

Traffic flow prediction models – A review of deep learning techniques

Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh K V & Soumya J Bhat |

To cite this article: Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh K V & Soumya J Bhat | (2022) Traffic flow prediction models – A review of deep learning techniques, Cogent Engineering, 9:1, 2010510, DOI: [10.1080/23311916.2021.2010510](https://doi.org/10.1080/23311916.2021.2010510)

To link to this article: <https://doi.org/10.1080/23311916.2021.2010510>



© 2021 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.



Published online: 27 Dec 2021.



Submit your article to this journal [↗](#)



Article views: 19639



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 16 View citing articles [↗](#)



Received 2 June 2021
Accepted 6 November 2021

*Corresponding author: Santhosh K V,
Manipal Academy of Higher
Education, Manipal Institute of
Technology, Manipal, India
E-mail: kv.santhu@gmail.com

Reviewing editor:
Fabio Galatioto, Modelling and
Visualisation, Ricardo Plc, United
Kingdom

Additional information is available at
the end of the article

COMPUTER SCIENCE | REVIEW ARTICLE

Traffic flow prediction models – A review of deep learning techniques

Anirudh Ameya Kashyap¹, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh K V* and Soumya J Bhat

Abstract: Traffic flow prediction is an essential part of the intelligent transport system. This is the accurate estimation of traffic flow in a given region at a particular interval of time in the future. The study of traffic forecasting is useful in mitigating congestion and make safer and cost-efficient travel. While traditional models use shallow networks, there has been an exponential growth in the number of vehicles in recent times and these traditional machine learning models fail to work in current scenarios. In our paper, we review some of the latest works in deep learning for traffic flow prediction. Many deep learning architectures include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Restricted Boltzmann Machines (RBM), and Stacked Auto Encoder (SAE). These deep learning models use multiple layers to extract higher level of features from raw input progressively. The latest deep learning models developed to tackle this very problem are reviewed and due to the complexity of transport networks, this review gives the reader information about how various factors influence these models and what models work best in different scenarios.

Subjects: Communication Technology; Automotive Technology & Engineering; Intelligent & Automated Transport System Technology

ABOUT THE AUTHOR

Mr. Anirudh Kashyap, Mr. Shravan, Ms. Ananya Devarakonda, Mr. Shamanth R Nayak K are graduate students and Ms. Soumya J Bhat doctoral student from Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India.
Dr. Santhosh K V is currently working as Associate Professor, in the Department of Instrumentation and Control Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India.

PUBLIC INTEREST STATEMENT

The Intelligent transport system is helpful in estimating the road network capacity, alleviating the traffic congestion and guiding the traffic participants, through traffic signal control, real-time traffic information collection and effective traffic data dissemination. Therefore, through our research paper we want to give readers and researchers a good idea on the latest deep learning models that exist to predict traffic flow. There are different architectures developed to solve this problem from Convolutional Neural Networks capturing spatial data, Hybrid CNN-LSTM which work very well because they capture both temporal and spatial aspects of traffic flow and other semi supervised networks like SAEs. It is very difficult choose from all these networks hence through our research paper we want to simplify these complex networks, their pros, cons and also in what environment they are being trained upon. This review gives the reader information about how various factors influence these models and what models work best in different scenarios.

Keywords: Deep learning; deep neural networks; hybrid models; intelligent transport system; traffic flow prediction; unsupervised learning

1. Introduction

Traffic flow prediction is an essential part of the Intelligent Transport System (ITS). This helps traffic stakeholders to make safer and smarter use of transport networks (Cen Chen et al., 2020; Sun, Boukerche et al., 2020). Traffic stakeholders include individual travelers, traffic managers, policy makers, and road users. Various traffic stakeholders are shown in Figure 1.

The effectiveness of these systems is determined by the quality of traffic data and only then an ITS will succeed. The global status on road safety of the World Health Organization (WHO) 2018, reported that road traffic deaths continue to increase, 1.35 million deaths recorded in the year 2016, making the study of traffic forecasting a useful way in mitigating congestion and make safer and cost-efficient travel. (Makaba et al., 2020; World Health Organization, 2018). The benefits of traffic forecasting are illustrated in Figure 2.

The big question is if traffic patterns, queuing patterns, and time be accurately predicted? Can predicting traffic flow enable decisions to resolve potential traffic congestion proactively? These questions have, therefore, been a critical area of research, especially in an urban context. In recent times the

Figure 1. Traffic stakeholders.

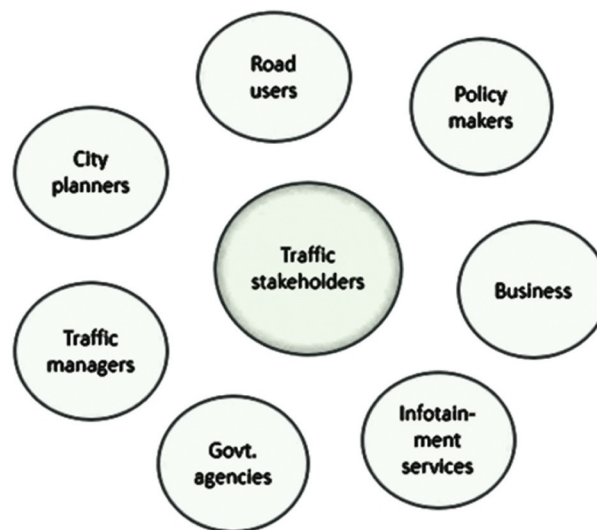
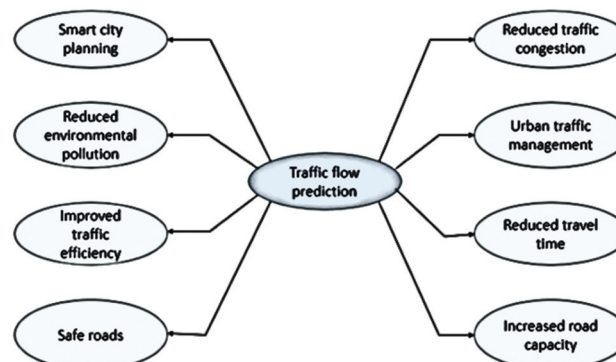


Figure 2. Benefits of traffic flow prediction.



urban fleet of vehicles has evolved to an extent where they can upload filtered sensor data directly to the cloud (Gerla et al., 2014). A vehicular cloud exists which provides services to autonomous vehicles. This further enables traffic flow prediction but the amalgamation of different features from different road segments and time-varying traffic patterns pose difficulties in accurately predicting traffic by different prediction models. The accurate prediction of traffic parameters in prediction models are difficult due to the dynamic and stochastic nature of traffic (Vijayalakshmi et al., 2021). Physical infrastructure and traffic rules impose huge constraints on prediction models and external factors like weather, accidents, and road closures play a huge role in influencing the model. Therefore, it is important to understand the various architectures which have been developed to tackle this problem.

In the last few years, we have entered the era of big data in transportation (Lv, Duan, Kang, Li, Wang et al., 2015). Traffic congestion forecasting is heavily dependent on sensors and other equipment to acquire relevant data, such as traffic speed, weather and accident data, etc. While traditional models use shallow networks, there has been an exponential growth in the number of vehicles in recent times and these traditional machine learning models fail to work in current scenarios. Therefore, this paper reviews deep learning models in-depth and also explores the realm of unsupervised learning with respect to traffic flow prediction.

Traditional models used parametric methods (Zheng et al., 2020). The Auto-Regressive Integrated Moving Average (ARIMA) model is a well-known framework and benchmark for short-term traffic flow prediction (Van Der Voort et al., 1996). Many changes to the ARIMA model were introduced and results indicated improved performances (Lee & Fambro, 1999; Williams, 2001; Williams & Hoel, 2003). However, due to the stochastic and non-linear nature of traffic flow, the parametric models failed to provide accurate results (Ken Chen et al., 2020). Therefore, non-parametric models were preferred, and thus Neural Networks became popular for traffic flow prediction. A shallow Back-Propagation Neural Network (BPNN) (Smith et al., 1994) showed promising results, but it failed to work in the big data era. Thus, came the emergence of deep learning, which uses multiple layers to extract a higher level of features from raw input progressively. Many deep learning architectures include Convolutional Neural Network (CNN) (Simonyan & Zisserman, 2015), Recurrent Neural Network (RNN) (Graves et al., 2013), Long Short-Term Memory (LSTM) (Sainath et al., 2015), Restricted Boltzmann Machines (RBM) (Good Fellow et al., 2013), Deep Belief Network (DBN) (Sarikaya et al., 2014), and Stacked Auto Encoder (SAE) (Gehring et al., 2013). This paper presents the review of these architectures. Due to the complexity of transport networks, it also gives the reader information about how various factors influence these models and what models work best in different scenarios. The work also presents the review of some of the unsupervised learning approaches and how they fare against supervised learning in deep learning.

2. Background study

Traffic congestion and resultant issues are the banes of our times, more so in developing countries with inadequate infrastructure. Traffic congestion impact in terms of lost productivity, energy costs, health risk cost through pollution, safety costs, and more are grossly underestimated and present a pressing opportunity to improve the standard of living and quality of life in general (Ahmed Yasser & Kader, 2017). The problem of traffic flow prediction is the accurate estimation of traffic flow in a given region at a particular interval of time in the future. The issue of traffic flow prediction has complex non-linear spatio-temporal dependencies and dependencies on external factors such as weekends, holidays, weather, events, road conditions, and more. Traditionally, traffic prediction models used statistical methods (Lee & Fambro, 1999; Van Der Voort et al., 1996; Williams, 2001; Williams & Hoel, 2003). However, due to the complex dependencies of traffic flow data, researchers have eventually moved from statistical methods to machine learning (Dong et al., 2018; L. Liu et al., 2019; Lu et al., 2020) and deep learning architectures (Sun, Wu, Xiang et al., 2020; Tampubolon & Hsiung, 2018; Yi et al., 2017) to solve the problem. Previous review papers addressed this shift from shallow Neural Network (NN)

architectures to Deep Neural Network (DNN) architectures for traffic flow predictions (Ali & Mahmood, 2018; Nguyen et al., 2018; Yin et al., 2020).

Deep learning models have shown promising results to represent the non-linearity of traffic flow prediction. For example, popular deep learning architectures such as CNNs have proved to adapt to spatial dependencies of traffic (Deng et al., 2019; Sun, Wu, Xiang et al., 2020). RNNs, especially LSTM architectures, adjust to long-term and short-term temporal dependencies of traffic flow (Osipov et al., 2020; B. Yang et al., 2019).

While there are several advantages to using the mentioned deep learning models to predict traffic flow individually, there are significant disadvantages. According to (Bae et al., 2018), as modern transportation systems rely on accurate data, any missing data profoundly impacts the accuracy of the deep learning model and can lead to suboptimal or ineffective performance. In the paper (Li et al., 2020), an attempt is made to solve the issue of missing data using tensor completion methods for data imputation, which are unsupervised methods.

Another limitation of deep learning architectures such as CNNs and RNNs is that they require large amounts of historical data for training. Here, achieving the requirement is considerably easy as we live in the era of big data. However, according to (Feng et al., 2020), large amounts of historical data can cause over-fitting of the model due to high fluctuations in traffic flow over a small time interval.

Thus, recently, researchers are starting to move from deep learning architectures to hybrid and unsupervised methods (Bhatia et al., 2020; Feng et al., 2020). The recent review papers have mainly focused on the review of machine learning based models, statistical models, and urban flow prediction models for traffic forecasting (Boukerche et al., 2020; Sun, Aljeri et al., 2020; Wang & Boukerche, 2020; Xie et al., 2020; Y. Zhang, 2020). In this review, we aim to critically address the various existing deep learning architectures used for traffic flow prediction and the rising popularity of hybrid methods (Z. Duan et al., 2018; Y. Liu et al., 2017; Petersen et al., 2019). In the proposed work, we discuss the potential for unsupervised learning methods for traffic flow prediction.

3. Deep Neural Networks (DNN)

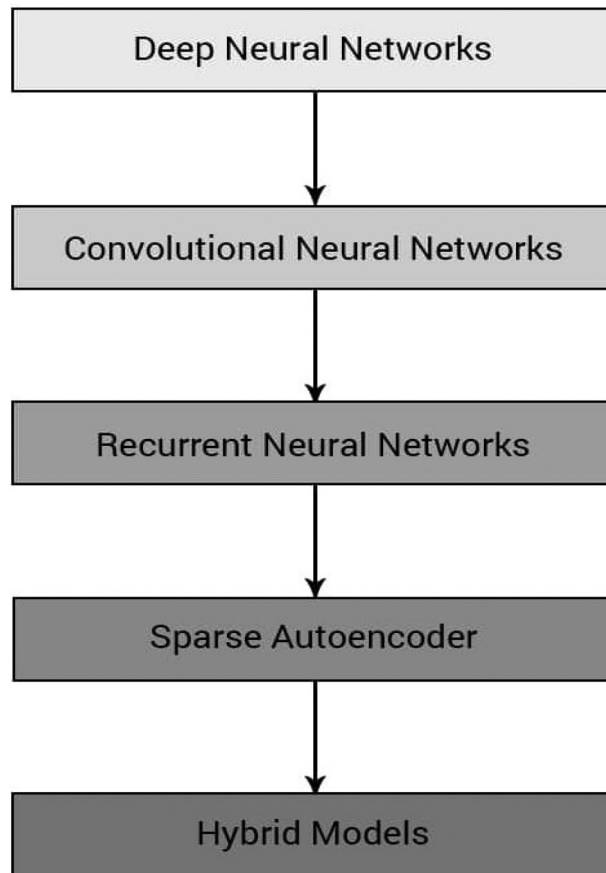
In this section, we review some of the Multilayer Neural Network (MLNN) ideas used in traffic flow prediction as in Figure 3 and also briefly describe various techniques to further optimize these networks. These NNs allow computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction (Lecun et al., 2015).

As traffic flow is affected by a variety of factors such as weather, accidents, holidays, etc., a deep architecture is suitable to solve the complex task of traffic flow prediction. This is now possible after Hinton proposed a breakthrough deep network that converted high-dimensional data to low-dimensional data which outperformed the PCA (Hinton & Salakhutdinov, 2006).

The authors in (Tampubolon & Hsiung, 2018) demonstrated hyper parameter tuning techniques to optimize the networks and yield better results to the generalization and over-fitting problem. They used a backpropagation algorithm to calculate the gradient and update the weights using Stochastic Gradient Descent (SGD). A dropout layer was added and batch normalization was done to ensure that the input distribution is standardized before passing the activations to the nonlinear layers.

A Cascaded Artificial Neural Network (CANN) was proposed in (S. Zhang et al., 2018) which consists of three Artificial Neural Networks (ANN): Long term, Medium-term, and Short term. The long-term network captures weekly recurrences in the traffic flow, the Medium-term network

Figure 3. MLNN techniques used in traffic flow prediction.



captures daily periodicity in traffic flow and the short-term network captures the numeric variations in trends of traffic flow. The CANN model showed better results upon using historical data.

A Wavelet Neural Network (WNN) introduced in (Q. Chen et al., 2021) uses a Morlet function as the wavelet basis function. It is a wavelet of a complex exponential function multiplied by a Gaussian window.

$$y = \cos(1.75x) \times e^{(-x^2)/2} \quad (1)$$

The proposed model uses one hidden layer with the Morlet function as the transfer function of the WNN. The input layer takes normalized inputs of traffic flow from 4 points, the hidden layer has 6 neurons and the output layer has 1 neuron. It was further improved by using the particle swarm optimization algorithm. The swarm algorithm iteratively calculates location updates on a d-dimensional search space based on the formula:

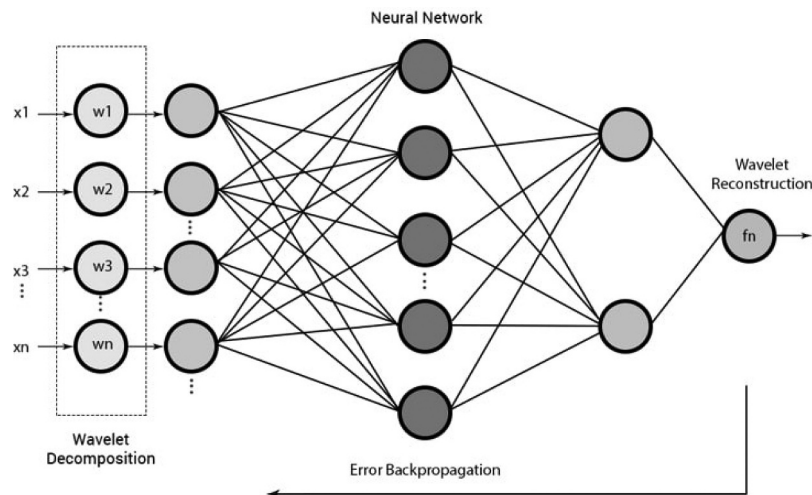
$$X_{id}^{t+1} = (X_{id}^t \cdot W_{ij}^t) + (V_{id}^t \cdot W_{ij}^t) + rand() \cdot g_{best}^t \quad (2)$$

Here V is the flight speed and X is the location. The improved algorithm offers an update mechanism for the weights W :

$$W_i = W_{max} - i(W_{max} - W_{min})/I_{max} \quad (3)$$

Here W_i is the i^{th} inertial weight and I_{max} is the maximum number of iterations. The algorithm uses the updates to iteratively reach an optimal solution. Figure 4 shows a representation of this model.

Figure 4. A representation of WNN model.



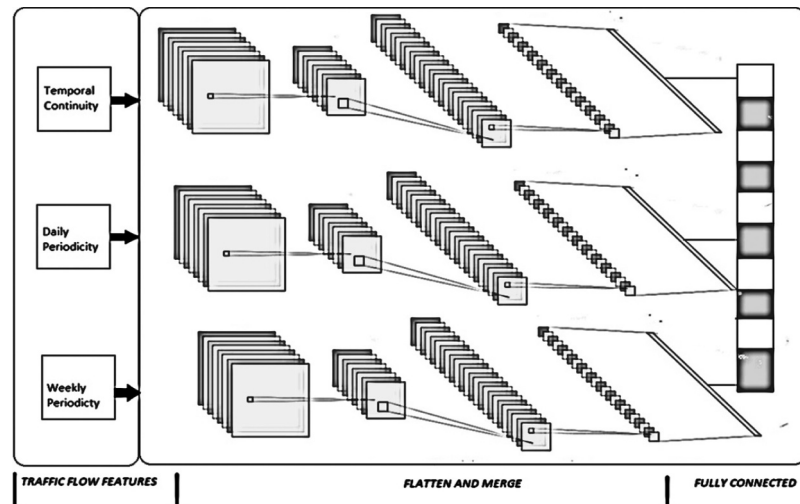
3.1. CNN

These are deep learning algorithms that possess the capabilities of taking in multi-dimensional inputs and carry complex computations with relative ease when compared to a conventional DNN. The Conv-nets prove to be an effective tool to capture spatial-temporal features with the help of filters that are self-learned. The pooling layers within a Conv-Net enhance the generalization ability of the model and reduce computation time. These features of CNNs act as a motivation for their application in the field of traffic flow prediction. To establish spatial correlations in the data, the authors introduced a Random Subspace learning based deep CNN (RSCNN) model in (Deng et al., 2019). The dataset was obtained from the Performance Measurement System (PeMS). For each location, three random subspaces are constructed and fed into three deep convolutional neural networks. The random subspace includes a batch of locations chosen from a candidate matrix which comprises locations with maximum spatial correlation. The model effectively considers correlations in traffic flow and produces promising results in high traffic situations. Although, the model fails to differentiate between weekdays and holidays, and other long-term periodicity. To overcome the previous shortcomings the next step taken by the authors was to represent locations using vectors and extract the similarities among them using the CBOW model. Negative sampling was adopted to reduce the training cost. Real data, average data, and mode data are fed to the CBOW model to generate a feature matrix. The feature matrix was fed to the convolutional neural network. The model also uses pool layers that serve the purpose of grouping. The Data Grouping CNN (DGCNN) introduced in (Xia et al., 2016) showed high accuracy for short-term prediction on being evaluated over the metrics of Root Mean Square Error (RMSE) and Mean Relative Error (MRE), but the network didn't achieve long term prediction.

Researchers have developed several methods to get higher accuracy on short-time prediction as well as long-time prediction. In (D. Yang et al., 2019), the authors fed real-time data, daily periodicity data, weekly periodicity data, and data regarding external factors such as events and holidays separately to the CNN. The outputs were then merged and flattened using a logistic regression layer. But the Multi-feature Fusion-CNN (MF-CNN) illustrated in Figure 5 managed to be accurate with only short-term prediction. The run time of the model did not prove to be optimum and the model didn't take into account the unprecedented changes due to accidents or other unforeseen circumstances.

Another interesting idea was reported in (P. Liu et al., 2019) to extract Spatio-temporal features to get better accuracy. The improved Spatio-temporal Residual Networks (ResNet) having four components: three describing Spatio-temporal relationships and one for scenario patterns captured both Spatio-temporal correlations as well as scenario patterns accurately by acquiring data from smart cards. The improved ResNet used a shortcut every three layers as compared to the

Figure 5. A representation of MF-CNN model.



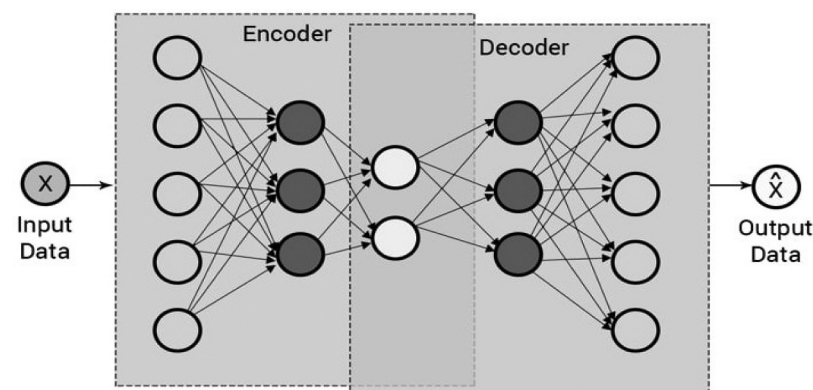
traditional ResNet that used a shortcut every two layers. The image-like tensor was divided into three factors: closeness, period, and trend. All these components had the same structure: one convolutional layer followed by two improved residual blocks which were then followed by a final convolutional layer. This defined the spatial relationships. The scenario patterns were a summation of the boarding flow and alighting flow for all time intervals which was normalized and encoded into a 1D matrix. This matrix was input into two fully connected layers to produce the output.

A 3D CNN (C. Chen et al., 2018) was used to learn the Spatio-temporal correlation features jointly from low-level to high-level layers for traffic data. Compared to 2D CNN, 3D CNN can model 3-dimension information owing to 3D convolution and 3D pooling operations. By preserving the temporal dependencies of the volumetric data resulting in an output volume, moreover, adopting the same kernel sharing across space and time dimensions, the model could take full advantage of spatial and temporal dependencies. This model did not rely on historical data to predict the future values and overlook spatial and external features. It also outperformed the ST-ResNet and the ConvLSTM models.

3.2. SAE

One of the earliest definitions of auto-encoders was given by (Ballard, 1987). They define auto-encoders or an auto-associative configuration as a model where the output is constrained to be identical to the input. Paper (Palm, 2012) explains that an auto-encoder or auto-associator is a discriminative graphical model that attempts to reconstruct its input signals. An SAE model is, as the name suggests, created by stacking auto-encoders to form a deep network, as shown in Figure 6. Stacking refers to taking the auto-encoder's output found on the layer below as the

Figure 6. SAE model.



input of the current layer (Bengio et al., 2013). SAEs are known to show promising results with traffic flow data as they inherently consider spatial and temporal correlations (Lv, Duan, Kang, Li, Wang et al., 2015b) and according to (Erhan et al., 2010), unsupervised pre-training helps prevent overfitting. Furthermore, SAEs do not require a labeled training set, being an unsupervised learning method, and thus, they can easily handle mislabeled data. However, they do have a significant disadvantage in situations where the hidden layer's dimensions are the same as or larger than the input layer. In such cases, the model can end up copying the input, without extracting more meaningful features (Bengio et al., 2007; Erhan et al., 2010). However, currently there exist two main approaches to tackle the issue of the model becoming potentially useless. The first is to use denoising auto-encoders (Erhan et al., 2010; Xu et al., 2018) and the second is to use sparse auto-encoders (Palm, 2012). An auto-encoder with sparsity constraints is called a sparse auto-encoder. All papers discussed in this section use sparse auto-encoders for their solution.

The first paper to use SAEs to predict traffic flow data is (Lv, Duan, Kang, Li, Wang et al., 2015) to the best of our knowledge. The model's training and test data were from the Caltrans PeMS database. Specifically, the traffic flow data for the first two months of 2013 comprised the training data, and the third month of 2013 was used as test data to carry out experiments. The paper uses a traditional SAE model trained in a greedy layer wise fashion. Kullback-Leibler (KL) divergence provides the sparsity constraint on the coding as $KL(p||p^*) = 0$ if $(p = p^*)$. Logistic regression is added as the last layer after the SAE layers for supervised traffic flow prediction. The SAE model, along with the logistic regression unit, comprises the architecture proposed by the paper.

Although it is straightforward to train the deep network by applying the back-propagation (BP) algorithm, networks trained with the BP method were inaccurate. Thus, the paper (Hinton & Salakhutdinov, 2006) uses a greedy layer wise unsupervised learning algorithm to pre-train the deep network from the bottom-up. Once pre-training is complete, the authors carried out fine-tuning using the BP algorithm with gradient-based optimization to tune the model's parameters in a top-down approach to obtain even more accurate results. The model was then used to predict traffic flow in intervals of 15, 30, 45, and 60-minutes. The paper concluded that the SAE model could discover nonlinear Spatio-temporal correlations from traffic data and performed better than Back-Propagation Neural Network (BPNN), Random Walk (RW), Support Vector Machine (SVM), and Radial Basis Function Neural Network (RBFNN) models.

Considering the model mentioned previously, another paper evaluated the proposed SAE model through 250 experimental tasks (Y. Duan et al., 2016). The model's performance was evaluated at different times of the day, both on weekdays and weekends. Further, the best combination of hyper-parameters was proposed for the same. Unlike the previous paper, data was collected from original vehicle detector stations rather than the processed global flow. Experimental results showed that the Mean Absolute Error (MAE) and RMSE during the day are larger than at night, while the MRE during the day is smaller than at night.

Although these studies tackled the main disadvantage of SAEs and obtained promising results, there were other causes for inaccuracies. For example, the low dimensionality of traffic flow data contributed to networks sticking to poor local minima due to large numbers of saddle points with zero gradients scattered on the landscape (Dauphin et al., 2014). However, large fluctuations and uncertainties in traffic flow data over hours can lead to overfitting of the model.

Thus, to combat these issues, (Zhou et al., 2017) introduces a boosting scheme for the SAE network to improve traffic flow forecasting accuracy. The boosting scheme uses prediction error, a measurement of the network's generalization ability, to retrain the SAEs by rearranging the training data, similar to (Huang et al., 2014). The authors use an adaptive ensemble scheme based on δ -agree AdaBoost regression to boost the model and tackle uncertainties. Accuracy is also improved by constructing an ensemble of SAEs. Experiments carried out used traffic flow data from four motorways A1, A2, A4, and A8, ending on Amsterdam's ring road (A10

motorway). The data provided was from May 20th, 2010 to June 24th, 2010, with one-minute aggregation collected by the MONICA sensor. Any zero values in the data due to incorrect measurements were filled by averaging the same moments of other weeks. The first 4 weeks were used for training, and the rest were used for testing. The training set was then divided further into 10 parts. The first nine were used for training the SAE, and the remaining one was used for cross-validation after every training epoch. Drop out regularization was used to prevent overfitting, and the limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) algorithm (Liu & Nocedal, 1989), a typical gradient descent algorithm, was used for optimization. It was observed that the model outperformed methods such as SVM Regression (SVR), Historical Average (HA), RW, autoregression, ANN, Kalman Filtering (KF), LSBoost, and traditional SAE. Although the proposed model could accurately predict heavy traffic scenarios, accuracy dropped when traffic flow was low such as early in the morning or late at night.

Building on the concept of using SAEs for traffic flow prediction in 2017, another novel method was proposed by (H. F. Yang et al., 2017). The authors used a stacked **auto-encoder Levenberg-Marquardt (SAE-LM) architecture**. The model was designed using the Taguchi method for optimized structure and to learn traffic flow features through layer-by-layer feature granulation with a greedy layer-wise supervised learning algorithm. The model was trained and tested with traffic flow data collected from the M6 freeway in the UK on working days between March 3rd, 2014, and March 28th, 2014, sensed by induction loops located at junctions J6, J7, J17, J18, J34, and J35. The dataset was divided into six road link data sets, where the first 15 days from each were used as the training set, and the remaining was used as the test set. The LM algorithm provides a numerical solution to the nonlinear problem of traffic flow prediction. The model uses a sigmoid activation function and uses the concept of the KL divergence penalty function as an extra penalty term, providing the sparsity constraint on the coding. LM is used as the last layer as it is a supervised learning algorithm that can fine-tune all parameters of the deep architecture. The evaluation results showed that the SAE-LM model with an optimized structure showed superior performance (about 90% accuracy rate). This result was achieved with five hidden layers (four auto encoders) and limited computational time. The model's main disadvantage is that it did not perform well if the observed traffic data has a highly smooth distribution. As traffic data is rarely smooth, the model works well for real-life applications. Further improvements can be made using an effective imputation method for missing data to apply to other datasets.

In 2018, (Jin et al., 2018) proposed a SAE model similar to (Bengio et al., 2013) and instead used the Rectified Linear unit (ReLU) as the activation and dropout regularization to prevent over-fitting. Furthermore, dropout prevents units from co-adapting. Each SAE layer is pre-trained with the greedy unsupervised learning algorithm to optimize the weights of the layer. The model is then fine-tuned using the BP algorithm. The authors trained the model using traffic flow data from the city ring way of Xi'an, Shaanxi, China, where data was collected every 5 minutes from the vehicle detector K27 + 000 from September to December 2017. The first three months were used as the training set, and the last month was used as the test set. The paper concluded that the SAE model performed better than SVM, DNNs, LSTM, and DBNs for 15 and 30-minute traffic flow prediction. Further refinement of the network could provide even more accurate results.

Recently, (Abdollahi et al., 2020) used the New York City Taxi and Limousine Commission trip records from January 2016 to June 2016 to predict traffic travel time- a concept closely related to traffic flow. The authors used SAEs with dropout regularization for feature extraction of traffic data as they found that this prevented overfitting. They used data augmentation and feature engineering techniques such as geospatial feature analysis, Principal Component Analysis (PCA), and k-means clustering. After obtaining a better feature set, deep SAEs were used to represent features in lower dimensions to decrease the chances of overfitting and increase accuracy. Finally, a deep multi-layer perceptron was used to make the architecture more robust. The algorithm was able to capture traffic flow dynamics. However, rare events such as heavy snow decreased the prediction accuracy of the model.

Although SAEs are found to work relatively accurately on their own, the trend seems to be towards hybrid models that exploit the advantages of LSTM and SAE. For instance, in recent times (Essien et al., 2021; Lin et al., 2019) used LSTM and SAE-based architectures to effectively capture Spatio-temporal relations in traffic data with decreased risk of overfitting.

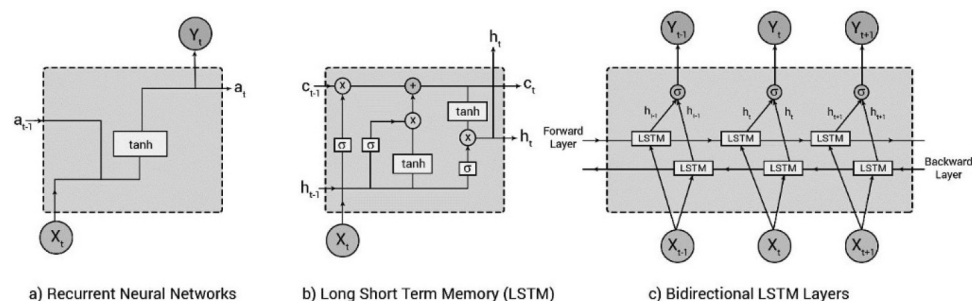
3.3. RNN

The failure of ANN and CNN in traffic flow prediction is mainly due to the Spatio-temporal nature of traffic. Therefore, RNN and its forms, which are illustrated in Figure 7, are widely used in traffic flow prediction. However, standard RNNs fail to solve problems that require long-term temporal dependencies due to the vanishing gradient and exploding gradient problem. A model called Long Short-Term Memory Recurrent Neural Network (LSTM RNN) was used to solve this problem (Hochreiter & Schmidhuber, 1997). The authors in this paper used the LSTM RNN to solve the traffic flow prediction problem (Tian & Pan, 2015). The primary objectives of LSTM RNN are to model long-term dependencies and determine the optimal time lags for time series problems. The dataset from Caltrans PeMS is used for building the model and comparing LSTM RNN with several well-known models. The generalization capability of LSTM RNN was validated by comparing it with other machine learning models namely RW, SVM, single layer Feed Forward Neural Network (FFNN), and SAE. The study did not account for spatial impact from neighbor observation stations and weather factors. Experiment results show that both MAPE and RMSE are lowest with different prediction intervals which proved that the proposed model achieved high accuracy.

However, it is difficult to solve exceedingly long-term dependencies, possibly because the LSTM errors increase as the sequence length increases. An LSTM+ model was proposed that can sense both long short-term memory and remarkably long distances (B. Yang et al., 2019). The LSTM+ model consists of four layers namely the mixed input layer, attention mechanism layer, the hidden layer, and the output layer. The sequence before the time step “t” in consideration is given to the attention layer, which captures most features of the long sequence. It is then passed to a softmax layer. The output of this softmax layer is given as the input to the LSTM model. In this way, the model aspires to capture most of the features of the past long sequence. In the attention mechanism layer, the input sequence is $x = (x_1, x_2, \dots, x_n)$, where it was experimentally proved that the values near the t^{th} time interval of each cycle have a high impact on the prediction value. Thus, only the data near the predicted time of each cycle t is given as the input to the attention layer to reduce redundancy. The long-distance problem arises in the prediction of traffic flow for the time intervals of 5 minute and 1 minute. The LSTM+ thus solved this problem and fared better than other models like BPNN, SVM, RBFNN.

Similar to LSTM, Gated Recurrent Units (GRU) was introduced to solve the vanishing gradient problem (Chung et al., 2015). Unlike the LSTM, GRU does not have a separate memory block which makes it easier and more efficient to train but it consists of two gates the reset and update gate which decide what should go to the output. The reset gate decides if any past information needs to be retained or deleted and the update gate decides the amount of past information to keep around. In recent years authors have tried using GRU to predict traffic (Fu et al., 2016). The PeMS dataset was used to train the

Figure 7. (a) RNN (b) LSTM (c) Bidirectional LSTM layers.



model. In this experiment, both LSTM and GRU models were trained and it showed that LSTM and GRU NNs have better performance than ARIMA models and GRU has a little better performance than LSTM. On 84% of the total time series, GRU NNs have better performance than LSTM NNs.

For many sequence processing tasks, it is useful to analyze the future as well as the past of a given point in the series. However, conventional recurrent neural nets are designed to analyze data in one direction only in the past. A better approach is provided by the bidirectional networks pioneered by Schuster and Baldi (Baldi et al., 1999; Schuster & Paliwal, 1997). A Deep Bidirectional LSTM (DBL) was introduced in (Essien et al., 2021) to solve the traffic flow problem. A total of 6 BiLSTM layers were used in this model. A deep hierarchy always leads to the vanishing gradient problem therefore residual connections were introduced among the DBL layers in a stack. The prediction architecture consists of four parts: the embedding layer, the DBL, the mean pooling layer, and the logistic regression layer. The data is mapped into the space of a 64—dimensional vector space. After the DBL encodes the time-aware traffic flow information, a sequence $h(0), h(1) \dots h(n-1)$ is produced. Then, the mean pooling layer extracts the mean values of the sequence over time intervals. Besides, the mean pooling layer makes the features encoded into a vector h . The vector is fed into the logistic regression layer at the top of the prediction architecture. The model was trained on the PeMS dataset and its results proved that DBL obtains a high prediction accuracy with the MAPE being 4.83% and the RMSE of DBL is 46.01%. The model outperformed ARIMA, SVM, DBN, SAE, and LSTM.

3.4. Hybrid networks

The field of Traffic Flow Prediction has seen the inception of hybrid models which aim to overcome the shortcomings of the state of the art models and also capitalize on their advantages.

A Traffic Flow Forecasting Network (TFFNet) was proposed (Selvaraju, et al., 2020) to predict short-term traffic flow. The TFFNet is made of two components one for spatiotemporal dependencies and the other for external influences. The traffic flow matrices are concatenated and fed into the first convolutional layer. The output from this is fed into a series of residual units to obtain output denoting spatiotemporal dependencies, XRes. A shorter connection is added to the framework. The output from this is concatenated to the output from the residual layers. A two-layer fully connected NN is used to embed external factors, and XExt describes the output. XRes and XExt are fused to obtain the final output XFinal. TFFNet_16 that consists of 16 residual units had an RMSE of 14.07. The low RMSE and relatively low training time make it the best version of the model.

The Concept of a Conv-LSTM model which is a combination of convolutional neural networks and a Long Term-Short Term memory network has had several renditions to it, in an attempt to capture the Spatio-temporal features of traffic flow effectively. One such model (Y. Liu et al., 2017) uses a Conv-LSTM to extract spatial-temporal features and a bidirectional LSTM to capture periodicity in the trends. The Conv-LSTM network receives a 1D time series vector as input. The vector undergoes 1D convolution, followed by pooling. This feature vector is then fed to an LSTM network to identify temporal features. The input to the Bi-LSTM includes time series info both before and after the forecast time. The output of the Bi-LSTM is further concatenated with that of the Conv-LSTM with the help of Fully Connected Layers. This Conv-LSTM hybrid does not predict short-term fluctuations effectively.

The authors propose an intelligent model for bus travel time prediction (Petersen et al., 2019) to exploit Spatio-temporal correlations in urban bus traffic. CNN's are also deployed for detecting spatial patterns across the links. The model applies convolutional filters in the input-to-state transitions as well as the state-to-state transitions of the LSTM. Here, the use of CNN's significantly reduces the number of parameters in comparison to a fully connected network. The model uses the encoder-decoder LSTM architecture along with a fully connected layer at the end, allowing predictions of future time steps with complex patterns. The 1D convolution can be extended to 2D to be able to predict across multiple bus routes. The model also uses RMSprop as an optimizer for efficient training. The model proves to be computationally heavy and does not effectively take into

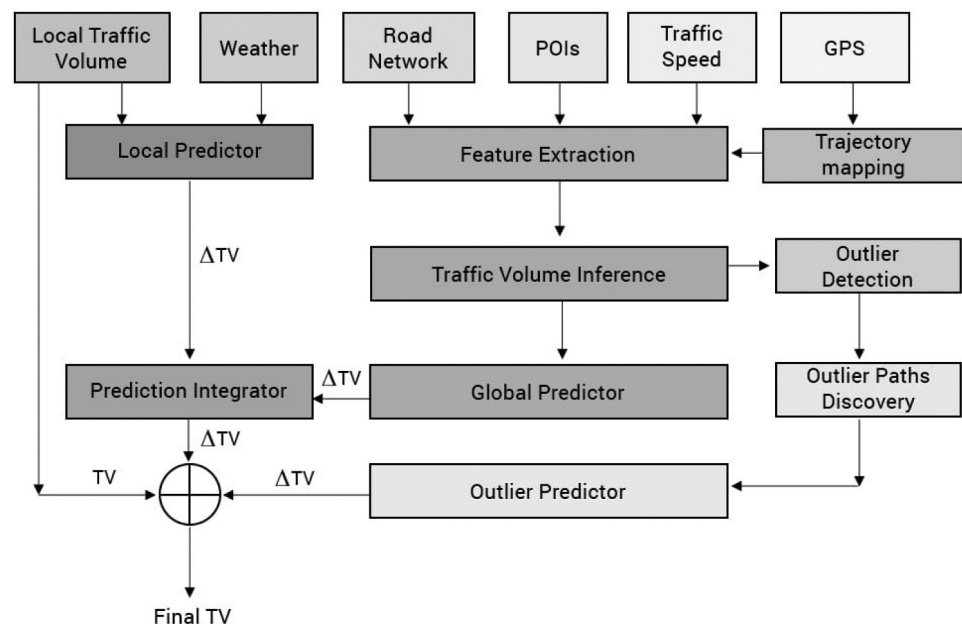
consideration the frequency of accidents, holidays, traffic signals at link intersections, weather conditions.

Several architectures also adapt to fragmented approaches with CNNs and LSTMs to combine their properties to reach optimum results. The hybrid deep neural network proposed in (Z. Duan et al., 2018) is fragmented into two heterogeneous sub-deep neural networks. One sub-network uses typical convolutional layers without pooling as no data should be lost in the process to accurately determine the spatial features. The other sub-network uses convolutional layers along with LSTMs to deduce the temporal features of traffic flow. Both the sub-networks are merged and averaged in the output layer of the network. The Epsilon-greedy policy is used in the process of learning. Implementing a residual network would prove to be beneficial in reducing training time, it compromises the accuracy of prediction as the ResNet tends to skip layers whereas the epsilon greedy policy proves to explore and garner enough substance from the available data to generate satisfactory results.

An attention model is introduced in (Wu et al., 2018) to use a speed matrix S spanning over space and time to learn an attention weight matrix A of the same size as S . The traffic flow matrix is multiplied with the attention matrix to form the weighted traffic flow matrix. The model uses convolutional layers for spatial feature mining and GRUs for temporal feature mining. The GRUs prove to be better than conventional LSTMs and are less expensive. The model also uses three 3-layer CNNs to extract near-term spatial features (CNN-n), daily periodicity (CNN-d), and weekly periodicity (CNN-w). The input data contains information regarding average speeds at specific time points during the previous day, the previous week and includes spatial information. This is given directly to CNN-d, CNN-w, and through the attention model to CNN-n forcing it to focus on near-term values. The outputs of these CNNs are combined and fed into a ReLu activated regression layer.

A rather unconventional hybrid model (Yu & Liu, 2019) uses 4 sections: local predictor, global predictor, prediction integrator, outlier predictor. This is shown in Figure 8. The local predictor manages to consider the temporal factors by adapting the method of linear regression. The global predictor accounts for spatial factors by using an unsupervised Bayesian network to predict traffic volume using data from neighboring roads. Conditional probability equations are used on the given data to iteratively calculate the values of parameters such as time, road type, observed data, and

Figure 8. An unconventional hybrid model.



points of interest features that determine traffic volume. A deep learning model is constructed with the help of these parameters by forming a stacked autoencoder using 3-layer sparse autoencoders along with a regression layer. The predictor integrator integrates the outputs of the local and global predictors using a regression tree. This outlier predictor takes into account the dynamic fluctuations in traffic flow inflection = $(k = 0, q-1(f_{i,j-k}-f_{i,t-k})^2)$. The vector f denotes features of road segment i at the given time step. Taking account of minimum inflection allows the model to eliminate recurring trends in traffic. A regression tree algorithm is used to obtain abnormal paths. The output of this section is added to that of the prediction generator. This model proposed gives a prediction accuracy above 90% for over 88% of the predictions. In recent times a graph convolutional neural network (Hu, 2021) has been developed to obtain the sensor node vector set and recurrent models are used to forecast the result. The nodes and the neighbor nodes are connected to represent the spatial correlation of traffic flow. Different moments of the same node are taken as a time series. The gated loop unit and attention mechanism process the time dynamics of the traffic flow. The spatial and temporal dependencies of traffic flow have been taken care of with the graph convolutional network and attention mechanism. Another paper (Zhang & Jiao, 2020) uses the attention mechanism with deep convolutions for short-term traffic flow prediction. The inception network is the base layer. Each layer obtains additional input from all the previous layers and uses its own features.

Because of these dense connections, the vanishing gradient problem arises. Then, the output of the base unit of the last layer is plugged into the regression layer to get the final prediction. In order to reduce the complexity 3×3 and 5×5 convolution operations, a 1×1 convolution operation are added and feature maps are obtained. Each of feature maps is plugged into a channel attention module to enhance the information of useful channels in the feature maps and suppress the information of redundant channels. The authors of this paper have achieved better RMSE results compared to conventional models like DCNN, Resnet, DNN, and dense net.

A hybrid CNN-LSTM model which is shown in Figure 9 is used to predict traffic flow at different time points in the future (Mihaita et al., 2019). Authors find that the CNN-LSTM model in this paper underperforms the LSTM model with its performance fluctuating. Temporal and Spatial features are incorporated and modeled by connecting the output of CNN to the input of each LSTM unit. The final prediction is made using a Fully Connected layer. Data profiling, outlier identification were done before the model was trained. The CNN-LSTM model manages to outperform traditional methods but the individual LSTM model outperformed the hybrid CNN-LSTM which indicated that complex deep learning models do not necessarily improve the prediction accuracy for our motorway flow prediction study. A comparison of various traffic flow prediction models is shown in Table 1.

Figure 9. A hybrid CNN-LSTM model.

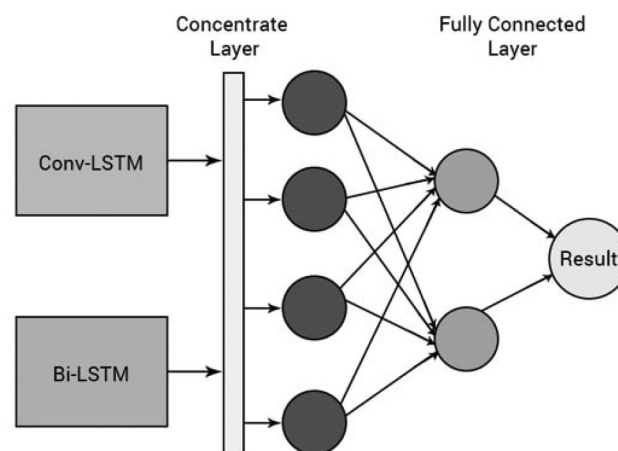


Table 1. Comparison table of traffic flow prediction models

Category: RNN

model	Dataset	Model Details	Features Considered	Inference
LSTM & GRU (Fu et al., 2016)	PeMS	A basic GRU and LSTM network	Spatio-Temporal	GRUs have outperformed LSTMs in traffic flow prediction
LSTM+(LSTM with feature enhancement) (B. Yang et al., 2019)	PeMS	Consists of 4 layers the mixed input layer, attention mechanism layer, the hidden layer and the output layer.	Spatio-Temporal	Superior than LSTM because it overcomes the shortcomings of the LSTM module when it comes to long sequences
Category: ANN				
IPSO-WNN (Q. Chen et al., 2021)	Intersection of an anonymous city	A particle swarm optimization algorithm along with a basic WNN network	Spatio-Temporal	The prediction accuracy of IPSO-WNN is higher because it solves the problem that the wavelet neural network is easy to fall into the local minimum
CANN (S. Zhang et al., 2018)	video surveillancecameras in Xiamen	The proposed Cascaded ANN consists of three ANNs: Long term, Medium term and Short term.	Spatial and Temporal and other external factors like weather, holidays	CANN model showed better results upon using historical data to forecast traffic flow for next 60 mins.
Category: SAE				
SAE (Lv, Duan, Kang, Li, Wang et al., 2015)	PeMS	KL divergence provides sparsity constraint. Logistic regression was used in the last layer for supervised trafficflow prediction. Greedy layer wise unsupervised learning algorithm was used to pretrain the data and BP was used for fine tuning	Spatial and Temporal	The predictor used in the paper is logistic regression. Further improvements can be made by using even more powerful predictors. (Van Der Voort et al., 1996) conducted 250 experiments to evaluate the performance of the network and found that MAE and RMSE was larger during the day. Further, in the experiments, the model was applied to weekday and weekends separately.
δ-agree AdaBoost SAE (Zhou et al., 2017)	A1, A2, A4, and A8, ending on Amsterdam's ring road (A10 motorway) collected by the MONICA sensor	Drop out regularization was used to prevent overfitting. The LBFGS algorithm was used for optimization.	Spatial and Short-term Temporal	The model worked well in heavy traffic scenarios but accuracy dropped with low traffic flow. For instance, traffic predictions for early in the morning or late at night were less accurate. Further, averaging was done to account for missing data. Effective data imputation techniques can be employed to improve the model's performance.
SAE-LM (H. F. Yang et al., 2017)	M6 freeway in the UK sensed by induction loops located at junctions J6, J7, J17, J18, J34, and J35	The model was designed using the Taguchi method. The LM algorithm was used to improve accuracy. KL divergence penalty function was used as the sparsity constraint.	Spatial and Short-term Temporal	The model performs well even with limited computational time. It was observed that the model was less accurate with highly smooth traffic data distributions. However, as traffic flow data is rarely smooth, this does not pose to be a major disadvantage.

(Continued)

Table1. (Continued)

Category: RNN

model	Dataset	Model Details	Features Considered	Inference
SAE (Jin et al., 2018)	City ring way of Xi'an, Shaanxi, China from the vehicle detector K27 + 000	Greedy layer wise unsupervised learning algorithm was used to pre train the data and BP was used for fine tuning. Dropout regularization was used.	Spatial and Short-term Temporal	As per experiments conducted by the authors, as the time interval increased, accuracy of the model decreased. Considering this, the model can be improved to account for longer intervals of time.
SAE + Multilayer Perceptron (Abdollahi et al., 2020)	New York City Taxi and Limousine Commission (TLC) trip records	Dropout regularization was used. Data augmentation and feature engineering techniques such as geospatial feature analysis, PCA, and k-means clustering were conducted. A deep multi-layer perceptron was used to make the architecture more robust.	Several features were considered including geospatial, weather, temporal, weekdays and weekends, speed	This paper talked about travel time prediction but was capable of capturing traffic flow dynamics. This was done using feature engineering techniques. However, the model showed less accurate results when rare events such as heavy snowfall occurred.
Category: hybrid				
Traffic Flow Forecasting Network (TFF Net) (Sun, Wu, Xiang et al., 2020)	Private Dataset—Wuhan, China	Hidden Markov Model, Residual Layers, CNNs	Spatial-Temporal, Short term	TFFNet manages to extract spatial-temporal features for short term prediction, with increased flexibility as desired external factors can be taken into consideration
CNN-LSTM (Y. Liu et al., 2017)	PeMS	1D CNN-LSTM, Bidirectional LSTM	Spatial-temporal, Long term, External factors	Suitable for long-term prediction but unable to predict short term fluctuations. Time series input includes data points before and after the forecast time.
CNN-LSTM (Petersen et al., 2019)	GPS Data—Automated Vehicle Location Systems	CNN, Encoder-Decoder	Spatial-Temporal, Long term	Model is flexible as multidimensional CNNs could be used to track alternate routes. Computationally heavy model doesn't consider fluctuations due to external factors
Deep Hybrid CNN-LSTM (Z. Duan et al., 2018)	Taxi GPS Data	CNN-LSTM subnetworks, Fully Connected layers	Spatial-Temporal	Efficient extraction of Spatial-temporal features. Use of Epsilon greedy policy results in increased accuracy.
DNN-BTF Model (Wu et al., 2018)	PeMS	CNNs, GRUs, Attention matrix	Spatial-Temporal, Temporal Correlations	Stronger correlations with recent past are quantified effectively. Weekly and daily periodicity is taken into consideration. GRUs prove to be less expensive than conventional LSTM.

(Continued)

Table1. (Continued)

Category: RNN

model	Dataset	Model Details	Features Considered	Inference
Hybrid Predictor (Yu & Liu, 2019)	Taxi GPS-Trajectory data, Road Network data—Shanghai Dalian Road Tunnel	Local Predictor—Regression Layer Global Predictor—Bayesian Network, Stacked Sparse Auto-encoders Predictor Integrator, Outlier Predictor - Regression Tree	Spatial temporal, Short term fluctuations	A fragmented approach to extract Spatial-Temporal features. The model accounts for dynamic fluctuations in traffic flow. Although the model is fairly complex in nature.
CNN-LSTM (Mihaita et al., 2019)	Monitoring Stations—Sydney Motorway	CNN-LSTM, Fully Connected Layers	Spatial Temporal, Short term	Spatial temporal factors are extracted, yet the model fails to outperform the LSTM model.
Category: CNN				
Improved Spatio-Temporal Residual Networks (P. Liu et al., 2019)	Beijing Transportation Smart Card Data	One convolutional layer followed by two improved residual blocks which was then followed by a final convolutional layer.	Spatio-temporal correlations and scenario patterns.	Achieved better results than the four base line models. More heterogeneous data could have been considered. There is scope for including the explainability of the model.
Multi feature Fusion-CNN (MF-CNN) (D. Yang et al., 2019)	JPEA, PeMS	Real time data, daily periodicity data, weekly periodicity data and data regarding external factors such as events and holidays separately to the CNN. The outputs were then merged and flattened using a logistic regression layer	Spatio-Temporal Matrices	Managed to be accurate with only short term prediction. Run time of the model was not optimum. The model didn't take into account the unprecedented changes due to accidents or other unforeseen circumstances.
Deep Convolutional Neural Networks (Deng et al., 2019)	PeMS	Random subspace learning based deep CNN (RSCNN), Deep CNN	Spatio-Temporal, Short Term	The model effectively considered correlations in traffic flow and produced promising results in high traffic situations. It failed to differentiate between weekdays and holidays, other long term periodicities.
The Data Grouping CNN (DGCNN) (Xia et al., 2016)	Real Vehicle Passage records from the surveillance system of a major metropolitan area	CBOW model Represented locations as vectors.	Spatio—relations, Short Term	Showed high accuracy for short term prediction on being evaluated over the metrics of RMSE and MRE, but the network didn't achieve long term prediction.
3D CNN (C. Chen et al., 2018)	NYC Bike dataset, BJTaxi dataset	Preserving the temporal dependencies of the volumetric data resulting in an output volume	Spatio-temporal correlation features jointly from low-level to high-level layers.	Adopting the same kernel sharing across space and time dimensions allowed model could take full advantage of spatial and temporal dependencies. It did not rely on historical data to predict the future values and overlook spatial and external features. Out-performed the ST-ResNet and the ConvLSTM models.

Table 2. Data from the Caltrans PeMS

5 Minutes	Lane 1 Flow (Veh/ 5 Minutes)	# Lane Points	% Observed
04/03/2016 0	16	1	100
04/03/2016 5	10	1	100
04/03/2016 10	11	1	100
04/03/2016 15	11	1	100
04/03/2016 20	6	1	100

4. Simulation

Data was obtained from the Caltrans PeMS. Data was collected in real-time from individual detectors spanning the freeway system across all major metropolitan areas of the State of California.

The first 5 rows of the dataset look as in [Table 2](#).

The model was trained using code from (GitHub, 2021). Different optimizers to change the hyperparameters were used to see how well the model fares. LSTM and GRU architectures were employed and therefore compared in this study. Different optimizers used here are Stochastic Gradient Descent (SGD), Adam, and rmsprop. It was trained on an NVIDIA GeForce 1660 Ti GPU. This simulation was done to understand the deviation from the actual data to what the model predicts when trained with different optimizers.

4.1. RMSprop

To deal with the vanishing and exploding gradient problem, RMSprop was developed as a stochastic technique for mini-batch learning. RMSprop deals with the above issues by using a moving average of squared gradients to normalize the gradient. This normalization balances the step size (momentum), decreasing the step for large gradients to avoid exploding and increasing the step for small gradients to avoid vanishing. [Figure 10](#) shows the comparison of traffic flow prediction for a duration of 20 minutes. Both LSTM and GRU were able to predict the traffic efficiently.

4.2. Adam

Adam adapts the parameter learning rates in real-time based on the average of the first and second moments. It calculates an exponential moving average of the gradient as well as the squared gradient. Then, the parameters beta1 and beta2 can control the decay rates of both moving averages. Finally, bias correction estimates are run before updating the parameters. Tests are conducted to study the deviation of LSTM and GRU from true data using Adam optimizer.

Figure 10. Traffic flow prediction with RMSprop optimizer for 20 data points.

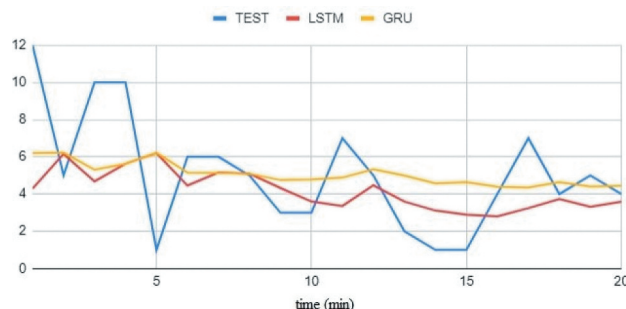


Figure 11 shows the comparison of traffic flow prediction using Adam optimizer for a short duration of 20 minutes.

4.3. Sgd

Batch gradient descent performs redundant computations for large datasets, as it re-computes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online. Figure 12 shows the comparison of traffic flow prediction for a duration of 20 minutes. In our tests, we have observed a higher deviation in the predicted data from the true data for the Sgd optimizer for a short duration.

4.4. Comparison discussion of optimizers

RMSProp is developed to overcome the shortcomings of initial optimizers like AdaGrad. In AdaGrad, the learning rate decays very aggressively. As a result, after a while, the frequent parameters will start receiving very small updates because of the decayed learning rate. To avoid this, RMSProp decays the denominator and prevents its rapid growth. The difference between RMSProp and AdaGrad is that AdaGrad gets stuck when it is close to convergence and it is no longer able to move in the vertical. RMSProp avoids this by being less aggressive on the decay. The Adam works similar to RMSProp with respect to how the decay takes place. In addition to that, the update rule for Adam is very similar to RMSProp, except we look at the cumulative history of gradients as well. Here, we control the rate of gradient descent in such a way that there is minimum oscillation when it reaches the global minimum while taking big enough steps (step-size) so as to pass the local minima hurdles along the way. In the tests conducted above, both RMSProp and ADAM optimizers were able to predict the traffic flow efficiently. However, we have observed higher traffic flow prediction errors using SGD optimizer compared to RMSProp and Adam.

A comparison of the results of Adam, RMSprop, and SGD optimizers on the different models is provided in Table 3, and a comparative analysis of various architectures used in traffic flow prediction is provided in Table 4.

Figure 11. Traffic flow prediction with Adam optimizer for 20 data points.



Figure 12. Traffic flow prediction with SGD optimizer for 20 data points.

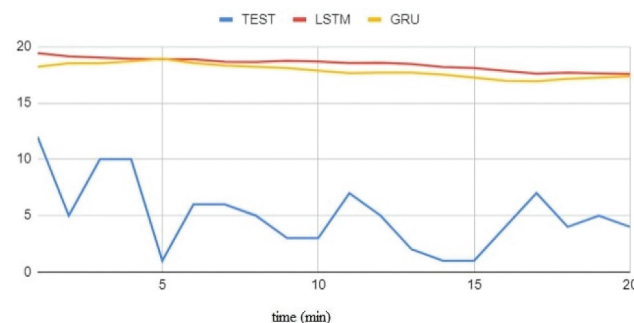


Table 3. Adam, RMSprop, SGD optimizers on the different models. (20 data points)

Index	Test	RMSPROP		ADAM		SDG	
		LSTM	GRU	LSTM	GRU	LSTM	GRU
1	12	4.266946	6.2062680	5.517144	6.634330	19.423101	18.211733
2	5	6.149686	6.2256785	6.994502	7.363862	19.13633	18.519939
3	10	4.683443	5.3094500	5.825087	6.736211	19.019585	18.509178
4	10	5.630908	5.6277200	7.639332	7.615730	18.921139	18.682638
5	1	6.202302	6.2164810	8.505941	9.622147	18.861729	18.901911
6	6	4.454712	5.1483600	6.310970	7.204504	18.863972	18.547182
7	7	5.161488	5.1331120	7.466517	7.633396	18.648914	18.329147
8	5	5.098215	5.0920150	6.745287	6.787171	18.622801	18.216377
9	3	4.324950	4.7523860	5.892681	6.016531	18.729589	18.102303
10	3	3.600009	4.7863040	5.144133	5.251890	18.667931	17.870478
11	7	3.358943	4.8848960	4.678200	4.914774	18.548157	17.635242
12	5	4.467793	5.3303800	5.039211	5.481043	18.551779	17.680256
13	2	3.595536	4.9956360	4.676708	5.542349	18.460207	17.693981
14	1	3.111748	4.5703720	4.368120	4.857191	18.187826	17.508757
15	1	2.895541	4.6446050	3.943196	4.490931	18.095139	17.241417
16	4	2.803970	4.3946860	3.716256	4.230009	17.832205	16.969788
17	7	3.241487	4.3541990	4.080602	4.392344	17.602140	16.916096
18	4	3.732073	4.6394800	4.530049	4.953875	17.686455	17.141809
19	5	3.312398	4.4020660	4.357851	4.866009	17.624826	17.245544
20	4	3.591156	4.4378440	4.824841	5.136835	17.581087	17.350965

Table 4. Comparative analysis of various architectures presented in the paper

MODEL	STRENGTHS	WEAKNESS
ANN	<ul style="list-style-type: none"> • Can perform nonlinear approximation • Does not make assumptions on distribution of data 	<ul style="list-style-type: none"> • Fails to solve complex problems • ANN cannot capture sequential information in the input data which is required for dealing with sequence data • The gradient vanishes or explodes as it propagates backward which leads to vanishing and exploding gradient.
WNN	<ul style="list-style-type: none"> • Requires smaller training time compared to Multilayer Perceptron • Fast Convergence • Fault-tolerance • Time frequency localization of wavelet transform 	<ul style="list-style-type: none"> • Difficulty of the choice of mother wavelet function • Lacking basis while selecting parameters
CNN	<ul style="list-style-type: none"> • High Accuracy in image recognition problems • Little dependence on pre-processing. 	<ul style="list-style-type: none"> • Requires a large amount of training data to obtain optimal features • Highly complex process for object detection
RNN	<ul style="list-style-type: none"> • Can model time dependency in data. • Each pattern can be assumed dependent on previous ones unlike ANN • Can be used for dynamic problems 	<ul style="list-style-type: none"> • Requires multiple parameters to update. • Suffers from exploding gradient

(Continued)

Table 4. (Continued)

MODEL	STRENGTHS	WEAKNESS
SAE	<ul style="list-style-type: none"> Reduces the dimensionality of data that is being used Great for feature extraction 	<ul style="list-style-type: none"> An auto encoder learns to capture as much information as possible rather than as much relevant information as possible. High time complexity because of multiple forward pass
HYBRID	<ul style="list-style-type: none"> Best at handling spatio-temporal Correlations because it combines the advantages of both CNN and LSTM 	<ul style="list-style-type: none"> Although hybrid networks which combine the advantages of both CNN and LSTM seem to be the most effective approach to solve traffic flow prediction in some cases, as shown in our paper, Hybrid networks have failed to outperform LSTM networks.

5. Conclusion

The movement of goods and humans is an integral part of existence. With the increase in the population and the necessity of social wellbeing of humans, travel is exponentially growing. As technology is evolving day by day so is the number of vehicles increasing. With this rapid rate of increase in vehicle, management of movement of vehicles is very critical. Vehicular management helps in optimizing the travel time and cost of the travel. For developing a precise vehicular management system, it is essential to have accurate background information. Traffic flow is one of the most important data which is required for developing a precise vehicular management system.

This paper presents a review of recent deep learning approaches in the field of traffic flow prediction. Most of the contributions are application based while very few articles have a strong contribution to theory. Deep learning models for traffic forecasting have shown promising results to represent the non-linearity of traffic flow prediction. While there are several advantages to using the deep learning models to predict traffic flow individually, there are significant disadvantages also. Thus, recently, researchers are starting to move from deep learning architectures to hybrid and unsupervised methods. This review addressed the various existing deep learning architectures used for traffic flow prediction and the rising popularity of hybrid methods.

Funding

No financial assistance received from any agency to carry out the work mentioned.

Author details

Anirudh Ameya Kashyap¹
 Shravan Raviraj
 Ananya Devarakonda
 Shamanth R Nayak K
 Santhosh K V
 E-mail: kv.santhu@gmail.com
 ORCID ID: <http://orcid.org/0000-0003-4394-5947>
 Soumya J Bhat

¹ Manipal Academy of Higher Education, Manipal Institute of Technology, Manipal, India.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Compliance with Ethical Standards

This article does not contain any studies involving human or animal subjects.

Citation information

Cite this article as: Traffic flow prediction models – A review of deep learning techniques, Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda,

Shamanth R Nayak K, Santhosh K V & Soumya J Bhat, *Cogent Engineering* (2022), 9: 2010510.

References

- Abdollahi, M., Khaleghi, T., & Yang, K. (2020). An integrated feature learning approach using deep learning for travel time prediction. *Expert Systems with Applications*, 139, 112864. <https://doi.org/10.1016/j.eswa.2019.112864>
- Ahmed Yasser, M. Z., & Kader, N. A. (2017). VANET routing protocol for V2V implementation: A suitable solution for developing countries. *Cogent Engineering*, 4(1), 1362802. <https://doi.org/10.1080/23311916.2017.1362802>
- Ali, U., & Mahmood, T. (2018). Using deep learning to predict short term traffic flow: A systematic literature review. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICTS*, 222. <https://doi.org/10.1007/978-3-319-93710-611>
- Bae, B., Kim, H., Lim, H., Liu, Y., Han, L. D., & Freeze, P. B. (2018). Missing data imputation for traffic flow speed using spatio-temporal cokriging. *Transportation Research Part C: Emerging Technologies*, 88, 124–139. <https://doi.org/10.1016/j.trc.2018.01.015>
- Baldi, P., Brunak, S., Frasconi, P., Soda, G., & Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15

- (11), 937–946. <https://doi.org/10.1093/bioinformatics/15.11.937>
- Ballard, D. H. (1987). Modular learning in neural networks. *Aai*, 279–284. <https://www.aai.org/Papers/AAAI/1987/AAAI87-050.pdf>
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828. <https://doi.org/10.1109/TPAMI.2013.50>
- Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Adv. Neural Inf. Process. Syst.*, (January), 153–160. <https://proceedings.neurips.cc/paper/2006/file/5da713a690c067105aeb2fae32403405-Paper.pdf>
- Bhatia, J., Dave, R., Bhayani, H., Tanwar, S., & Nayyar, A. (2020). SDN-based real-time urban traffic analysis in VANET environment. *Computer Communications*, 149 (October 2019), 162–175. <https://doi.org/10.1016/j.comcom.2019.10.011>
- Boukerche, A., Tao, Y., & Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer Networks*, 182, 107484. <https://doi.org/10.1016/j.comnet.2020.107484>
- Chen, C., Li, K., Teo, S. G., Chen, G., Zou, X., Yang, X., Vijay, R. C., Feng, J., Zeng, Z. (2018). Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction. *Proc. - IEEE Int. Conf. Data Mining, ICDM, 2018-Novem* (61625202), 893–898. <https://doi.org/10.1109/ICDM.2018.00107>
- Chen, C., Li, K., Teo, S. G., Zou, X., Li, K., & Zeng, Z. (2020). Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(4), 1–23. <https://doi.org/10.1145/3385414>
- Chen, K., Chen, F., Lai, B., Jin, Z., Liu, Y., Li, K., Wei, L., Wang, P., Tang, Y., Huang, J., Hua, X. (2020). Dynamic spatio-temporal graph-based CNNs for traffic flow prediction. *IEEE Access*, 8, 185136–185145. <https://doi.org/10.1109/ACCESS.2020.3027375>
- Chen, Q., Song, Y., & Zhao, J. (2021). Short-term traffic flow prediction based on improved wavelet neural network. *Neural Comput. Appl.*, 33, 8181–8190. <https://doi.org/10.1007/s00521-020-04932-5>
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015). Gated feedback recurrent neural networks. Accessed 10 12 2020 <https://arxiv.org/pdf/1502.02367v4.pdf>
- Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2, 2933–2941. <https://dl.acm.org/doi/10.5555/2969033.2969154>
- Deng, S., Jia, S., & Chen, J. (2019). Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data. *Applied Soft Computing*, 78, 712–721. <https://doi.org/10.1016/j.asoc.2018.09.040>
- Dong, X., Lei, T., Jin, S., & Hou, Z. (2018). Short-term traffic flow prediction based on XG-Boost. 7th Data Driven Control and Learning Systems Conference, 25–27 May 2018. IEEE Enshi, China. <https://doi.org/10.1109/DDCLS.2018.8516114>
- Duan, Y., Lv, Y., & Wang, F. Y., “Performance evaluation of the deep learning approach for traffic flow prediction at different times,” *Proc. - 2016 IEEE Int. Conf. Serv. Oper. Logist. Informatics, SOLI 2016*, pp. 223–227, 2016, <https://doi.org/10.1109/SOLI.2016.7551691>.
- Duan, Z., Yang, Y., Zhang, K., Ni, Y., & Bajgain, S. (2018). Improved deep hybrid networks for urban traffic flow prediction using trajectory data. *IEEE Access*, 6, 31820–31827. <https://doi.org/10.1109/ACCESS.2018.2845863>
- Erhan, D., Bengio, Y., Courville, A., Manzagol, P., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, 625–660. <https://www.jmlr.org/papers/volume11/erhan10a/erhan10a.pdf>
- Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*, 24, 1345–1368. <https://doi.org/10.1007/s11280-020-00800-3>
- Feng, B., Xu, J., Lin, Y., & Li, P. (2020). A period-specific combined traffic flow prediction based on travel speed clustering. *IEEE Access*, 8, 85880–85889. <https://doi.org/10.1109/ACCESS.2020.2992657>
- Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. 31st Youth Academic Annual Conference of Chinese Association of Automation, 11–13 Nov. 2016. IEEE: Wuhan, China. <https://doi.org/10.1109/YAC.2016.7804912>
- Gehring, J., Miao, Y., Metze, F., & Waibel, A. (2013). Extracting deep bottleneck features using stacked auto-encoders. *International Conference on Acoustics, Speech and Signal Processing*, 26–31 May 2013. IEEE, Vancouver, Canada. <https://doi.org/10.1109/ICASSP.2013.6638284>
- Gerla, M., Lee, E. K., Pau, G., & Lee, U., “Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds,” *IEEE World Forum on Internet of Things*, 6–8 March 2014, Seoul, Korea. IEEE, 2014, <https://doi.org/10.1109/WF-IoT.2014.6803166>.
- GitHub. *GitHub - xiaochus/TrafficFlowPrediction: Traffic flow prediction with neural networks (SAEs, LST, GRU).* <https://github.com/xiaochus/TrafficFlowPrediction> (accessed February 13, 2021)
- Good Fellow, I. J., Mirza, M., Courville, A., & Bengio, Y., “Multi-prediction deep Boltzmann machines,” *Proceedings of the 26th international conference on neural information processing systems*, Lake Tahoe, USA.1 <https://papers.nips.cc/paper/2013/hash/0bb4aec1710521c12ee76289d9440817-Abstract.html>, 2013.
- Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *International Conference on Acoustics, Speech and Signal Processing*, 26–31 May 2013, Vancouver, Canada. <https://doi.org/10.1109/ICASSP.2013.6638947>
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hu, Y., “Research on city traffic flow forecast based on graph convolutional neural network,” 2nd *International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering*, 26–28 March 2021, Nanchang, China, 2021, pp. 269–273, <https://doi.org/10.1109/ICBAIE52039.2021.9389951>.
- Huang, W., Zhang, N., Hu, W., Hong, H., song, G., & Xie, K. (2014). Dynamic boosting in deep learning using reconstruction error, *International Joint Conference*

- on *Neural Networks*, 6–11 July 2014, Beijing, China. <https://doi.org/10.1109/IJCNN.2014.6889534>
- Jin, Y., Xu, W., Wang, P., & Yan, J. (2018). SAE network: A deep learning method for traffic flow prediction, *International Conference on Information, Cybernetics, and Computational Social Systems*, 16–19 Aug. 2018, Hangzhou, China. <https://doi.org/10.1109/ICCSS.2018.8572451>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lee, S., & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, 1678(1), 179–188. <https://doi.org/10.3141/1678-22>
- Li, Q., Tan, H., Wu, Y., Ye, L., & Ding, F. (2020). Traffic Flow Prediction with Missing Data Imputed by Tensor Completion Methods. *IEEE Access*, 8, 63188–63201. <https://doi.org/10.1109/ACCESS.2020.2984588>
- Lin, F., Xu, Y., Yang, Y., & Ma, H. (2019). A spatial-temporal hybrid model for short-term traffic prediction. *Mathematical Problems in Engineering*, 2019, 4858546. <https://doi.org/10.1155/2019/4858546>
- Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1–3), 503–528. <https://doi.org/10.1007/BF01589116>
- Liu, L., Chen, R. C., Zhao, Q., & Zhu, S. (2019). Applying a multistage of input feature combination to random forest for improving MRT passenger flow prediction. *Journal of Ambient Intelligence and Humanized Computing*, 10(11), 4515–4532. <https://doi.org/10.1007/s12652-018-1135-2>
- Liu, P., Zhang, Y., Kong, D., & Yin, B. (2019). Improved spatio-temporal residual networks for bus traffic flow prediction. *Applied Sciences*, 9(4), 615. <https://doi.org/10.3390/app9040615>
- Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017). Short-term traffic flow prediction with Conv- LSTM, *International Conference on Wireless Communications and Signal Processing*, 11–13 Oct. 2017, Nanjing, China. <https://doi.org/10.1109/WCSP.2017.8171119>
- Lu, Z., Xia, J., Wang, M., Nie, Q., & Ou, J. (2020). Short-term traffic flow forecasting via multi-regime modeling and ensemble learning. *Applied Sciences*, 10(1), 356. <https://doi.org/10.3390/app10010356>
- Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015b, April). Traffic flow prediction with big data: A deep learning approach. *IEEE, Trans. Intell. Transp. Syst.*, 16(2), 865–873. <https://doi.org/10.1109/TITS.2014.2345663>
- Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE, Trans. Intell. Transp. Syst.*, 16(2), 865–873. <https://doi.org/10.1109/TITS.2014.2345663>
- Makaba, T., Doorsamy, W., & Paul, B. S. (2020). Exploratory framework for analysing road traffic accident data with validation on Gauteng province data. *Cogent Engineering*, 7(1), 1834659. <https://doi.org/10.1080/23311916.2020.1834659>
- Mihaita, A. S., Li, H., He, Z., & Rizioiu, M. A. (2019). Motorway traffic flow prediction using advanced deep learning, *Intelligent Transportation Systems Conference*, 27–30 Oct. 2019, Auckland, New Zealand. 1683–1690. <https://doi.org/10.1109/ITSC.2019.8916852>
- Nguyen, H., Kieu, L. M., Wen, T., & Cai, C. (2018, November). Deep learning methods in transportation domain: A review. *IET Intelligent Transport Systems*, 12(9), 998–1004. <https://doi.org/10.1049/iet-its.2018.0064>
- Osipov, V., Nikiforov, V., Zhukova, N., & Miloserdov, D. (2020). Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers. *Neural Comput. Appl.*, 32, 14885–14897. <https://doi.org/10.1007/s00521-020-04843-5>
- Palm, R. B. (2012). Prediction as a candidate for learning deep hierarchical models of data, Technical University of Denmark, DTU Informatics. <https://www2.imm.dtu.dk/pubdb/pubs/6284-full.html>
- Petersen, N. C., Rodrigues, F., & Pereira, F. C. (2019). Multi-output bus travel time prediction with convolutional LSTM neural network. *Expert Systems with Applications*, 120(15), 426–435. <https://doi.org/10.1016/j.eswa.2018.11.028>
- Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks, *International Conference on Acoustics, Speech and Signal Processing*, 19–24 April 2015, South Brisbane, Australia. <https://doi.org/10.1109/ICASSP.2015.7178838>
- Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4), 778–784. <https://doi.org/10.1109/TASLP.2014.2303296>
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. <https://doi.org/10.1109/78.650093>
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vision*, 128(336–359). <https://link.springer.com/article/10.1007/s11263-019-01228-7>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition, *International Conference on Learning Representations*, May 7–9, 2015, San Diego, USA. <https://arxiv.org/abs/1409.1556>
- Smith, B. L., Demetsky, M. J., & . (1994). Short-term traffic flow prediction: Neural network approach. *Transp. Res. Rec.*, 98–104, 0361–1981 <https://onlinepubs.trb.org/Onlinepubs/trr/1994/1453/1453-011.pdf>
- Sun, P., Aljeri, N., & Boukerche, A. (2020). Machine learning-based models for real-time traffic flow prediction in vehicular networks. *IEEE Network*, 34(3), 178–185. <https://doi.org/10.1109/MNET.011.1900338>
- Sun, P., Boukerche, A., & Tao, Y. (2020). SSGRU: A novel hybrid stacked GRU- based traffic volume prediction approach in a road network. *Computer Communications*, 160, 502–511. <https://doi.org/10.1016/j.comcom.2020.06.028>
- Sun, S., Wu, H., & Xiang, L. (2020, January). City-wide traffic flow forecasting using a deep convolutional neural network. *Sensors (Switzerland)*, 20(2 421). <https://doi.org/10.3390/s20020421>
- Tampubolon, H., & Hsiung, P. A., “Supervised deep learning based for traffic flow prediction,” in *International Conference on Smart Green Technology in Electrical and Information Systems* 25–27 Oct. 2018 Bali, Indonesia, July. 2018, pp. 95–100, <https://doi.org/10.1109/ICSGTEIS.2018.8709102>.
- Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. *IEEE International Conference on Smart City/ SocialCom/SustainCom (SmartCity)*, 19–21 Dec. 2015,

- Chengdu, China. <https://doi.org/10.1109/SmartCity.2015.63>
- Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5), 307–318. [https://doi.org/10.1016/S0968-090X\(97\)82903-8](https://doi.org/10.1016/S0968-090X(97)82903-8)
- Vijayalakshmi, B., Ramar, K., Jhanjhi, N. Z., Verma, S., Kaliappan, M., Vijayalakshmi, K., Vimal, S. K., & Ghosh, U. (2021). An attention-based deep learning model for traffic flow prediction using spatiotemporal features towards sustainable smart city. *International Journal of Communication Systems*, 34 (e4609). <https://doi.org/10.1002/dac.4609>
- Wang, J., & Boukerche, A., “The scalability analysis of machine learning based models in road traffic flow prediction,” In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 7–11 June 2020, Dublin, Ireland, pp. 1–6, IEEE, 2020, <https://doi.org/10.1109/ICC40277.2020.9148964>
- 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. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(664\)](https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664))
- Williams, B. M. (2001). Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transportation Research Record: Journal of the Transportation Research Board*, 1776(1), 194–200. <https://doi.org/10.3141/1776-25>
- World Health Organization. (2018). “GLOBAL STATUS REPORT ON ROAD SAFETY 2018 SUMMARY.” [On- line]. Available: <https://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf?ua=1>
- Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90(March), 166–180. <https://doi.org/10.1016/j.trc.2018.03.001>
- Xia, F., Yu, C., B, W. Q., & Zhou, A. (2016). A data grouping CNN algorithm for short-term traffic flow forecasting. *APWeb2016*, 1(December 2014), 183–195. <https://doi.org/10.1007/978-3-319-45814-4>
- Xie, P., Li, T., Liu, J., Du, S., Yang, X., & Zhang, J. (2020). Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, 59, 1–12. <https://doi.org/10.1016/j.inffus.2020.01.002>
- Xu, L., Cao, M., Song, B., Zhang, J., Liu, Y., & Alsaadi, F. E. (2018). Open-circuit fault diagnosis of power rectifier using sparse autoencoder based deep neural network. *Neurocomputing*, 311, 1–10. <https://doi.org/10.1016/j.neucom.2018.05.040>
- Yang, B., Sun, S., Li, J., Lin, X., & Tian, Y. (2019). Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*, 332, 320–327. <https://doi.org/10.1016/j.neucom.2018.12.016>
- Yang, D., Li, S., Peng, Z., Wang, P., Wang, J., & Yang, H. (2019). MF-CNN: Traffic flow prediction using convolutional neural network and multi-features fusion. *IEICE Transactions on Information and Systems*, E102.D(8), 1526–1536. <https://doi.org/10.1587/tran-sinf.2018EDP7330>
- Yang, H.F., Dillon, T.S., & Chen, Y.P.P. (2017). Optimized structure of the traffic flow forecasting model with a deep learning approach. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2371–2381. <https://doi.org/10.1109/TNNLS.2016.2574840>
- Yi, H., Heejin, J., & Bae, S. (2017). Deep neural networks for traffic flow prediction, *IEEE International Conference on Big Data and Smart Computing (BigComp)*, 13–16 Feb. 2017 Jeju. <https://doi.org/10.1109/BIGCOMP.2017.7881687>
- Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B., “A comprehensive survey on traffic prediction,” April. 2020, <https://arxiv.org/pdf/2004.08555v1.pdf>
- Yu, G., & Liu, J. (2019). A hybrid prediction approach for road tunnel traffic based on spatial-temporary data fusion. *Applied Intelligence*, 49(4), 1421–1436. <https://doi.org/10.1007/s10489-018-1339-3>
- Zhang, S., Kang, Z., Hong, Z., Zhang, Z., Wang, C., & Li, J. (2018). Traffic flow prediction based on cascaded artificial neural network. *IEEE International Geoscience and Remote Sensing Symposium*, 22–27 July 2018, Valencia, Spain. <https://doi.org/10.1109/IGARSS.2018.8518853>
- Zhang, Y. (2020). Short-term traffic flow prediction methods: A survey. *Journal of Physics. Conference Series*, 1486(5), 052018. <https://doi.org/10.1088/1742-6596/1486/5/052018>, IOP Publishing
- Zhang, Z., & Jiao, X. (2020). A dense inception network with attention mechanism for short-term traffic flow prediction, *Chinese Automation Congress (CAC)*, 6–8 Nov. 2020, Shanghai, China. <https://doi.org/10.1109/CAC51589.2020.9327049>
- Zheng, H., Lin, F., Feng, X., & Chen, Y. (2020). A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 6910–6920. <https://doi.org/10.1109/TITS.2020.2997352>
- Zhou, T., Han, G., Xu, X., Lin, Z., Han, C., Huang, Y., & Qin, J. (2017). δ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting. *Neurocomputing*, 247, 31–38. <https://doi.org/10.1016/j.neucom.2017.03.049>



© 2021 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



***Cogent Engineering* (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.**

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

