the actual value, prediction value and the total number of samples, respectively. And v indicates that mean value of the actual traffic flow data.

Table 3 Optimum hyper-parameter settings

Model	Parameter	Value
LSTM hyper-parameters (base learner)	Number of layers	4
	Number of units	512,512,32,32
	Activations	relu, relu, relu,tanh
	Dropout rate	0.0
	Learning rate	0.0001
GRU hyper-parameters (base learner)	Number of layers	4
	Number of units	512, 512, 32, 96
	Activations	tanh,relu,relu,relu
	Dropout rate	0.5
	Learning rate	0.0001
CNN hyper-parameters (base learner)	Number of hidden layers	3
	Number of units	512, 96, 128
	Filter size	64
	Activations	tanh, relu, tanh, relu
	Dropout rate	0.0
	Learning rate	0.0001
Meta-Learner hyper-parameters	Number of layers	4
	Number of units	352, 512, 96, 416
	Activations	tanh, tanh, tanh, tanh
	Dropout rate	0.2
	Learning rate	0.0001

Table 4 Comparison of prediction performances of the proposed ensemble model and other competitive models for eight time horizons

Prediction horizon (h)	Metrics	LR	DT	KNN	RF	LSTM	GRU	CNN	Ensl	Ens2
1	MSE	0.1007	0.1162	0.0882	0.0840	0.0676	0.0687	0.0676	0.0641	**0.0613**
	MAE	0.2079	0.2065	0.1728	0.1740	0.1656	0.1657	0.1675	0.1590	**0.1553**
2	MSE	0.1501	0.1702	0.1218	0.1245	0.1139	0.1095	0.0989	0.0899	**0.0862**
	MAE	0.2510	0.2484	0.2057	0.2075	0.2252	0.2166	0.2039	0.1889	**0.1840**
3	MSE	0.1918	0.2196	0.1460	0.1640	0.1499	0.1195	0.1271	0.1141	**0.1119**
	MAE	0.2801	0.2794	0.2226	0.2333	0.2401	0.2168	0.2296	0.2102	**0.2065**
4	MSE	0.2161	0.2911	0.1705	0.1877	0.2330	0.1506	0.1406	0.1464	**0.1313**
	MAE	0.2994	0.3188	0.2488	0.2566	0.3412	0.2560	0.2449	0.2408	**0.2302**
5	MSE	0.2395	0.2940	0.1780	0.2174	0.1517	0.1514	0.1881	0.1408	**0.1388**
	MAE	0.3136	0.3117	0.2463	0.2611	0.2455	0.2438	0.2911	0.2329	**0.2300**
9	MSE	0.2817	0.4037	0.2164	0.3058	0.2090	0.2213	0.2066	**0.1893**	**0.1904**
	MAE	0.3391	0.3622	0.2767	0.3097	0.3017	0.3071	0.2923	0.2767	**0.2752**
12	MSE	0.2938	0.5464	0.2418	0.4308	0.2160	0.3016	0.2117	0.2165	**0.2079**
	MAE	0.3472	0.4100	0.2999	0.3530	0.3021	0.3725	0.2938	0.2911	**0.2847**
24	MSE	0.3256	0.4957	0.3019	0.3874	0.2598	0.2682	0.2754	0.2548	**0.2539**
	MAE	0.3648	0.4569	0.3452	0.3914	0.3224	0.3396	0.3467	0.3186	**0.3180**

Bold text is used to highlight the result of the most successful model for each prediction horizon

Table 5 Comparison of prediction performances of the proposed ensemble models and base models for eight time horizons

Prediction horizon (h)	Metrics	LSTM	GRU	CNN	Ensl	Ens2
1	MSLE	0.0140	0.0142	0.0139	0.0134	0.0125
	R^2	0.9248	0.9213	0.9258	0.9301	0.9366
2	MSLE	0.0229	0.0222	0.0196	0.0183	0.0172
	R^2	0.8722	0.8715	0.8860	0.9028	0.9050
3	MSLE	0.0250	0.0221	0.0243	0.0213	0.0208
	R^2	0.8245	0.8605	0.8583	0.8730	0.8737
4	MSLE	0.0400	0.0280	0.0267	0.0266	0.0240
	R^2	0.6247	0.8321	0.8349	0.8422	0.8543
5	MSLE	0.0277	0.0273	0.0332	0.0253	0.0248
	R^2	0.8181	0.8281	0.7876	0.8311	0.8367
9	MSLE	0.0383	0.0384	0.0360	0.0332	0.0331
	R^2	0.7254	0.7469	0.7506	0.7653	0.7562
12	MSLE	0.0376	0.0560	0.0361	0.0369	0.0352
	R^2	0.7255	0.5598	0.7101	0.7274	0.7338
24	MSLE	0.0440	0.0456	0.0468	0.0435	0.0426
	R^2	0.6563	0.6297	0.5832	0.6637	0.6724

6.5 Results and Discussions

In order to assess the performance of our ensemble model (i.e., Ens2 in Tables 4, 5), we compare the proposed model with base models, i.e., LSTM, GRU and CNN. Besides these models, we also compare our model with some selected traditional machine learning models including LR, KNN, DT, RF. We use raw input connection in meta-learner so that our ensemble model can be dynamic and capture the traffic conditions well. We construct an alternative model, which has the same architecture as the proposed ensemble model except the raw input connection (i.e., Ens1 in Tables 4, 5). We also compared the prediction performance of this model with the model we propose.

The 3-day forecasting performance of the base models and the ensemble models are shown in Fig. 6a, b. Figure 6a shows single-step prediction (i.e., prediction horizon = 1) and Fig. 6b shows multi-step predictions (i.e., prediction horizon = 24). The forecasting graphs belonging to the same period (month/day/hour) were selected to compare the performance

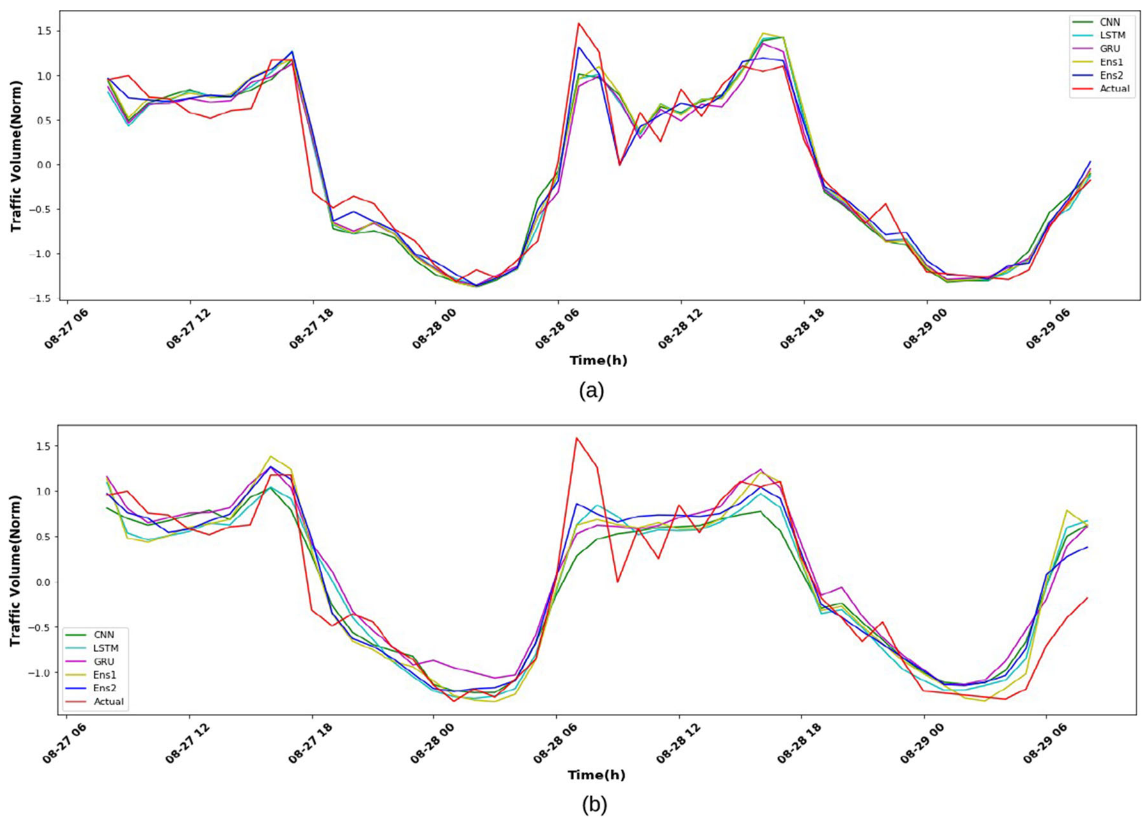

Fig. 6 Comparison of the prediction results: **a** TimeHorizon = 1 (single-step prediction). **b** TimeHorizon = 24 (multi-step prediction)

of the top and bottom forecasting horizons. Based on these figures, the predictive success of the proposed deep ensemble model has increased considerably for both time horizons compared to single models. In the same figure, the histograms in Fig. 7a–e show forecasting performances for time horizon = 1 for different days of the week and different times of the day. To obtain these histograms, for each model, first, the difference between the each forecasting point and its ground truth was taken separately. Then, for each forecasting point, the model with the smallest of this difference was awarded a score, and in the end, the models' scores were added up. These histograms show the total score of each model. The highest score in all five histograms belongs to the proposed ensemble model. In fact, these histograms show that the proposed model is decisively ahead of the other models.

Table 4 shows MSE and MAE results we have obtained as a result of our experiments for eight time horizons. As can be seen in the table, our model (i.e., Ens2) is the most successful model in all time horizons except time horizon = 9, in which Ens1 model was the most successful. These results prove that the ensemble models perform better, especially in long-term traffic flow prediction.

In addition to these main results, we can list the other results we achieved when we carefully examine the table as follows:

- Traditional machine learning models are not sufficient for long-term prediction.
- Among the traditional machine learning-based (ML-based) models we have compared, the best performance belongs to KNN. However, the performance of tree-based models is quite low. The prediction performance of random forest (RF), which is a tree-based ensemble model, is quite far behind KNN.
- Among the deep learning (DL) models, CNN has shown the best performance in many horizons. This result is interesting because CNN was not originally developed for time series problems. But before we can generalize this result, we need to do more experiments. For the dataset we used in this study, CNN performs quite successfully. In other words, we can say that we have developed a prediction model compatible with the dataset. Nevertheless, we cannot make a general conclusion that CNN is the most successful DL technique for time series problems. However, we can say that CNN is promising for such problems.
- According to the results of our experiments, although CNN is more successful DL model than others, the performance of the other DL models is roughly competitive with CNN.

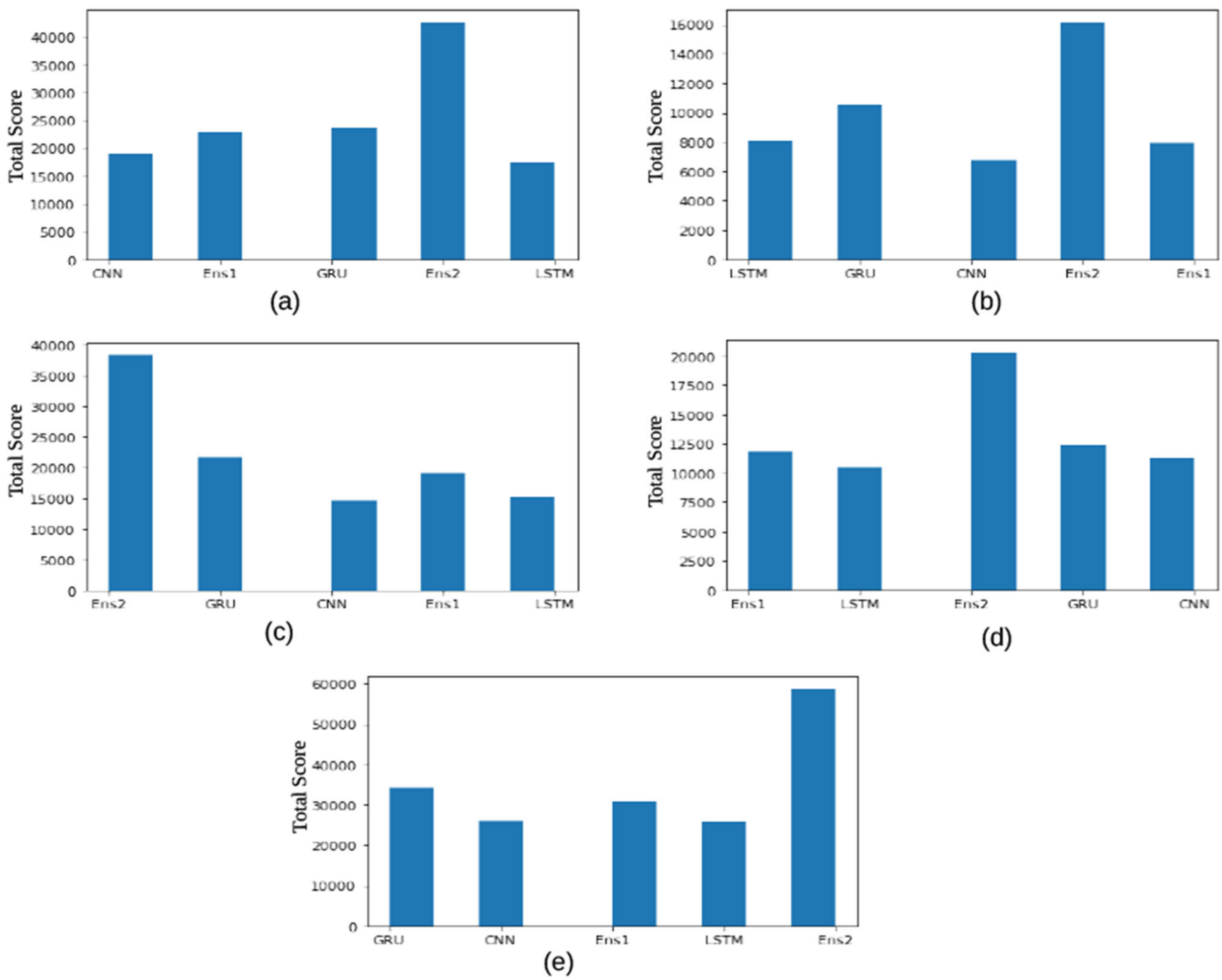

Fig. 7 Comparison of the prediction results: **a** prediction results on weekdays. **b** Prediction results on weekends. **c** Prediction results on rush hours. **d** Prediction results on off-peak hours. **e** Overall performance (TimeHorizon = 1)

- These results indicate that DL-based models offer the opportunity to develop more successful prediction models because they can better capture long-term dependencies.
- Based on our observations during our experiments, we can also make a comparison between DL-based models in terms of computation times. CNN also performs best in terms of computation time among DL-based models. This is probably due to the fact that CNN can be parallelized more efficiently than GRU and LSTM. However, LSTM was the model with the worst performance in terms of computation time.
- When Fig. 6a, b is compared, it is seen that the proposed model is relatively more successful in sharp ups and downs.
- When the histograms in Fig. 7 are examined, it is seen that the prediction performance of the proposed model is quite good both on weekdays and during peak hours. However, the performance of CNN among the single models is the lowest for these two categories.
- When we examine the histograms in Fig. 7, we see that the most successful single model is GRU. GRU outperforms even our alternative ensemble model (Ens1) for all categories.
- In Table 5, we compare ensemble models with base models using MSLE and R^2 metrics. We chose these two metrics because the first metric measures relative error rather than actual error. That is, it gives approximately equal weight to small and large differences between actual and predicted values. The second metric is used to compare the quality of models with each other, rather than to decide the overall quality of a model. This metric takes a value between 0 and 1, and the model is considered good as the value gets closer to 1. However, it is very

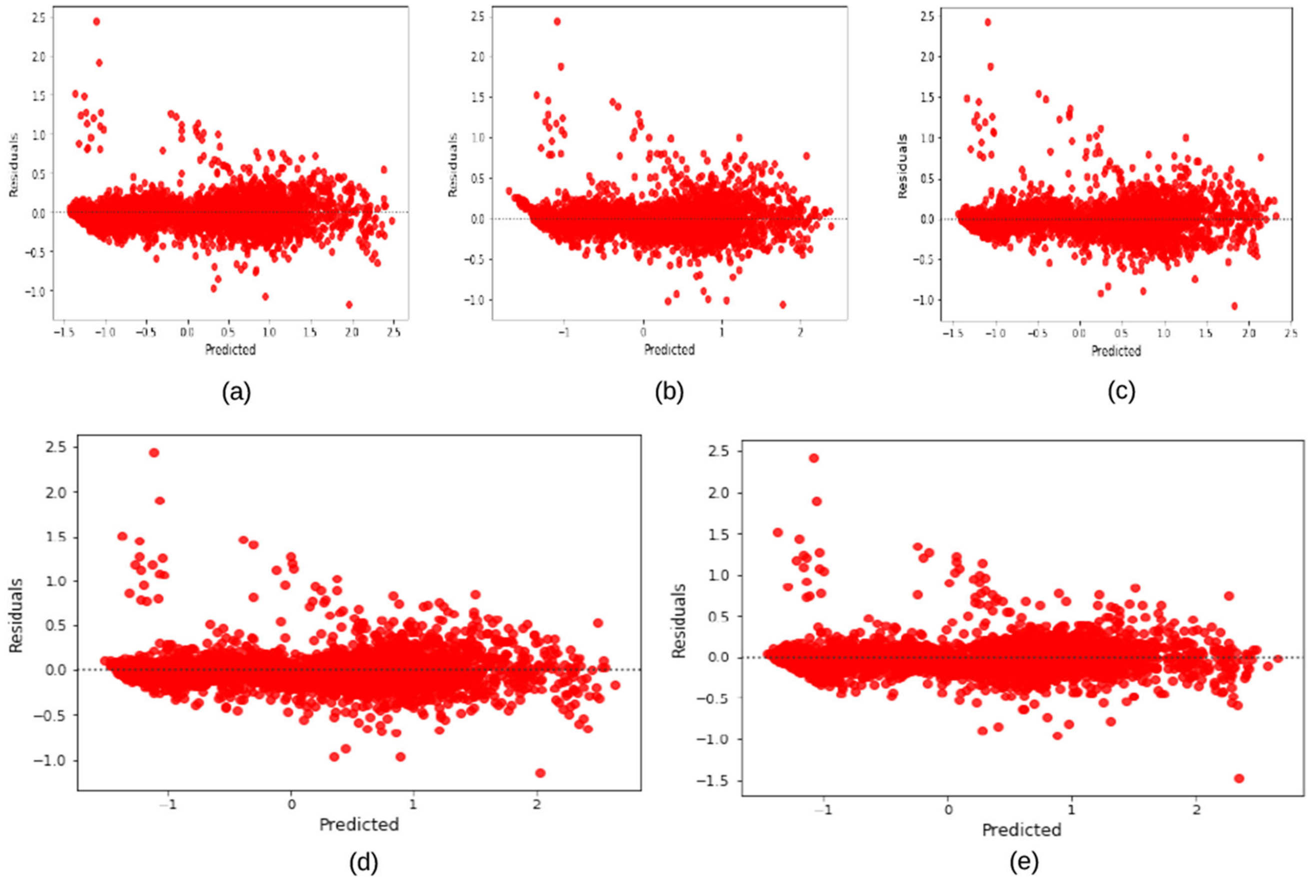

Fig. 8 Residual plots: **a** CNN, **b** GRU, **c** LSTM, **d** Ens1 and **e** Ens2

difficult to obtain values close to 1 for difficult problems such as traffic flow prediction. Therefore, each prediction problem should be evaluated on its own.

- When we examine both metrics values in Table 5, we see that the model we recommend is the best model for all prediction horizons except 9 h. This shows us the results in Table 5 are consistent with the results in Table 4.
- The results show that we can achieve a significant performance improvement when we combine ensemble learning architecture and deep learning techniques.
- In general, as prediction horizon increases, the prediction performance of all the models we compare, including the model we propose, decreases, which proves that the long-term prediction is more difficult.
- Although the prediction performance of our proposed model decreases as the prediction horizon increases, this decrease is small compared to the other models we compared. For example, when the forecast horizon is 4 h, the MSE of our model is 0.1313, and this value increases to 0.1388 when the forecast horizon is 5 h. In contrast, when the forecast horizon is 4 h, the MSE of CNN is 0.1406. However, when the forecast horizon increases to 5 h, this value increases to 0.1881.
- Figure 8 shows the residual plots of our proposed model and the models we compared. The residual plot shows the difference between actual values and predicted values. The performance of the model is directly proportional to the closeness of the points forming the graph to the starting point. When these graphs are examined in detail, the superiority of our proposed model over other models is clearly seen.

7 Conclusion and Future Work

Long-term traffic flow forecasting is vital for traffic management issues such as congestion control and better route selection. This importance will become more evident in the future with the development of related technologies. Therefore, it is critical to try to improve long-term traffic flow forecasting performance. That is why, this study proposed a novel ensemble model for long-term traffic flow prediction. The proposed model is a deep ensemble model built by properly combining three different deep learning techniques as base models. We designed our model that can dynamically produce the weights of the base models based on both each base model's performance and traffic condition. Experimental results show that the proposed approach outperforms all the models compared. In future research, we plan to investigate the effectiveness of our model with using different base models and datasets. We will also implement a 1D-CNN followed by a recurrent neural network (such as LSTM or GRU) as base learner and investigate the effect of including this network into our ensemble model. In addition, the fact that the CNN-based prediction model we developed was quite successful compared to other DL models motivated us to conduct more research in this area. As a future work, we plan to make more experiments to compare the forecasting performance of CNN using different time series datasets. More than that, we will try to understand why CNN is performing better. We also plan to address the issue of interpretability of DL-based models. Although deep learning algorithms provide high prediction performance, the interpretability of DL-based models is very low. This is also true for our model. Therefore, as a future study, we plan to analyze the outputs of the base learners of our model separately. Thus, we will try to discover the critical hours that affect the outcome for each model. It would also be beneficial to try to understand the temporal and spatial components to which our ensemble model gives more weight.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

References

1. Qu, L.; Li, W.; Li, W.; Ma, D.; Wang, Y.: Daily long-term traffic flow forecasting based on a deep neural network. Expert Syst. Appl. **121**, 304–312 (2019)
2. Sattar, K.; Chikh Oughali, F.; Assi, K.; Ratrout, N.; Jamal, A.; Masiur Rahman, S.: Transparent deep machine learning framework for predicting traffic crash severity. Neural Comput. Appl. **35**(2), 1535–1547 (2023)
3. Lana, I.; Del Ser, J.; Velez, M.; Vlahogianni, E.I.: Road traffic forecasting: recent advances and new challenges. IEEE Intell. Transp. Syst. Mag. **10**(2), 93–109 (2018)
4. Do, L.N.; Taherifar, N.; Vu, H.L.: Survey of neural network-based models for short-term traffic state prediction. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **9**(1), 1285 (2019)
5. Abbar, S.; Stanojevic, R.; Mustafa, S.; Mokbel, M.: Traffic routing in the ever-changing city of Doha. Commun. ACM **64**(4), 67–68 (2021)
6. Doğan, E.: LSTM training set analysis and clustering model development for short-term traffic flow prediction. Neural Comput. Appl. **33**(17), 11175–11188 (2021)
7. Wang, Z.; Su, X.; Ding, Z.: Long-term traffic prediction based on lstm encoder–decoder architecture. IEEE Trans. Intell. Transp. Syst. **22**(10), 6561–6571 (2020)
8. Bogaerts, T.; Masegosa, A.D.; Angarita-Zapata, J.S.; Onieva, E.; Hellinckx, P.: A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transp. Res. Part C Emerg. Technol. **112**, 62–77 (2020)
9. Li, Y.; Chai, S.; Ma, Z.; Wang, G.: A hybrid deep learning framework for long-term traffic flow prediction. IEEE Access **9**, 11264–11271 (2021)
10. Belhadi, A.; Djenouri, Y.; Djenouri, D.; Lin, J.C.-W.: A recurrent neural network for urban long-term traffic flow forecasting. Appl. Intell. **50**, 3252–3265 (2020)
11. Chen, H.; Grant-Muller, S.; Mussone, L.; Montgomery, F.: A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. Neural Comput. Appl. **10**, 277–286 (2001)
12. Do, L.N.; Vu, H.L.; Vo, B.Q.; Liu, Z.; Phung, D.: An effective spatial-temporal attention based neural network for traffic flow prediction. Transp. Res. Part C Emerg. Technol. **108**, 12–28 (2019)
13. Sagi, O.; Rokach, L.: Ensemble learning: a survey. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **8**(4), 1249 (2018)
14. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q.: A survey on ensemble learning. Front. Comput. Sci. **14**, 241–258 (2020)
15. Hong, W.-C.; Dong, Y.; Zheng, F.; Lai, C.-Y.: Forecasting urban traffic flow by SVR with continuous ACO. Appl. Math. Model. **35**(3), 1282–1291 (2011)
16. Chen, Y.; Chen, H.; Ye, P.; Lv, Y.; Wang, F.-Y.: Acting as a decision maker: traffic-condition-aware ensemble learning for traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **23**(4), 3190–3200 (2020)
17. Liu, J.; Wu, N.; Qiao, Y.; Li, Z.: Short-term traffic flow forecasting using ensemble approach based on deep belief networks. IEEE Trans. Intell. Transp. Syst. **23**(1), 404–417 (2020)
18. Gu, Y.; Lu, W.; Xu, X.; Qin, L.; Shao, Z.; Zhang, H.: An improved Bayesian combination model for short-term traffic prediction with deep learning. IEEE Trans. Intell. Transp. Syst. **21**(3), 1332–1342 (2019)
19. Aljuaydi, F.; Wiwatanapataphee, B.; Wu, Y.H.: Multivariate machine learning-based prediction models of freeway traffic flow under non-recurrent events. Alex. Eng. J. **65**, 151–162 (2023)
20. Guo, C.; Li, D.; Chen, X.: Unequal interval dynamic traffic flow prediction with singular point detection. Appl. Sci. **13**(15), 8973 (2023)
21. Williams, B.M.: Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp. Res. Rec. **1776**(1), 194–200 (2001)
22. Shahriari, S.; Ghasri, M.; Sisson, S.; Rashidi, T.: Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction. Transportmetrica A Transp. Sci. **16**(3), 1552–1573 (2020)
23. Okutani, I.; Stephanedes, Y.J.: Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. Part B Methodol. **18**(1), 1–11 (1984)
24. Sun, H.; Liu, H.X.; Xiao, H.; He, R.R.; Ran, B.: Use of local linear regression model for short-term traffic forecasting. Transp. Res. Rec. **1836**(1), 143–150 (2003)
25. Cai, P.; Wang, Y.; Lu, G.; Chen, P.; Ding, C.; Sun, J.: A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transp. Res. Part C Emerg. Technol. **62**, 21–34 (2016)

26. Polson, N.G.; Sokolov, V.O.: Deep learning for short-term traffic flow prediction. Transp. Res. Part C Emerg. Technol. **79**, 1–17 (2017)
27. Dai, X., Fu, R., Lin, Y., Li, L., Wang, F.-Y.: Deeptrend: a deep hierarchical neural network for traffic flow prediction. arXiv:1707.03213 (2017)
28. Zhang, D.; Kabuka, M.R.: Combining weather condition data to predict traffic flow: a GRU-based deep learning approach. IET Intell. Transp. Syst. **12**(7), 578–585 (2018)
29. Zhang, W.; Yu, Y.; Qi, Y.; Shu, F.; Wang, Y.: Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. Transportmetrica A Transp. Sci. **15**(2), 1688–1711 (2019)
30. Lv, M.; Hong, Z.; Chen, L.; Chen, T.; Zhu, T.; Ji, S.: Temporal multi-graph convolutional network for traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **22**(6), 3337–3348 (2020)
31. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Li, K.; Zeng, Z.: Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. ACM Trans. Knowl. Discov. Data (TKDD) **14**(4), 1–23 (2020)
32. Méndez, M.; Merayo, M.G.; Núñez, M.: Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model. Eng. Appl. Artif. Intell. **121**, 106041 (2023)
33. Shi, X.; Qi, H.; Shen, Y.; Wu, G.; Yin, B.: A spatial-temporal attention approach for traffic prediction. IEEE Trans. Intell. Transp. Syst. **22**(8), 4909–4918 (2020)
34. Chen, Z.; Lu, Z.; Chen, Q.; Zhong, H.; Zhang, Y.; Xue, J.; Wu, C.: Spatial-temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism. Inf. Sci. **611**, 522–539 (2022)
35. Wu, Y.; Tan, H.; Qin, L.; Ran, B.; Jiang, Z.: A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. Part C Emerg. Technol. **90**, 166–180 (2018)
36. Zheng, H.; Lin, F.; Feng, X.; Chen, Y.: A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **22**(11), 6910–6920 (2020)
37. Lu, S.; Zhang, Q.; Chen, G.; Seng, D.: A combined method for short-term traffic flow prediction based on recurrent neural network. Alex. Eng. J. **60**(1), 87–94 (2021)
38. Zhan, H.; Gomes, G.; Li, X.S.; Madduri, K.; Sim, A.; Wu, K.: Consensus ensemble system for traffic flow prediction. IEEE Trans. Intell. Transp. Syst. **19**(12), 3903–3914 (2018)
39. Chen, X.; Chen, H.; Yang, Y.; Wu, H.; Zhang, W.; Zhao, J.; Xiong, Y.: Traffic flow prediction by an ensemble framework with data denoising and deep learning model. Phys. A Stat. Mech. Appl. **565**, 125574 (2021)
40. Zheng, G.; Chai, W.K.; Katos, V.; Walton, M.: A joint temporal-spatial ensemble model for short-term traffic prediction. Neurocomputing **457**, 26–39 (2021)
41. Liu, Y.; Liu, Z.; Vu, H.L.; Lyu, C.: A spatio-temporal ensemble method for large-scale traffic state prediction. Comput.-Aided Civ. Infrastruct. Eng. **35**(1), 26–44 (2020)
42. Chen, L., Chen, C.P.: Ensemble learning approach for freeway short-term traffic flow prediction. In: 2007 IEEE International Conference on System of Systems Engineering, pp. 1–6. IEEE (2007)
43. Yan, H.; Fu, L.; Qi, Y.; Yu, D.-J.; Ye, Q.: Robust ensemble method for short-term traffic flow prediction. Future Gener. Comput. Syst. **133**, 395–410 (2022)
44. Zhang, Y.; Xin, D.: A diverse ensemble deep learning method for short-term traffic flow prediction based on spatiotemporal correlations. IEEE Trans. Intell. Transp. Syst. **23**(9), 16715–16727 (2021)
45. Zhao, F.; Zeng, G.-Q.; Lu, K.-D.: EnLSTM-WPEO: short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization. IEEE Trans. Veh. Technol. **69**(1), 101–113 (2019)
46. Xiao, J.; Xiao, Z.; Wang, D.; Bai, J.; Havyarimana, V.; Zeng, F.: Short-term traffic volume prediction by ensemble learning in concept drifting environments. Knowl.-Based Syst. **164**, 213–225 (2019)
47. Dai, G.; Tang, J.; Luo, W.: Short-term traffic flow prediction: an ensemble machine learning approach. Alex. Eng. J. **74**, 467–480 (2023)