



SURVEY ARTICLE

Intelligent Traffic Flow Prediction Using Deep Learning Techniques: A Comparative Study

Sayed A. Sayed¹  · Yasser Abdel-Hamid¹ · Hesham Ahmed Hefny¹

Received: 9 October 2023 / Accepted: 19 November 2024
 © The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd. 2024

Abstract

Congestion in major cities poses a serious threat to long-term city planning and growth. There are several factors that contribute to traffic congestion, including construction zones, adverse weather, special events, and accidents. Recently, with the rise of Artificial Intelligence technology and Intelligent Transportation Systems, new models and frameworks for traffic flow prediction have been rapidly created to improve forecast accuracy, especially in urban areas and overpopulated cities that suffer severe problems with high traffic. This study aims to achieve the following primary objectives: First, it emphasizes the diverse evaluation criteria employed in research to measure the classification results of the Deep Learning models. Second, it presents a comparative study of the most popular Deep Learning techniques adopted in traffic prediction. Third, it provides a brief comparison of the application of the discussed evaluation metrics in the reviewed studies, along with guidelines for selecting the appropriate metrics. Fourth, it compares the presented studies in terms of the methodology followed, the dataset used, and the techniques adopted to gain insights about the pros and cons of each study, as well as the most powerful deep learning techniques used. Lastly, it identifies the inherent obstacles to applying Deep Learning in the domain of traffic prediction. Prominent scientific digital libraries such as Springer, Elsevier, IEEE, and IOP Publishing published the articles under investigation in this study. Articles were retrieved using a variety of search terms, including Deep Learning, traffic flow prediction, short-term traffic prediction, and Intelligent Transportation System. This study focuses on works that are directly related to using Deep Learning methods for traffic flow prediction. In this study, both primary research and reviews were considered. This investigation led to the selection of 30 papers for in-depth evaluation.

Keywords Artificial intelligence · Deep learning · Intelligent transportation systems · Traffic congestion · Traffic prediction

Abbreviations

AI	Artificial intelligence	Bi-LSTM	Bidirectional long short-term memory
ANN	Artificial neural network	BPNN	Back propagation neural network
APIs	Application programming interfaces	BRNN	Bilinear recurrent neural network
AR	Autoregressive	CEC	Constant error carousel
ARIMA	Auto-regressive integrated moving average	CNN	Convolutional neural network
ARMA	Autoregressive moving average	COE	Comprehensive error coefficient
AS	Average speed	COI	Comprehensive optimization index
AST-MTL	Attention-based spatio-temporal multi-task learning	CPM-ConvLSTM:	Congestion propagation model-convolutional long short-term memory
AVI	Automatic vehicle identification	DAN	Deep autoencoder
BA	Bayesian algorithm	D-BLRNN	Dynamic bilinear recurrent neural network
BCM	Bayesian combination model	DBN	Deep believe network
		DELA	Deep and embedded learning approach
		DL	Deep learning
		DNN	Deep neural networks
		ELM	Extreme learning machine

✉ Sayed A. Sayed
 sayed.ahmed@narss.sci.eg

¹ Computer Science Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt

ER	Error rate	SARIMA	Seasonal ARIMA
FCD	Floating car data	SD	Standard deviation
FCNN	Fully connected neural network	SG-CNN	Spatial graph convolutional neural network
FI-RNN	Feature-injected recurrent neural network	SMAPE	Symmetric mean absolute percentage error
FIS	Fuzzy inference system	SMBO	Sequential model-based optimization
FN	False negative	SRCN	Spatiotemporal recurrent convolutional networks
FP	False positive	ST-CGCN	Spatial–temporal complex graph convolution network
FTP	File transfer protocol	SVM	Support vector machine
GARCH	Generalized auto regressive conditional heteroskedasticity	SVR	Support vector regression
GCN	Graph convolution network	TCI	Traffic congestion index
GIS	Geographic information systems	TF	True negative
GPS	Global positioning system	TMC	Turning movement counts
GPU	Graphics processing unit	TOPIS	Seoul transport and information service
Graph-CNN	Graph convolution neural network	TP	True positive
GRU	Gated recurrent unit	TTI	Traffic time indices
HGC-LSTM	High-order graph convolutional long short-term memory network	VAPE	Variance of absolute percentage error
IBCM-DL	Inverse Bayesian combination model-deep learning	WNN	Wavelet neural network
IoT	Internet of Things		
IoVs	Internet of vehicles		
ITS	Intelligent transportation systems		
KDD CUP	Knowledge discovery and data mining tools competition CUP		
KNN	K-Nearest neighbour		
LSTM	Long short-term memory		
MA	Moving average		
MAE	Mean absolute error		
MAPE	Mean absolute percentage error		
MDN	Mixture density network		
MIR	Multilayered information retrieval		
ML	Machine learning		
MLP	Multi-layer perceptron		
MRE	Mean root error		
MSE	Mean square error		
NLP	Natural language processing		
OCR	Optical character recognition		
P	Precision		
PCA	Principal component analysis		
PeMS	Performance measurement system		
PSO	Particle swarm optimization		
PSO	Particle swarm optimization		
R	Correlation coefficient		
R ²	Coefficient of determination		
RCL	Recurrent convolutional layer		
R-CNN	Regional convolutional neural networks		
RF	Random forest		
RMSE	Root mean squared error		
RMSPE	Root mean squared percentage error		
RNN	Recurrent neural network		

Introduction

Developing solutions for precise forecasting and congestion control based on Intelligent Transportation Systems (ITS) is becoming increasingly important as cities continue to grow and become more crowded [1]. ITS aims to integrate communications, data, and computing resources for transportation. This strategy aims to better connect pedestrians, drivers, and motorists [2]. We can use ITS to build a comprehensive, dependable, and efficient transportation management system [3, 4]. We can improve traffic flow, travel speed, dependability, safety, and passenger pleasure, while greatly reducing dangers, high incident rates, traffic congestion, and air pollution [4].

Accurate traffic flow forecasting allows traffic participants (passengers, traffic managers, decision-makers, and road users) to use transport networks safely and intelligently [5, 6]. Only reliable data can achieve true success for an ITS. An estimated 1.35 million persons died in traffic-related incidents in 2016, in accordance with the 2018 World Health Organization's report on global road safety status. As a result, researching traffic forecasting is a good way to reduce congestion and ensure relatively safe, cost-effective transportation [7, 8].

Time series analysis and other parametric models have been used to predict traffic flow. Each x in a time series represents measurements taken at t . To improve traffic

predictions, the goal is to analyze prior traffic data for temporal patterns. Kalman Filtering, a time-series analysis paradigm for mobile stochastic scenarios, can reduce variance and regression difficulties to yield optimal outcomes [9]. The Auto-Regressive Integrated Moving Average (ARIMA) paradigm is popular for short-term traffic flow forecasting [10]. Many ARIMA technique improvements have improved performance [11–14].

Time series analysis in traffic flow prediction can be categorized into linear and non-linear time series models. Network traffic prediction widely uses linear time series techniques because they effectively model the covariance structure within the data. Two popular sub-groups of these models are the Autoregressive (AR) and Moving Average (MA) models, which can be combined to form the Autoregressive Moving Average (ARMA) model. Despite being a traditional approach, linear time series models remain prevalent in network traffic forecasting. We utilize the ARMA model, which integrates both AR and MA components, to enhance the accuracy of network traffic predictions. While effective for well-behaved time series data, ARMA models are less suitable for datasets with random fluctuations. Nevertheless, ARMA continues to be significant in network traffic prediction, with several researchers applying various ARMA-based algorithms [15]. In [16], the authors employed the ARMA model to predict network traffic using File Transfer Protocol (FTP) data. By capturing and analyzing network packets over time, they concluded that transferring smaller files resulted in better network performance and suggested breaking larger files into smaller ones for efficient transfer. In [17], the authors introduced a K-factor ARMA model for predicting high-speed network traffic from multiple sources. They found that their model outperformed traditional AR models when measured using Mean Absolute Error (MAE) and Error Ratio (ER). The ARIMA model extends ARMA by integrating differencing to handle non-stationary data. It plays a significant role in time series modeling for network traffic prediction [15]. In [18], the authors proposed an ARIMA model to predict network traffic using mobile data, achieving high prediction accuracy as measured by the Mean Absolute Percentage Error (MAPE). For better accuracy in predicting public safety trunked radio network traffic, the authors of [19] combined the Seasonal ARIMA (SARIMA) model with a K-means clustering approach. Nonlinear time series models address features that linear models cannot, such as time-varying variance, asymmetric cycles, and thresholds. These models use techniques like neural networks and fuzzy logic to predict network traffic [15]. One type of non-linear time series model is the Generalized Auto Regressive Conditional Heteroskedasticity (GARCH). It was suggested by [20] to show how internet traffic spikes using data from the Abilene backbone network. Their results showed that GARCH outperformed ARIMA

in handling variance fluctuations in network traffic. Neural networks techniques, first introduced as alternatives to statistical methods in network traffic prediction, consist of interconnected neurons with learnable weights. They have been investigated by several researchers such as [21] in which the authors applied a neural network to predict noise in video network traffic, using autocorrelation and Mean Square Error (MSE) to measure performance. Their results showed that neural networks were more effective than traditional linear time series models. Also, the authors in [22] introduced a Dynamic Bilinear Recurrent Neural Network (D-BLRNN) to improve network traffic prediction. By comparing D-BLRNN with the Bilinear Recurrent Neural Network (BRNN), they showed that D-BLRNN was better at making predictions, as measured by MSE. In summary, both linear (ARMA, ARIMA) and nonlinear (GARCH, neural networks) time series models have been extensively used in network traffic prediction, each with its unique strengths and limitations. Linear models are simpler and effective for well-behaved data, while nonlinear models offer a more sophisticated approach to capture complex dynamics within network traffic. Continued exploration of these methods is crucial for advancing the accuracy and robustness of network traffic prediction [15].

K-Nearest Neighbour (KNN), Random Forest (RF), Bayesian Algorithm (BA), Principal Component Analysis (PCA), and Support Vector Machine (SVM) techniques [9] are non-parametric methods that can account for traffic's stochastic and non-linear-nature and are increasingly used to estimate traffic flow. Neural networks have gained widespread use in forecasting traffic quantities recently [23]. Back-Propagation yielded promising results in the large data age [24], but it failed. Deep learning approaches, which use multiple layers to differentiate basic features from complex ones in the input, were created after this failure. Deep learning architectures include Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) Restricted Boltzmann Machines, Deep Belief Networks, and Stacked Auto Encoders.

This paper introduced a comparative study on the deep learning technique's role in traffic prediction to answer the following questions. The first question identifies the most widely used deep learning techniques that can be utilized in traffic flow prediction. The second question is trending the most popular datasets used in traffic prediction studies. The Third question indicates the obstacles and challenges in applying deep learning techniques in traffic prediction.

The primary contributions of this study are as follows: First, it reviews the various evaluation criteria employed in research to assess the performance of deep learning techniques. Second, it offers a comprehensive comparative analysis of the most widely used deep learning techniques in traffic prediction tasks. Third, it offers a brief comparison

of how the discussed evaluation metrics have been applied in the studies reviewed and provides guidelines on how to select the appropriate metrics. Fourth, it compares the reviewed studies based on their methodologies, datasets, and adopted techniques, highlighting the strengths and weaknesses of each to identify the most effective deep learning approaches. Lastly, it identifies the challenges and opportunities associated with applying deep learning in traffic prediction.

The rest of this paper is structured as follows: The theoretical foundations of traffic prediction problems and Deep Learning are discussed in Sect. "[Background](#)". The various evaluation criteria being used in research to measure the classification results of the deep learning models are presented in Sect. "[Popular Evaluation Metrics](#)". The survey methodology and a literature assessment of the most important deep learning techniques used in traffic flow prediction are presented in Sect. "[Survey Methodology](#)". Sect. "[Comparative Analysis of Evaluation Metrics in the Reviewed Studies](#)" presents a comparative analysis of the evaluation metrics used in the reviewed studies. Sect. "[Results and Discussion](#)" discusses the results obtained from this study. Limitations and Challenges that face applying deep learning techniques in traffic flow prediction are discussed in Sect. "[Limitations and Future Directions](#)". Finally, the paper is concluded in Sect. "[Conclusion](#)".

Background

ITS makes it easy to gather extensive traffic data for more accurate traffic flow forecasts [25]. Traffic prediction is a time-series problem that uses data from one or more observation sites to predict flow count. Traffic engineers must be able to predict traffic trends. By predicting traffic patterns in real-time, authorities can improve traffic control and advise road users to improve their travel options and save money. Machine learning and deep learning, Artificial Intelligence (AI) subcategories, have grown exponentially [26].

Deep Learning

Deep Learning (DL) became a powerful machine learning technique around the beginning of the century, and it has since shown promising results in a variety of practical settings. The basis of DL methods is the use of Deep Neural Networks (DNN) to learn complex properties retrieved from data with minimal external influence [27]. These algorithms can learn new, more complex features on their own, eliminating the need for handcrafted features [28]. DNNs have become increasingly popular due to their flexibility, ability to model non-linear relationships, and their data-centric approach to model development [29, 30].

The academic community is very interested in DL since it is a new AI paradigm with greater promise than traditional approaches [31]. When compared to machine learning, DL is the superior method in terms of efficiency, monitoring, time, and cost. Although it's a specialized approach, its adaptability to various approaches and contexts can aid in resolving a variety of complex problems. In a reasonably open-ended fashion, the method taught illustrative and differential properties [32, 33]. DL is built on a set of machine learning approaches for data modeling, aiming to construct high-level abstractions through multiple nonlinear transformations. With DL technology, the Artificial Neural Network (ANN) system can function [34, 35]. Because DL networks contain more layers for acquiring high-level properties and eliminating undesired data, they outperformed machine learning algorithms [36].

Effective web search, computer vision, autonomous vehicles, and Optical Character Recognition (OCR) are just some examples of how machine learning techniques have changed our daily lives. Improvements in human-level AI have also been made possible using machine learning techniques [37–39]. However, using traditional machine learning algorithms to process human information such as voice and vision yields subpar results. By the end of the twentieth century, the idea of DL methodologies emerged due to the complexity of understanding and making speeches for people. Common DL techniques include CNNs, RNNs, LSTM networks, and Regional Convolutional Neural Networks (R-CNN). CNNs are highly effective in visual tasks because they take advantage of the spatial characteristics inherent in images and videos. In contrast, RNNs are adept at capturing temporal correlations within data, making them particularly well-suited for time series analysis. A notable variant of RNNs, LSTM network, is especially proficient at learning order dependencies in sequence prediction problems [40].

Convolutional Neural Network

In image processing, CNNs commonly assign significance to various features within an image to distinguish between different elements. However, other contexts can also apply CNNs, as long as the input data follows the structure of an image [41].

Text analysis, biological picture recognition, face recognition, and human organ localization are just a few of the many applications of CNNs [42]. In 1988, Fukushima originally presented the CNN structure [43]. However, the limited availability of computing resources prevented its widespread use. In [44], the authors were able to solve the problem of classifying handwritten numbers after they applied a gradient-based learning technique to CNNs. Next, experts improved CNNs incrementally and shared their best work across a variety of recognition tasks.

Among their many benefits, CNNs feature a structure that is well-tuned for processing 2D and 3D images, and they are efficient learners and extractors of abstract 2D information, making them more like the human visual processing system. CNNs' max-pooling layer excels in reducing the effects of shape fluctuations. In addition, because CNNs use sparse connections with linked weights, they can function with many fewer parameters than a fully connected network of the same size. CNN training employs a gradient-based learning strategy that is more resilient to the diminishing gradient problem. The gradient-based training strategy, which optimizes the network in terms of a single error criterion, enables CNNs to generate very efficient weights [45].

In most research, prediction models are often impacted by shifts caused by various complex factors. The inherent characteristics of traffic flow, along with external influences that directly affect observed values, are not fully accounted for. From a temporal perspective, traffic flow demonstrates not only continuity over time but also exhibits daily and weekly periodic patterns. Understanding these periodicities is crucial for gaining insights into traffic behaviours and identifying travel patterns. Spatially, the traffic flow on a particular road is influenced not only by its neighbouring roads but also by distant roads, as suggested by propagation theory. Additionally, adverse weather conditions and holidays play a significant role in increasing travel time and crash risk, thereby affecting traffic flow. Without a comprehensive understanding of these diverse traffic features, prediction models will continue to struggle with shifts induced by complex factors. Therefore, incorporating more realistic conditions to enhance the accuracy and robustness of traffic flow models remains an important area for future investigation [46].

Recurrent Neural Network

The 1980s saw the creation of the RNN DL model, which is now extremely popular. RNNs are a widely used approach for addressing time-series problems, making them a common choice for predicting traffic flow by capturing temporal dependencies. RNNs are a specific type of neural network that utilize hidden states, allowing previous outputs to be fed back as inputs. However, the gradient vanishing or exploding problem, a challenge inherent to the original RNN architecture, limits their ability to capture long-term dependencies in sequential data. RNNs have demonstrated relatively accurate short-term traffic predictions due to their shared weights and recurrent architecture [47].

There are numerous varieties of these networks, each with its own memory for storing the data it has accumulated thus far [48]. RNNs are also reliable designs for time series analysis due to their ability to extrapolate results from the past. In this setup, the networks' hidden layers consist of

recursive loops that temporarily store input data and utilize it to predict subsequent outputs. After the hidden layer's output has been processed, it is sent back to the hidden layer t times. The recursive layer only passes on its output to the subsequent layer after reaching the specified number of iterations. This results in a wider distribution of results and a longer preservation of acquired wisdom. We then update the weights based on the re-tuned mistakes [49]. Speech processing and Natural Language Processing (NLP) are two common applications for RNN [50, 51].

In contrast to CNN, RNN uses sequential data in its neural network. This trait is crucial to many different applications, including NLP, because the underlying structure in the data sequence provides useful information. We can view RNN as a unit of short-term memory, with x serving as the input layer, y as the output layer, and s as the state (hidden) layer [52]. Furthermore, in [53], the authors proposed a deep RNN to reduce learning difficulty in deep networks while still providing the advantages of a deeper RNN. This deep RNN makes use of three distinct deep RNN techniques: the "Hidden-to-Hidden," "Hidden-to-Output," and "Input-to-Hidden" architectures [54].

Long Short-Term Memory

The LSTM, an enhanced version of the RNN, addresses the limitations in managing long-range dependencies. LSTM balances short-term and long-term memory well by using complex gate mechanisms [47]. This makes the gradient vanishing and exploding problems less severe. LSTM effectively captures the temporal and spatial correlation of the traffic network, leading to improved performance [55].

Since it shares the RNN's memory and multiplicative gates, LSTM is a subclass of RNN. The authors detailed the various cell configurations of the first LSTM in [56]. Popular services such as Siri, Alexa, Cortana, Google Voice Assistant, and Google Translate all owe something to LSTM [57]. LSTM is a component of RNN that can handle missing gradient problems. In most cases, RNN uses an LSTM network to prevent error reproduction. As a result, the RNN can acquire knowledge across different time intervals. Some of the cells in an LSTM are memory cells, which store data separately from any recurrent connections. The cell's gate functions similarly to a computer's memory, determining when to store, write, read, or wipe information. To shield LSTM from the flaws of standard RNN, a memory block includes a self-connected linear unit called a Constant Error Carousel (CEC), which is a major component. Weight matrices and activation functions are what make up an input gate and an output gate, respectively [58].

The LSTM cell consists of an input layer, an output layer, and a hidden layer that self-connects to both the input and the output levels. It's possible that the 'conventional' units

that feed the subsequent LSTM cells are in the hidden layer. The capabilities of a standard LSTM cell were restricted. It was noted that the function's hand-converted ordinal unit could reach saturation with its gradual expansion. As a result, they included a forget gate layer to allow unwanted data to be erased and forgotten [59]. The most advanced network topologies, including bidirectional, hierarchical, convolutional, grid, autoencoder, and cross-modal LSTM, use LSTM gating [60].

Regional Convolution Neural Network

In response to the discovery of rich recurrent connections in animal visual systems, researchers developed R-CNNs, a new type of CNN. R-CNN relies on a special layer called the Recurrent Convolutional Layer (RCL) to integrate cross-neuron recurrent connections. Because RCL's neurons' receptive fields expand infinitely in size as the number of recurrent computations increases, it is not compatible with biological facts [61]. The standard R-CNN model first appeared in [62, 63]. There is local connectivity between feed-forward and recurrent connections and shared weights between nodes in the R-CNN architecture, as described in [62].

From a computational standpoint, R-CNN offers several benefits. First, R-CNN's recurrent connections permit each unit to incorporate context information from an arbitrary broad region of the current layer. Second, the recurrent connections increase the network's depth and help maintain a constant number of tuneable parameters using shared weights. This conforms to the standard practice of modern CNN design. Thirdly, an unfolded R-CNN is a CNN that has several possible routes from the input layer to the output layer, making it easier to train. One positive aspect of longer routes is that they allow the model to pick up more nuanced details. However, training gradient back-propagation may benefit from shorter pathways [62].

Popular Evaluation Metrics

Evaluation metrics refer to a collection of statistical measures utilized to assess and ascertain the efficacy and sufficiency of a binary, multi-class, or multi-labelled classifier in connection to the classification data being modelled. We employ the assessment metric to quantitatively assess and consolidate the effectiveness of the trained classifier using previously unseen data. The outcomes of these evaluation measures will ascertain if the classifier has achieved optimal performance or necessitates further refining [64]. This section delves into the widely used evaluation metrics that gauge the effectiveness of machine learning models.

Correlation Coefficient

We commonly use the correlation coefficient (R) to evaluate the quality of linear regression models in ANN. In this study, the symbol "R" denotes the Pearson correlation coefficient. The function assesses the level of correlation between the observed and estimated variables. Equation (1) illustrates the computation of R. The range of values for the magnitude of R extends from -1 to $+1$. The sign of the R-value can be positive (+) to denote a direct correlation or negative (−) to denote an inverse correlation. Five degrees categorize the strength of correlation between two variables: perfect correlation (1), very high correlation (0.8 to 1), strong correlation (0.6 to 0.8), moderate correlation (0.4 to 0.6), weak correlation (0.2 to 0.4), and very weak correlation (0 to 0.2). Equation (1) [65] demonstrates the computation of R.

$$R = \frac{\frac{1}{n} \sum_n^1 (D_{act} - \bar{D}_{act})(D_{pre} - \bar{D}_{pre})}{\sqrt{\frac{1}{n} \sum_n^1 (D_{act} - \bar{D}_{act})^2} \sqrt{\frac{1}{n} \sum_n^1 (D_{pre} - \bar{D}_{pre})^2}} \quad (1)$$

where D_{act} is the actual variable, D_{pre} is the predicted variable, $\bar{D}_{act}$ is the mean value of the actual variable, $\bar{D}_{pre}$ is the mean value of the predicted variable and n is the amount of collected data.

Coefficient of Determination

ANN employs the Coefficient of Determination (R^2) as another metric to evaluate the quality of linear regression models. A regression model can account for the proportion of the predicted variable [66]. Alternatively put, it signifies the percentage of the dependent variable that the independent variables can explain, considering the total variability of the dependent variable. We calculate the coefficient of determination (R^2) as the square of the correlation between the real and the estimated variables [67]. Therefore, the variable R^2 exhibits a range of values between 0 and 1. A coefficient of determination of 0 signifies that the regression model does not account for any variation in the dependent variable, indicating a lack of connection between the two variables. A value of 1 signifies that the regression model fully accounts for the variability in the predicted variables, indicating a perfect correlation between the two variables [65].

Mean Squared Error

Scale-dependent metrics are those based on absolute errors or squared errors. The scale of the scale-dependent metrics is the same as that of the raw data [68]. Scale-dependent metrics include, for example, the MSE. MSE compares the

expected value to the actual value. The vertical distance between each data point's actual value and the matching predicted value on the fit line is measured, and the value is squared. The total squared values are then added together and divided by the number of points. As a result, the square of the original unit serves as the MSE unit. The squared errors prevent the positive and negative numbers from cancelling each other out [65]. The MSE has a range of 0 to $+\infty$; the lower the MSE value, the more accurate the prediction model is. MSE has a perfect value of 0, which signifies that the prediction model is accurate. In DL, MSE is the loss function for linear regression by default [65]. MSE can be calculated according to Eq. (2) [69].

$$MSE = \frac{1}{n} \sum_{i=1}^2 (D_{pre} - D_{act})^2 \quad (2)$$

Root Mean Squared Error

Root Mean Squared Error (RMSE) is another example of Scale-Dependent Metrics which measures the average magnitude of error between the predicted value and actual value. As a result, the RMSE is the average vertical distance measured from the actual value to the predicted value on the fit line. Simply, RMSE is the square root of MSE and in the same manner, the range of RMSE is $(0, +\infty)$. It is worth noting that a lower RMSE value corresponds to a higher level of accuracy in the prediction model. In contrast to MSE, RMSE retains the same units as the original data, hence enhancing its interpretability compared to MSE [65]. Equation (3) can compute RMSE [69].

$$RMSE = \sqrt{MSE} \quad (3)$$

Mean Absolute Error

The MAE is another example of Scale-Dependent Metrics which measures the average magnitude of error between the predicted value and actual value [65]. The range of MAE extends from zero to positive infinity. A lower MAE value indicates a higher level of accuracy in the prediction model. The unit of MAE is consistent with that of the original data [65]. The calculation and understanding of this function are straightforward, rendering it a commonly employed symmetrical loss function (Flores, 1986). MAE can be computed using Eq. (4) [69].

$$MAE = \frac{1}{n} \sum_{i=1}^n |D_{pre} - D_{act}| \quad (4)$$

Mean Absolute Percentage Error

The most popular Percentage-Dependent Metric is MAPE. The Percentage-Dependent Metrics measure the size of errors in terms of percentages and give us a way to think about how good the predictions are. The MAPE figures out what the average percentage error is. It is a way to measure how accurate a prediction is, especially when trying to figure out a trend. Regression models in ML use MAPE as the loss function because it simplifies the understanding of the relative error. The range of the MAPE is $(0\%, +\infty)$. A lower value of MAPE indicates a higher level of accuracy in the prediction model. A MAPE value of 0% signifies that the prediction model is accurate. If MAPE exceeds 100%, it signifies that the predictive model is of notably low quality. MAPE has a weakness in that it is sensitive to scale. If the real value is small, the MAPE will give you extreme values. Therefore, you shouldn't use the MAPE to evaluate small amounts of data [65]. Equation (5) is used to calculate MAPE [70].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{D_{pre} - D_{act}}{D_{act}} \right| \quad (5)$$

Symmetric Mean Absolute Percentage Error

Another popular Percentage-Dependent Metric is the Symmetric Mean Absolute Percentage Error (SMAPE). It is the optimized version of MAPE. Both positive and negative errors incur the same penalty. However, if the actual value is zero and the prediction value is also zero or near to zero, SMAPE will likely have a value of infinity. The range of the SMAPE is $(0\%, +\infty)$. A lower value of SMAPE indicates a higher level of accuracy in the prediction model. A MAPE value of 0% signifies that the prediction model is accurate. If SMAPE exceeds 100%, it signifies that the predictive model is of notably low quality [65]. Equation (6) is used to calculate SMAPE [70].

$$SMAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{D_{pre} - D_{act}}{(|D_{pre}| + |D_{act}|)/2} \right| \quad (6)$$

Confusion Matrix

The confusion matrix is one of the easiest and most intuitive metrics used for finding the precision of a classification model. In 2×2 confusion matrix, there are four categories: True Positive (TP), False Positive (FP), True Negative (TF), and False Negative (FN) [64]. The confusion matrix is a fundamental concept in ML, providing a comprehensive

representation of the actual and expected classifications made by a classification system. The confusion matrix is a two-dimensional matrix in which one dimension is associated with the actual class of an object, while the other dimension is associated with the class predicted by the classifier [71].

Precision

Precision (P) is a metric that quantifies the proportion of accurate predictions made by a model, especially in cases where the model predicts true positives. The computation of the metric is derived by dividing the number of true positives by the sum of true positives and false positives, as represented by Eq. (7) [64].

$$P = \frac{TP}{TP + FP} \quad (7)$$

Recall

Recall quantifies the model's capacity to accurately predict true positives across all accessible categories and assesses the model's effectiveness in making these predictions. The formal calculation of this metric is expressed as the division of true positives by the sum of true positives and false negatives, as depicted in Eq. (8) [64].

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

Accuracy

Accuracy (A) is commonly described as the proportion of accurate predictions made for the given test data, expressed as a percentage. The formal calculation of this metric involves determining the ratio of true positives and true negatives, and then summing this value with the total number of forecasts, as represented by Eq. (9) [64].

$$A = \frac{TP + TN}{TP + FP + TN + FN} \quad (9)$$

F1—Score

F1—Score is a function of Recall and Precision. It can be used to measure the efficiency of the unbalanced dataset. F1—Score can be computed as shown in Eq. (10) [64].

$$F1 - Score = \frac{P * Recall}{P + Recall} \quad (10)$$

Error Rate

The Error Rate (ER) is a metric that quantifies the proportion of inaccurate predictions in relation to the overall number of cases assessed. The equations representing the calculation of ER can be shown in Eq. (11) [64].

$$ER = \frac{FP + FN}{TP + FP + TN + FN} \quad (11)$$

Survey Methodology

Prominent scientific digital libraries such as Springer, Elsevier, IEEE, and IOP Publishing publish the articles investigated in this review. Articles are retrieved using a variety of search terms, including those related to Deep Learning, traffic flow prediction, short-term traffic prediction, and intelligent transportation systems. We begin the first search round by choosing the fields to search within. These fields are the Article's Title, Abstract, Keywords, Methodology, and Conclusion. Afterwards, we applied the following selection criteria to identify the desired papers.

- Articles published in journals.
- Articles published in conference proceedings.

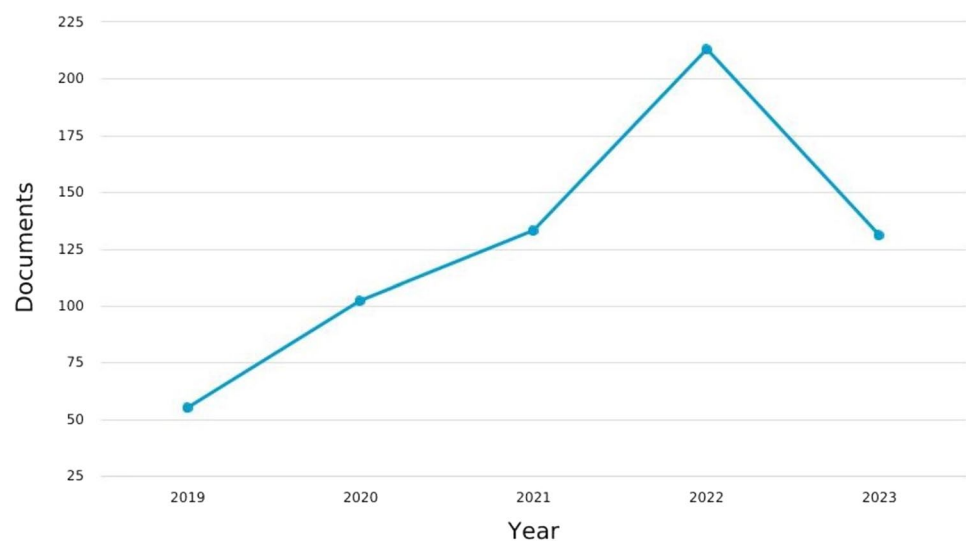
As a result of the previous selection criteria, a total of 634 papers covering the period from 2019 to 2023 were selected, as depicted in Table 1 and Fig. 1.

Then, search round 2 begins by applying a second selection procedure on the 634 papers obtained in round 1. After applying the second selection criteria, the authors selected only 30 papers to be thoroughly investigated. Table 2 illustrates how these papers were finally chosen after applying the second selection criteria.

This literature review focuses on works that are directly related to using DL methods for traffic flow prediction. For this effort, both primary research and reviews on the topics were considered.

Table 1 The number of documents published per year

Year	# of Documents
2019	55
2020	102
2021	133
2022	213
2023	131

Fig. 1 Number of documents per year**Table 2** The number of documents published per year

Process	Selection criteria	Total
Round 1 Search	Journal articles	341
	Conferences articles	634
Round 2 Search	Articles written in English only	380
	Articles covered the search terms explicitly	210
	Full-TEXT ARTICLES	130
	Articles directly related to the topic of this study	86
	Not duplicated papers	30

Deep Learning Techniques for Traffic Flow Prediction

This survey compares several traffic flow forecasting methods. It examines the most used DL approaches for predicting traffic flow. This paper presents a comprehensive analysis of the approaches and algorithms used for forecasts, performance evaluations, and the instruments employed for these processes.

Big Data and Machine Learning for Traffic Prediction

One of the most important jobs in ITS nowadays is traffic flow forecasting [72]. Recent years have seen a rise in the use of statistical approaches, artificial intelligence, and data mining techniques for analyzing traffic data and forecasting future indicators [73]. The findings obtained ensured that no single technique could evaluate huge datasets on its own. This means that the right technology must be used, depending on the data structure and its volume, to get the most useful information out of the data that has been gathered [74].

An innovative and comprehensive approach to traffic forecasting at scale, speed, and in real-time was proposed

in [73]. The four auxiliary cutting-edge technologies that have been integrated are DL (CNN and LSTM), big data, Graphics Processing Units (GPUs), and in-memory computing. When training deep networks, the authors used information gathered by the California Department of Transportation (Caltrans) [75] over the course of 11 years. The proposed model was evaluated using MAPE. The MAPE speed forecast parameter was 2.59, whereas it was 5.96 for the flow forecast parameter. The proposed method used a limited data set and had low prediction accuracy.

The LSTM model and a novel approach to traffic prediction with low prediction errors were both created using DL in [76]. The large volume of traffic data collected by the Performance Measurement System (PeMS) served as the basis for this study. The proposed model was evaluated using the Mean Root Error (MRE) and RMSE. To achieve better prediction results, the authors used the Multilayered Information Retrieval (MIR) method to optimize three parameters of the LSTM prediction model: Split window length L , state vector S_{state} , and learning rate η . The MIR LSTM parameter optimization approach aids in the selection of the most appropriate parameter combinations, with a minimum MRE and RMSE of 0.984 and 18.037, respectively. The number of fine-tuned variables in this analysis needs to be expanded. Moreover, it is necessary to determine a time frame for model training.

A pathway-based Bidirectional LSTM (Bi-LSTM) DL framework was suggested in [77]. It improves citywide traffic projections. Most importantly, the paradigm was rational and easy to apply to urban mobility. The assessed road network had 112 roadways. Automatic Vehicle Identification (AVI) detectors in central Xuancheng, China, provided the dataset. The experimental findings revealed that the proposed model predicted best. KNN, ANN, CNN,

and LSTM reduce MSE by 35.40%, 21.09%, 9.92%, and 16.90%, respectively, in all prediction situations.

Deep Learning Techniques for Traffic Prediction

Metropolitan road network traffic flow patterns are unpredictable due to periodicity and short-term unpredictability. Thus, in [78], the authors built a Particle Swarm Optimization (PSO) Bi-LSTM model to estimate short-term traffic flow utilizing PSO and Bi-LSTM neural networks. Instead of linear weights, PSO discovers the ideal model parameters globally using nonlinear variable inertial weights. PSO optimizes the Bi-LSTM network prediction model for fast convergence, resilience, and global search. To test the model, Inner Ring Road, South Extension, Delhi, India, traffic flow data from 24–02–2020 to 28–02–2020 was collected. PSO-Bi-LSTM was compared against Bi-LSTM, LSTM, Extreme Learning Machine (ELM), Gated Recurrent Unit (GRU), Wavelet Neural Network (WNN), Multi-Layer Perceptron (MLP), and ARIMA. PSO-Bi-LSTM outperformed others in accuracy and stability in experiments. The suggested model was assessed using RMSE, MAE, MAPE%, R, and Explained Variation. The results were 16.0867, 6.3450, 1.6162, 0.9995, and 0.9862.

For the objective of making accurate long-term forecasts of traffic volumes, many DL models were developed in [79]. The best predictions may be made by using two hidden LSTM layers, one with 64 neurons and another with 32 neurons, as demonstrated by the simulation results. Over 93% of the predictions had an error of less than $\pm 2.0\%$. Peaks in some exceptional circumstances are the primary drivers of variance analysis. Two data sources, electromagnetic loops, and cameras were combined for this purpose. The accuracy of the sources used in data fusion is calibrated depending on environmental factors like lighting and daylight hours. After the data fusion process, the integration results were compared with the actual data to show that prediction outcomes had improved at peak periods.

In [80], the authors used Python to analyze uphill traffic speeds. Therefore, a centralized analysis technique was presented employing central Kunming City, China, traffic data. The study used data from an online map (<https://pro.arcgis.com/en/pro-app/toolreference/spatial-analyst/how-kernel-density-works.htm>). The average data acquisition time is 15 min. After mistakes were removed, 45,508,944 road traffic data points were retrieved from May 6–18, 2019. A PSO-LSTM model predicted road traffic speed trends. After that, the local and static heat map describes rush hour. Finally, the writers examined Kunming's main road traffic and plateau mountain features. The quantitative analysis employed the Comprehensive Error Coefficient (COE), R^2 , and the Comprehensive Optimization Index (COI). Using

these measures, the proposed model was trained at 1.192, 0.165, and 1.027 and predicted at 0.876, 0.462, and 0.414.

In [81], four different DL methods—Deep Autoencoder (DAN), DBN, Recurrent Fuzzy, and LSTM—were proposed for use in building a traffic prediction system. This method is typically employed to predict the flow of traffic in densely populated areas. This analysis relied heavily on variables such as region, season, time of day, road capacity, and vehicle type. The performance of the proposed model can be measured by using the accuracy, precision, recall, and error value metrics of ML approaches. Of the four approaches, LSTM generated 95.2% accuracy. The dataset employed is not mentioned in this paper.

In [82], the primary goal was to use neural networks to estimate how long it would take to get from point A to point B along a given path. The journey time was estimated using several DL and neural network algorithms, including the K-Means algorithm in conjunction with a few other parameters. This research made use of a dataset collected with the help of the Waze Live Map Application Programming Interfaces (APIs). To improve the efficacy and dependability of their work, the authors should consider aspects like the weather.

In [83], it was suggested that a recurrent mixture density network, which is a mix of an RNN and a Mixture Density Network (MDN), could be used to predict short-term traffic flow. In this analysis, the authors used data collected by sensors installed throughout road networks in Shenzhen, China, to track the movement of traffic. It covered the first three months of 2019 (January through March) and the second six months (October through December) of 2019. A major problem with this investigation is the small size of the data set used. In 10 min prediction, the recurrent MDN improves the accuracy of the Traffic Time Indices (TTI) prediction and Average Speed (AS) prediction by 1.28% and 0.8%, respectively.

In [84], the authors improved the DBN, a DL approach, to provide reliable traffic forecasts even in severe weather. The Internet of Vehicles (IoVs) were initially employed to capture data during severe weather and traffic bottlenecks instead of inductance coils. The Support Vector Regression (SVR) technique was then applied to improve the standard DBN. The improved DBN has two layers: a regular DBN for unsupervised learning of traffic data basic features and an SVR for supervised traffic forecasting. The authors used two data sets in this investigation. First, adjacent weather stations provide weather data, then a highway control centre provides traffic data. The experimental findings revealed that the enhanced DBN controlled traffic prediction error within 9% and was robust despite the extended time gap. The main problem identified by this investigation is optimizing the upgraded DBN's computing time.

Combining efficient scheduling methods for traffic lights with traffic prediction methods is the basis for the urban traffic light control system proposed in [85]. This paper used Back Propagation Neural Network (BPNN) to predict traffic flow. The aim was to lessen the number of cars waiting at the many signalized intersections along the road system. An urban traffic control system framework was first suggested, with features like traffic flow forecasting and optimized signal control. Second, an interactive traffic light method was implemented to help with traffic flow. To ensure the effectiveness of the proposed approach, experiments were conducted using real-world traffic data from the Aliyun Tianchi platform. Both the proposed system and the signal control optimization method performed well in the comparison.

In [86], the authors aimed to forecast real-time queue wait times using DL. They contemplated installing automobile detectors at the crossing and transferring data to the next intersections through a corridor. The target intersection's and two upstream intersections' lines estimated the following cycle's crossing point queue length. The adaptive traffic control system InSync was used to train an LSTM neural network model to detect signal queuing patterns. Sequential Model-Based Optimization (SMBO) selected the best dropout from many stacking layers. This prevented overfitting and optimized hyperparameter selection. InSync adaptive traffic light data from December 18, 2017, to February 14, 2018, is used in this study. From East Orlando, Florida, Lake Waterford offered Alafaya Trail (SR-434) information. McCulloch Road crosses 11 roads. The InSync database stores the Turning Movement Counts (TMC), stage time, length, line, and waiting time. Lack of data (30–60 s) prohibited precise vehicle motion data. The authors used RMSE and MAE to assess model accuracy. Most RMSE and MAE are below 1. Accessing this data may improve model performance.

Attention-Based Spatio-Temporal Multi-Task Learning (AST-MTL), a multitasking learning model, predicts multi-horizon traffic flow and speed on a road network as proposed in [87]. Multi-headed attention and a Fully Connected Neural Network (FCNN) learn related tasks and increase generalization performance. The model gets Spatio-Temporal traffic data from the Graph Convolution Network (GCN) and GRU. FCNN compares many standard representation functions first. The attention mechanism uses task-specific and task-shared representations to find important information and enhance model prediction. A new Global Positioning

System (GPS) was utilized to estimate highway and city traffic in the testing. Table 3 shows the scale-dependent RMSE, MAE, and MASE results for the competing forecasting techniques.

The Feature-Injected Recurrent Neural Network (FI-RNN) blend temporal-sequential data with contextual attributes to assess if traffic context and state are related, as proposed in [88]. This model examined traffic data with a stacked RNN. Meanwhile, a sparse automatic encoder was trained to highly improve contextual element encoding and representation. Researchers created fusion characteristics by adding context to sequence data. Finally, the forecaster incorporates built-in traffic trend and speed prediction features. More feature extraction and merging procedures should be studied to improve the proposed model's accuracy and performance. Other aspects must be considered.

The authors in [89] devised an array-based traffic situational awareness method using SVR and LSTM. Applying graph convolution to a network of detectors revealed traffic flow spatial patterns. The authors then created a weight matrix from the returning features to average the underlying models' predictions based on test results. This study used Caltrans PeMS traffic data. Five statistical metrics—MAE, MRE, RMSE, and Standard Deviation (SD) of MAE and MRE—were utilized to evaluate the suggested traffic forecast approach. The measurements yielded findings of 12.9501, 0.1049, 17.3852, ± 2.6812 , and ± 0.0249 , respectively. Enhancing network structure and parameter choices was the main finding of this research.

In [90], the authors employed a DL model to predict the traffic of individual neighbourhoods was proposed. In this model, LSTM and Graph-CNN were the backbones. One way to approximate traffic density is to compare the rate at which cars pile up in a certain area to the average speed at which people make their way through it on a given day. In this study, the authors used a condensed version of the Highway Network in the San Francisco Bay Area.

In [91], an Inverse Bayesian Combination DL Model (IBCM-DL) was created to overcome the error amplification problem of typical summing techniques and enhance traffic flow prediction. It combines DL techniques like GRU and LSTM with Bayesian Combination Model (BCM). The modified model uses Wang's BCM architecture [92]. This study uses real-world traffic data from microwave sensors deployed on Beijing's roads. This article measured traffic forecasting performance using MAE, MAPE%, and VAPE%,

Table 3 Performance of the proposed model in [87] for different metrics

Experiment	Flow			Speed		
	RMSE	MAE	MASE	RMSE	MAE	MASE
Belgian freeway system	5.37	3.37	0.71	16.05	8.59	0.80
Bruxelles region urban network	0.76	0.40	0.67	8.90	3.88	0.62

which were 7.41%, 12.44%, and 13.94%. Weather, traffic, accident, speed, and occupancy data could increase the model's accuracy.

In [93], the authors explored urban traffic projections using Floating Car Data (FCD). Four DL approaches can predict urban traffic with CNN and RNN, according to the authors. The authors examined High-Order Graph Convolutional Long Short-Term Memory Network (HGC-LSTM), LSTM, GRU, RNN, and Spatiotemporal Recurrent Convolutional Networks (SRCN) models. Traffic simulation creates FCD inputs in proposed ways. The first FCD was created using micro-traffic simulator Aimsun (2018). Periodically logged simulated vehicle data includes identification number, speed, section, and lane. Setup took 10 s. This study examined prediction models using Camp Nou and Amara data from Barcelona. LSTM and GRU were better for maximum (100% and 75%) and lowest (50% and 25%) penetration in Amara. Camp Nou's HGC-LSTM model has the lowest penetration. Lower error rates make LSTM and GRU better than convolutional algorithms (SRCN and HGC-LSTM).

CNN and Deep ANN traffic prediction models were recommended by the authors in [94] to predict traffic speeds on upstream roadway segments such as connecting freeways. The proposed models can identify upstream mainline and downstream connecting line bottlenecks. The suggested models forecast work zone traffic velocity using baseline traffic volume, baseline traffic speed, work area capacity, distance from the work area, vertical gradient of the road, downstream traffic volume, and highway section type. The suggested models use dropout control to prevent ANN overfitting. Finding additional sources to routinely update traffic volume to reflect traffic flow would improve CNN model accuracy. Additionally, a simulation model can help the new CNN model predict office space capacity. When warehouses automate databases, it may be easier to find new things and upgrades. If sufficient data is provided, the model may forecast opposite-direction delays.

In [95], the authors took a snapshot of the Seoul Transport and Information Service (TOPIS), a free and publicly available web-based traffic congestion mapping service and proposed (1) an efficient and city-wide data acquisition mechanism and (2) a blended neural network architecture to retrieve spatio-temporal information from the input image and predict bottlenecks. The suggested design includes an LSTM network between the convolutional encoder and decoder. The convolutional encoder encodes the visual stream into low-resolution latent states. After learning to express time series as a sequence of events, a convolutional decoder converts the LSTM network's latent state, restoring its accuracy. Weather reports—snow, rain, and fog—should be considered to improve prediction accuracy. Additionally, the proposed paradigm must be more effective. Estimates are more accurate with more data from more sources.

By filling time and space gaps, the authors in [96] suggested an LSTM-based traffic congestion prediction approach. The suggested method pre-processes traffic data by deleting unnecessary details using the average absolute deviation and adjusting Spatio-temporal values based on temporal and geographical trends and patterns before predicting. The prediction approach trained time-series data with the LSTM model. This fixed the issue that time-series data don't learn as effectively. This study collected data in Gangnam-gu, Seoul, in November 2018. The proposed method's MAPE was about 5% to evaluate its performance. The proposed approach should help predict traffic congestion in slow-moving areas and big cities. Also, authors should create a model to improve users' jobs.

Fuzzy Inference System (FIS) and CNN (F-CNN) were used by [97] to improve traffic forecasts. To account for CNN's uncertainty when first entering traffic accident features, this technique uses a fuzzy representation. The region was divided into 32×32 square blocks, and three direction sequences were created for each type of incoming and exiting traffic to extract the spatio-temporal data. Second, a fuzzy inference system inferred traffic accident data from traffic patterns. The F-CNN model can train with trend sequence, unconfirmed traffic accidents, and external data. To learn F-CNN parameters, effective pre-training and tweaking techniques were created. The Beijing taxi real route and meteorological datasets were used to verify that the proposed strategy outperforms current methods. RMSE, MSE, and MAE assessed the suggested model. Table 4 shows metric findings. Traffic flow prediction parameters and better DL models need more investigation.

Short-term traffic forecasting was proposed in [98]. The GRU and spatial-temporal analyses are also included. The suggested model used temporal and geographic correlation tests on traffic flow data to identify the input time frame and geographical data size for spatiotemporal feature selection. The raw traffic data is transformed into a two-dimensional matrix that incorporates traffic spatio-temporal flow information. The GRU examined the spatial and temporal properties of the internal traffic flow matrix to complete the forecasting procedure. The suggested model was assessed using RMSE, MAE, and MAPE%. Table 5 shows metric findings.

Table 4 Performance of the proposed model in [97] for different metrics

Model	RMSE	MSE	MAE
F-CNN (5 quantities FIS)	17.59	309.60	9.79
F-CNN (7 quantities FIS)	17.53	307.56	9.93
F-CNN (9 quantities FIS)	17.14	293.91	9.96

Table 5 Performance of the proposed model in [98] for different traffic flow periods

Period	MAE	RMSE	MAPE%
5 min	7.05	9.59	17.8
10 min	7.81	10.84	19.52
15 min	9.27	12.57	21.33
20 min	10.03	13.28	21.99

There are issues with this method, such as the fact that just traffic on a given stretch of road is predicted and other factors (such as weather) are ignored.

Spatial–Temporal Modeling for Traffic Flow Prediction

In [99], the authors proposed a Spatial–Temporal Complex Graph Convolution Network (ST-CGCN) based traffic flow prediction system. Distance, data correlation, and comfort measurement matrices were created using geographic locations, historical data, and traffic node interference. These three matrices were fused into a complex correlation matrix utilizing self-learning dynamic weights to improve Spatial–Temporal feature and external factor modeling. Next, spatial and temporal feature extraction modules described dynamic Spatial–Temporal characteristics. A graph convolution operator, complex correlation matrix, and residual unit make up the spatial feature extraction module. The temporal feature extraction module uses 3D convolution and LSTM. This article used four PeMS datasets and the SZ-taxi dataset from Luohu District, Shenzhen, China cabs. This study objectively investigated model prediction effects using RMSE, Root Mean Squared Percentage Error (RMSPE), MAE, and MAPE. The suggested model improved prediction accuracy by 2.839%, 2.448%, 2.321%, and 4.932% using RMSE, RMSPE, MAE, and MAPE.

Traffic flow forecasting is a hot topic in transportation research. Traffic projections are harder due to the exponential expansion of traffic data. In [100], the authors developed a CNN and Bi-LSTM multi-step prediction technique to solve this problem. Traffic time series features were extracted by the BI-LSTM model using spatial data attributes. The authors assessed and verified the model's impact using PeMS's dataset. The experimental findings showed that the Bi-LSTM model predicted better than the SVR and GRU models. MAE, MAPE, and RMSE comparisons showed that the proposed model was 30.4%, 32.2%, and 39.6% more accurate.

Table 6 Performance of the proposed model in [102] for different metrics

Model	60 min prediction			120 min prediction		
	MAPE	MAE	RMSE	MAPE	MAE	RMSE
DELA	0.1721	24.37	43.19	0.1707	23.31	43.11

In [101], the authors created a Traffic Congestion Index (TCI) using free-flowing speeds and distances. The author showed how to group road linkages by sub-graph to pre-train the DL model and share information. They exhibited Spatial Graph Convolutional Neural Network (SG-CNN), a traffic congestion prediction model that blends the best of the CNN model with traffic data, and how road segment aggregation can speed up training. MAE, RMSE, and MAPE indices were used to assess the model's accuracy: 0.01578, 0.02264, and 0.06148. Weather, pedestrians, road conditions, and other factors affecting traffic congestion must be considered to improve the TCI. Given their time requirements, segment aggregation methods would be interesting to improve.

In [102], the authors proposed the usage of Deep and Embedded Learning Approach (DELA) to explicitly acquire precise traffic information, road structure, and meteorological conditions. Officially offered by the Knowledge Discovery and Data Mining Tools Competition, the initial highway traffic data set included the three-month period between July 19, 2016, and October 17, 2016. (KDD CUP 2017). The prediction accuracy of the chosen DL techniques (CNN and LSTM) for the proposed DELA model was evaluated by MAE, MAPE, and RMSE and is illustrated in Table 6.

For the chosen DL models (CNN and LSTM), the suggested model provides weak explanations. Not only that but its embedded component's capacity for learning is somewhat restricted.

Congestion levels were predicted in [103] using refined GPS trajectory data. To compare GPS path information to the road system, the Hidden Markov model has been used. Using GPS trajectory data from adjacent areas, the true speed of road segments can be determined. Congestion levels were predicted using a mix of three traditional machine learning models (the ARIMA model, SVR, and ridge regression) and four DL methods (CNNs, RNNs, LSTM, and GRU). The performance of the CNN, RNN, LSTM, and GRU models is illustrated in Table 7.

It was noted that there were certain issues with this study. To begin, Additional GPS information must be considered. There's also room for modification in the CNN network's architecture to maximize model efficacy.

A spatiotemporal model (CPM-ConvLSTM) can estimate short-term congestion along the entire path, as proposed in [104]. The suggested model uses a geographical matrix to account for the spatial link between road segments and congestion distribution. Data on Helsinki traffic was collected. The scientists fed a time series of previous spatiotemporal

Table 7 Accuracy of the proposed model in [103]

Metric	Model	Road network 1	Road network 2
Accuracy (%)	CNN	96.32	96.23
	RNN	95.59	95.88
	LSTM	96.08	96.65
	GRU	95.40	95.39

matrices into the widely used ConvLSTM DL model to predict the future short-range matrix. Adding landmarks, climate, and geography improves prediction accuracy. The model was assessed using MSE and MAE. Individual metric findings are shown in Table 8.

CNN-based direct traffic forecasting uses a time–space diagram [105]. CNN employs dynamic time–space diagram data to predict traffic accurately. This approach has three advantages: (1) A learning mechanism that understands the time–space diagram's key features for successful forecasting can be used directly as input without abstraction or aggregation. Diagram (2) can be utilized directly in foresight. These variables greatly affect traffic flow and vehicle interactions, which may affect future traffic conditions; (3) this approach gave location-specific responses that needed to be re-calibrated for another location, addressing the transferability problem of non-parametric models. The CNN model predicted traffic states in multiple main diagram regions better than SVR, MLP, and ARIMA nonparametric models. The proposed model was evaluated by MSE, MAE, and MAPE. In Table 9, metrics are shown.

Both synthetic and real-world data were used in the training of the recommended CNN model. However, the impact of switching lanes on traffic dynamics and prediction accuracy was not investigated in this study.

Comparative Analysis of Evaluation Metrics in the Reviewed Studies

The evaluation of traffic prediction models employs a range of metrics, each serving to measure specific aspects of model performance. The following discussion outlines key metrics and their applications, providing a comprehensive comparison based on their usage in recent studies.

Table 8 Performance of the proposed model in [104] for different metrics

Model	Weekday		Weekend	
	MSE	MAE	MSE	MAE
CPM-ConvLSTM	0.270992	0.187260	0.075692	0.110878

Table 9 Performance of the proposed model in [105] for different metrics

Criteria	MSE	MAE	MAPE
Flow	16,939.98	91.83	177.62
Density	9.70	2.30	10.57

R and R²

The R metric assesses the linear relationship between observed and predicted values, ranging from -1 to $+1$. It is widely used to evaluate the strength and direction of linear relationships. For instance, the study in [78] reported an R of 0.9995, indicating a near-perfect prediction using a PSO-Bi-LSTM model. R² metric measures the proportion of variance in the dependent variable explained by the model. Higher R² values signify better explanatory power. In [80], the model's R² value was used to show its ability to capture traffic flow variability. R and R² are crucial for models where capturing linear dependencies is key. High values suggest that the model accurately reflects the underlying patterns in traffic data.

MSE and RMSE

The MSE metric quantifies the average squared difference between predicted and actual values, penalizing larger errors. Although MSE is less interpretable due to its squared units, it is effective in highlighting models sensitive to outliers. Studies like [100] and [104] used MSE to compare various traffic prediction models. The RMSE is the square root of MSE, retaining the original data units and enhancing interpretability. In [104], the RMSE was used to assess the CPM-ConvLSTM model, achieving a value of 0.270992 for weekday congestion predictions. The RMSE is generally favored for its interpretability, especially when understanding the magnitude of errors is crucial in traffic forecasting.

MAE

The MAE metric measures the average magnitude of errors without considering their direction, making it a straightforward and interpretable metric. The study in [100] reported a 30.4% improvement in MAE using a Bi-LSTM model, indicating its effectiveness in handling traffic data. It is commonly employed due to its simplicity and unit consistency. It provides a consistent measure of prediction accuracy, especially in datasets with fewer outliers.

MAPE and SMAPE

The MAPE metric calculates the average percentage difference between predicted and actual values, offering insight into the model's relative accuracy. For example, [87] achieved a MAPE of 0.1721 in 60-min traffic predictions, demonstrating low relative error. The SMAPE metric provides an optimized version of MAPE, accounting for both positive and negative errors symmetrically. Although less frequently used, it addresses some of the limitations of MAPE, especially in scenarios where actual values are near zero. Both MAPE and SMAPE are valuable for assessing relative errors. However, they can be sensitive to small actual values, necessitating caution in their application.

Confusion Matrix, Precision, Recall, F1-Score, and Accuracy

The Confusion Matrix is a fundamental metric for evaluating classification models, detailing true positives, false positives, true negatives, and false negatives. The Precision and Recall metrics measure the accuracy of positive predictions and the model's ability to identify all relevant instances, respectively. In traffic congestion prediction studies, such as [81], these metrics are crucial for understanding the model's classification performance. The F1-Score metric combines precision and recall, offering a balanced metric for datasets with imbalanced classes, such as varying congestion states. The Accuracy metric quantifies the overall correctness of the model's predictions, useful for providing a general sense of model performance. All These metrics are essential for classification tasks in traffic prediction. Their combined use, particularly the F1-score, provides a nuanced evaluation of model performance, especially in handling diverse traffic states.

ER

The ER metric represents the proportion of incorrect predictions out of the total number of predictions. It offers a straightforward way to understand model errors, especially when accuracy alone may not be sufficient. Although less detailed than other metrics, ER provides a quick overview of model performance, helping to identify the need for further optimization.

To summarize this section, Table 10 compares the previous metrics in terms of the purpose of each metric, the studies which utilized it, the key findings, and implications.

This comparison illustrates that selecting appropriate metrics depends on the model's goals and data characteristics. RMSE and MAE are common choices for regression-based models due to their interpretability and straightforward application. Meanwhile, MAPE and SMAPE are

preferred for evaluating relative errors but require caution with small data values. For classification tasks, precision, recall, F1-score, and confusion matrix metrics are indispensable for a comprehensive evaluation, particularly when dealing with imbalanced traffic states.

Results and Discussion

This paper presented a comparative study of the most popular DL techniques adopted in traffic prediction. It focused on works that are directly related to using DL methods for traffic flow prediction. The total number of papers that were discussed in the literature review of this study was 30 which spanned years from 2019 to 2023. Figure 2 illustrates the number of papers discussed each year. In this section, the reviewed papers were classified by year from the oldest to the newest. Table 11 compares previous works discussed in the literature review of this study in terms of the methodology proposed in each study, the adopted techniques, the used datasets, and the findings of each study.

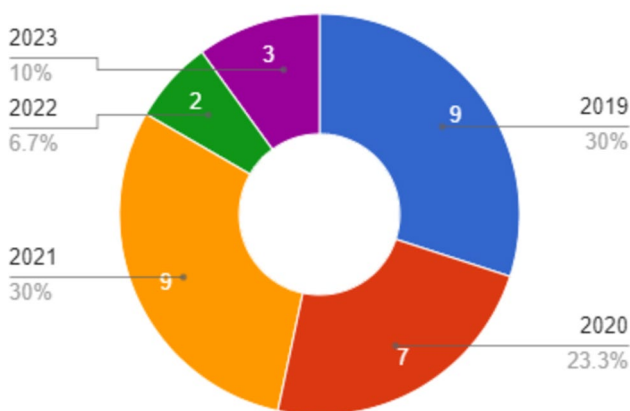
As shown in Table 3, the most widely used DL techniques in 2019 and 2020 were CNN, LSTM, and their variations. In 2021, multiple DL techniques were used. Whereas, in 2022 and 2023, the most widely used DL techniques were LSTM and its variations. The previous comparison indicates that LSTM and its variants (such as Bi-LSTM, GRU, and hybrid models like PSO-LSTM) consistently deliver high prediction accuracy across diverse datasets and environments. LSTM's ability to model long-term dependencies and temporal patterns in sequential traffic data makes it a leading choice for traffic forecasting. Also, Combining LSTM with other techniques, such as CNN for spatial data (e.g., ConvLSTM) or optimization algorithms (e.g., PSO-LSTM), further enhances predictive performance and robustness. Given the evidence, LSTM-based models especially when integrated into hybrid frameworks with CNNs or optimization methods (e.g., PSO) appear to be the most effective techniques for accurate and robust traffic prediction.

Limitations and Future Directions

Traffic congestion remains a critical issue, particularly in densely populated urban areas, making accurate traffic forecasting increasingly important. To address this challenge, effective and reliable methods for predicting future traffic patterns are essential. This review has identified a gap in the current literature related to traffic flow forecasting, specifically highlighting the need for more computationally efficient methods and algorithms. Additionally, the availability of high-quality training data remains a concern. Many network models have been trained on partial or incomplete

Table 10 Comparison of Evaluation Metrics

Metric	Purpose	Studies utilizing it	Key findings	Implications
R	Measures the strength of linear correlation	[78, 80]	$R = 0.9995$ in [78] suggests near-perfect prediction	Essential for evaluating linear relationships
R^2	Explains the variance in the dependent variable	[80]	High R^2 indicates strong model explanatory power	Indicates the model's fit to data variance
MSE	Measures average squared errors, penalizing large errors	[100, 104]	Highlights model sensitivity to outliers	Useful for models sensitive to large deviations
RMSE	Square root of MSE; retains original units	[87, 100, 104]	$RMSE = 0.270992$ in [104] reflects accurate congestion prediction	Preferred for its interpretability in original units
MAE	Measures the average magnitude of errors	[87, 100, 104]	30.4% improvement in MAE using Bi-LSTM in [100]	Simple, unit-consistent, good for general error assessment
MAPE	Measures relative error in percentages	[87, 103, 104]	MAPE of 0.1721 indicates low prediction error	Useful for relative error; sensitive to small actual values
SMAPE	Optimized version of MAPE for balanced error assessment	[87]	Provides symmetric evaluation of positive/negative errors	Suitable for cases with near-zero actual values
Confusion Matrix	Provides detailed classification accuracy	[81, 82]	Used to understand false positives/negatives in models	Essential for classification tasks
Precision, Recall, F1-Score	Measures positive prediction accuracy and balance	[81]	Crucial for understanding imbalanced traffic states	Necessary for evaluating classification performance
Accuracy	Measures overall prediction correctness	[81, 103]	Commonly used for a general performance overview	Simple but insufficient for imbalanced datasets
ER	Quantifies the proportion of incorrect predictions	[82, 83]	Provides a quick assessment of prediction failures	Basic metric for quick performance evaluation

Number of Discussed Papers per Year**Fig. 2** Number of discussed papers per year

datasets due to limitations in matching city-specific traffic flow data. As a result, the performance and accuracy of deep learning (DL) models in predicting traffic flow are inherently constrained.

The scarcity of well-documented spatiotemporal associations in DL methods presents another gap, particularly in capturing the complex relationships between road segments,

congestion patterns, and high-traffic zones. Furthermore, the limitations in computational power and distributed storage present significant challenges for processing large volumes of real-time traffic data. These factors require further exploration to enhance the accuracy and scalability of traffic forecasting models.

The scope of this review is limited to the methods and algorithms discussed in the analyzed articles. It is possible that other effective techniques and models were overlooked, suggesting the need for a more comprehensive investigation into alternative DL approaches. Future research should prioritize the exploration of other popular DL techniques, particularly those not covered in this study, to further expand the body of knowledge in traffic forecasting.

The field of traffic forecasting using deep learning has vast potential, yet it faces numerous challenges and limitations. Overcoming these challenges requires future research efforts to develop innovative solutions that push the boundaries of current forecasting capabilities. Research should explore innovative methods for gathering data, such as incorporating data from smart city initiatives, Internet of Things (IoT) devices, and real-time traffic monitoring systems. This approach would contribute to the development of robust models capable of accurate traffic predictions, even in smaller cities with more variable traffic conditions. Future

Table 11 Comparison of related works grouped by year

Year	References	Datasets	Techniques	Findings
2019	[98]	Xiangjiang Middle Road in Changsha City's data	GRU	Road segment traffic was forecast Weather was ignored
	[97]	The Beijing cab real route	FIS and CNN	Potential traffic factors needed Effective DL models required
	[105]	Synthetic and real-world data	CNN	CNN model outperformed SVR, MLP, and ARIMA No prediction accuracy
	[104]	Traffic data in Helsinki, Finland	ConvLSTM	Integration of external factors enhanced accuracy
	[103]	GPS trajectory data	CNNs, RNNs, LSTM, and GRU	More GPS information needed Network's architecture needs optimization
	[77]	Data from AVI detectors in Xuancheng, China	KNN, ANN, CNN, and LSTM	High accuracy
	[76]	Traffic data collected by PeMS	MIR and LSTM	More fine-tuned variables needed Time frame for model training required
	[73]	11 years Caltrans's traffic data	CNN, LSTM, GPU, Big Data, and In-memory computing	Limited data set Low accuracy
	[102]	KDD CUP 2017	CNN and LSTM	Weak models' explanations Restricted learning capacity
2020	[96]	Data collected in Gangnam-gu, Seoul during November, 2018	LSTM	MAPE approximately 5% Accurate traffic prediction
	[94]	TOPIS service data	LSTM	Low prediction accuracy Other data sources required
	[93]	Plan4Safety, OpenReach, NJCMS, NJSLD, and probe vehicle databases	CNN and Deep ANN	With enough data, the model may forecast delays in the other direction
	[92]	FCD collected using Aimsun (2018)	HGC-LSTM, LSTM, GRU, RNN, and SRCN	In Amara, LSTM and GRU are ideal Camp Nou scenario HGC-LSTM is best
	[91]	Beijing's microwave sensor traffic data	GRU, LSTM, and BCM	Other data needed to improve accuracy
	[90]	San Francisco Bay Area Highway Network condensed	LSTM and Graph-CNN	The model was useful for congestion tolling
	[89]	Caltrans PeMS	SVR and LSTM	Network structure needs enhancements Parameter choices need optimization
2021	[88]	DRIVE Net dataset	FI-RNN	Low Model accuracy Other factors needed
	[87]	Belgian Freeway System, Bruxelles Region Urban Network	FCNN, GPS	Low model accuracy
	[81]	Adaptive traffic light data from InSync	SMBO, LSTM	Public data availability can improve the model performance
	[101]	PeMS dataset and Los Angeles' highway data	SG-CNN	Other factors must be considered Poor model accuracy
	[85]	Aliyun Tianchi platform data	BPNN	High model 's performance
	[84]	IoV's data during severe weather	DBN and SVR	Low prediction error Long computing time
	[83]	Road networks data in Shenzhen, China	RNN and MDN	Small size dataset
	[82]	Waze Live Map APIs' data	K-Means and ANN	Other factors needed
	[81]	Online dataset	DAN, DBN, and LSTM	LSTM: 95.2% accuracy No dataset mentioned
2022	[80]	Traffic information in central Kunming City, China	PSO-LSTM	High prediction accuracy
	[100]	Caltrans PeMS	CNN and Bi-LSTM	High prediction accuracy

Table 11 (continued)

Year	References	Datasets	Techniques	Findings
2023	[79]	Electromagnetic loops and cameras	LSTM	Low error rate
	[99]	Five datasets: Four from PeMS, and the SZ-taxi dataset in Shenzhen, China	ST-CGCN and LSTM	High prediction accuracy
	[78]	Inner Ring Road, South Extension, Delhi, India	PSO-Bi-LSTM	High accuracy and stability

studies should also focus on developing models that can capture intricate spatiotemporal relationships more effectively by the integration of advanced graph-based neural networks, attention mechanisms, Geographic Information Systems (GIS) data, weather patterns, and real-time events to understand traffic dynamics in various spatial contexts. Research into optimizing computational efficiency will be crucial for advancing traffic forecasting models. Future studies could explore the integration of data from multiple modes of transportation, such as public transit, biking, and pedestrian movement. This multi-modal approach would provide a more holistic view of urban traffic dynamics, allowing DL models to make more informed predictions. Finally, there is a need for the development of real-time adaptive traffic forecasting systems that can dynamically respond to changes in traffic patterns. Such adaptive systems could significantly enhance the effectiveness of traffic management strategies.

Conclusion

This research provides a comprehensive overview of the most significant DL techniques currently employed in traffic forecasting, along with an exploration of the challenges inherent to this field. A total of 30 papers were carefully selected for in-depth evaluation, revealing the critical role of DL in achieving accurate traffic forecasts. The findings of this study have profound implications for urban planning and traffic management, as precise traffic predictions can inform the design and optimization of road infrastructure, enhance the efficiency of transportation networks, and support the development of proactive traffic control measures. Additionally, this research contributes to the growing body of literature on DL methods for traffic flow forecasting and serves as a valuable resource for researchers aiming to further advance the field. By identifying key challenges and opportunities, this study paves the way for future investigations that can drive more effective urban planning, reduce congestion, and promote safer, more sustainable transportation systems.

Funding No funding was received for this work.

Declarations

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Nellore K, Hancke G. A survey on urban traffic management system using wireless sensor networks. *Sensors*. 2016;16:157. <https://doi.org/10.3390/s16020157>.
- Patel P, Narmawala Z, Thakkar A. A survey on intelligent transportation system using internet of things. *Emerg Res Comput, Inform, Commun Appl*. 2019. https://doi.org/10.1007/978-981-13-5953-8_20.
- An S, Lee B.-H, Shin D.-R. A survey of intelligent transportation systems. 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks. 2011. <https://doi.org/10.1109/CICSyN.2011.76>
- Qureshi KN, Abdullah AH. A Survey on intelligent transportation systems. *Middle-East J Sci Res*. 2013;15:629–42. <https://doi.org/10.5829/idosi.mejsr.2013.15.5.11215>.
- Chen C, Li K, Teo SG, 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)*. 2020;14(4):1–23. <https://doi.org/10.1145/3385414>.
- Sun P, Boukerche A, Tao Y. SSGRU: A novel hybrid stacked GRU- based traffic volume prediction approach in a road network. *Comput Commun*. 2020;160:502–11. <https://doi.org/10.1016/j.comcom.2020.06.028>.
- Makaba T, Doorsamy W, Paul BS. Exploratory framework for analyzing road traffic accident data with validation on Gauteng province data. *Cogent Eng*. 2020;7(1):1834659. <https://doi.org/10.1080/23311916.2020.1834659>.
- World Health Organization. “Global status report on road safety 2018 SUMMARY.” [Online]. Available: <https://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf?ua=1>. 2018. Accessed 05 November 2022.
- Bengio Y. Learning deep architectures for AI. *Found Trends Mach Learn*. 2009;2(1):1–127. <https://doi.org/10.1561/2200000006>.
- Van Der Voort M, Dougherty M, Watson S. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. *Transp Res Part C: Emerg Technol*. 1996;4(5):307–18. [https://doi.org/10.1016/S0968-090X\(97\)82903-8](https://doi.org/10.1016/S0968-090X(97)82903-8).
- Lee S, Fambro DB. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transp Res Rec: J Transp Res Board*. 1999;1678(1):179–88. <https://doi.org/10.3141/1678-22>.
- Williams BM. Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling. *Transp Res Rec: J Transp Res Board*. 2001;1776(1):194–200. <https://doi.org/10.3141/1776-25>.

13. Williams BM, Hoel LA. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *J Transp Eng*. 2003;129(6):664–72. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(664\)](https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664)).
14. Chen K, Chen F, Lai B, Jin Z, Liu Y, Li K, Wei L, Wang P, Tang Y, Huang J, Hua X. Dynamic Spatio-temporal graph-based CNNs for traffic flow prediction. *IEEE Access*. 2020;8:185136–45. <https://doi.org/10.1109/ACCESS.2020.3027375>.
15. Joshi M, Hadi TH. A review of network traffic analysis and prediction techniques. *arXiv preprint arXiv:1507.05722*. 2015.
16. Hoong NK, Hoong PK, Tan IK, Muthuvelu N, Seng LC. Impact of utilizing forecasted network traffic for data transfers. In: 13th International Conference on Advanced Communication Technology (ICACT2011). 2011. pp. 1199–1204. IEEE.
17. Sadek N, Khotanzad A. Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model. In 2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577) (Vol. 4, pp. 2148–2152). IEEE; 2004.
18. Yu Y, Wang J, Song M, Song J. Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient. In: 2010 International Conference on Intelligent System Design and Engineering Application (Vol. 1, pp. 980–983). IEEE; 2010.
19. Vujicic B, Chen H, Trajkovic L. Prediction of traffic in a public safety network. In 2006 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 4–pp). IEEE; 2006.
20. Anand NC, Scoglio C, Natarajan, B. GARCH—Non-linear time series model for traffic modeling and prediction. In: NOMS 2008–2008 IEEE Network Operations and Management Symposium (pp. 694–697). IEEE; 2008.
21. Yu ES, Chen CR. Traffic prediction using neural networks. In: Proceedings of GLOBECOM'93. IEEE Global Telecommunications Conference (pp. 991–995). IEEE; 1993.
22. Park DC, Woo DM. Prediction of network traffic using dynamic bilinear recurrent neural network. In 2009 Fifth International Conference on Natural Computation (Vol. 2, pp. 419–423). IEEE. 2009.
23. Kashyap AA, Raviraj S, Devarakonda A, Nayak KSR, Bhat SJ. Traffic flow prediction models—a review of deep learning techniques. *Cogent Eng*. 2022;9(1):2010510. <https://doi.org/10.1080/23311916.2021.2010510>.
24. Smith BL, Demetsky, MJ. Short-term traffic flow prediction models—a comparison of neural network and nonparametric regression approaches. In: Proceedings of IEEE international conference on systems, man and cybernetics (Vol. 2, pp. 1706–1709). IEEE. <https://doi.org/10.1109/ICSMC.1994.400094>. 1994.
25. Zhang J, Wang F-Y, Wang K, Lin W-H, Xu X, Chen C. Data-driven intelligent transportation systems: a survey. *IEEE Trans Intell Transp Syst*. 2011;12(4):1624–39. <https://doi.org/10.1109/TITS.2011.2158001>.
26. Chowdary GJ. Machine learning and deep learning methods for building intelligent systems in medicine and drug discovery: a comprehensive survey. 2021; <https://doi.org/10.48550/arXiv.2107.14037>.
27. Alpaydin E. Introduction to machine learning. 2020: MIT press.
28. Mikolov T, et al. Efficient estimation of word representations in vector space. 2013; <https://doi.org/10.48550/arXiv.1301.3781>.
29. Noor A, Pattanaik P, Khan MZ, Alromema W, Noor TH. Deep feature detection approach for COVID-19 classification based on X-ray Images. *Int J Adv ComputSci Appl*. 2023. <https://doi.org/10.14569/IJACSA.2023.0140514>.
30. Pinto Neto EC, Baum DM, Almeida JRD Jr, Camargo JB Jr, Cugnasca PS. Deep learning in air traffic management (ATM): a survey on applications, opportunities, and open challenges. *Aerospace*. 2023;10(4):358.
31. Nguyen G, Dlugolinsky S, Bobák M, Tran V, García ÁL, Heredia I, et al. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artif Intell Rev*. 2019;52(1):77–124. <https://doi.org/10.1007/s10462-018-09679-z>.
32. Aggour KS, Gupta VK, Ruscitto D, Ajdelsztajn L, Bian X, Brosnan KH, et al. Artificial intelligence/machine learning in manufacturing and inspection: a GE perspective. *MRS Bull*. 2019;44(7):545–58. <https://doi.org/10.1557/mrs.2019.157>.
33. Khan FN, Fan Q, Lu C, Lau APT. Machine learning methods for optical communication systems and networks. In: Optical fiber telecommunications VII. New York: Academic Press; 2020. p. 921–78.
34. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP, et al. A survey on deep learning: algorithms, techniques, and applications. *ACM Comput Surv (CSUR)*. 2018;51(5):1–36. <https://doi.org/10.1145/3234150>.
35. Dargan S, Kumar M, Ayyagari MR, Kumar G. A survey of deep learning and its applications: a new paradigm to machine learning. *Arch Comput Methods Eng*. 2019;27(4):1–22. <https://doi.org/10.1007/s11831-019-09344-w>.
36. Lauzon FQ. An introduction to deep learning. In: 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), IEEE. 2012; pp. 1438–1439. <https://doi.org/10.1109/ISSPA.2012.6310529>.
37. Ling ZH, Kang SY, Zen H, Senior A, Schuster M, Qian XJ, Deng L. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *IEEE Signal Process Mag*. 2015;32(3):35–52. <https://doi.org/10.1109/MSP.2014.2359987>.
38. Schmidhuber J. Deep learning in neural networks: An overview. *Neural Netw*. 2015;61:85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>.
39. Yu D, Deng L. Deep learning and its applications to signal and information processing [exploratory dsp]. *IEEE Signal Process Mag*. 2010;28(1):145–54. <https://doi.org/10.1109/MSP.2010.939038>.
40. Almukhalifi H, Noor A, Noor TH. Traffic management approaches using machine learning and deep learning techniques: a survey. *Eng Appl Artif Intell*. 2024;133: 108147.
41. Gomes B, Coelho J, Aidos H. A survey on traffic flow prediction and classification. *Intell Syst Appl*. 2023;20: 200268.
42. Yap MH, Pons G, Martí J, Ganau S, Sentsis M, Zwigelaar R, Martí R. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE J Biomed Health Inform*. 2017;22(4):1218–26. <https://doi.org/10.1109/JBHI.2017.2731873>.
43. Fukushima K. Neocognitron: a hierarchical neural network capable of visual pattern recognition. *Neural Netw*. 1988;1:119–30.
44. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86:2278–324. <https://doi.org/10.1109/5.726791>.
45. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Asari VK. A state-of-the-art survey on deep learning theory and architectures. *Electronics*. 2019;8(3):292. <https://doi.org/10.3390/electronics8030292>.
46. Yang D, Li S, Peng Z, Wang P, Wang J, Yang H. MF-CNN: traffic flow prediction using convolutional neural network and multi-features fusion. *IEICE Trans Inf Syst*. 2019;102(8):1526–36.
47. Wang X, Wang Z, Yang K, Song Z, Bian C, Feng J, Deng C. A survey on deep learning for cellular traffic prediction. *Intell Comput*. 2024;3:0054.
48. Apaydin H, Feizi H, Sattari MT, Colak MS, Shamshirband S, Chau KW. Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water*. 2020;12(5):1500. <https://doi.org/10.3390/w12051500>.

49. Graves A, Mohamed, AR, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. 2013. pp. 6645–6649. <https://doi.org/10.1109/ICASSP.2013.6638947>.
50. Batur Dinler Ö, Aydin N. An optimal feature parameter set based on gated recurrent unit recurrent neural networks for speech segment detection. *Appl Sci*. 2020;10(4):1273. <https://doi.org/10.3390/app10041273>.
51. Jagannatha AN, Yu H. Structured prediction models for RNN-based sequence labeling in clinical text. In: Proceedings of the conference on empirical methods in natural language processing. Conference on empirical methods in natural language processing. 2016; (Vol. 2016, p. 856). NIH Public Access. <https://doi.org/10.18653/2Fv1/2Fd16-1082>.
52. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Farhan L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J Big Data*. 2021;8(1):1–74. <https://doi.org/10.1186/s40537-021-00444-8>.
53. Pascanu R, Gulcehre C, Cho K, Bengio Y. How to construct deep recurrent neural networks. 2013; <https://doi.org/10.48550/arXiv.1312.6026>.
54. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics; 2010; p. 249–56.
55. Wang J, Wang Y, Yun M, Yang X. Development of urban road network traffic state dynamic estimation method. *Math Probl Eng*. 2015;2015(1): 714149.
56. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
57. Smagulova K, James AP. A survey on LSTM memristive neural network architectures and applications. *Eur Phys J Spec Top*. 2019;228(10):2313–24. <https://doi.org/10.1140/epjst/e2019-900046-x>.
58. Setyanto A, Laksito A, Alarfaj F, Alreshoodi M, Oyong I, Hayaty M, Kurniasari L. Arabic language opinion mining based on long short-term memory (LSTM). *Appl Sci*. 2022;12(9):4140. <https://doi.org/10.3390/app12094140>.
59. Gers FA, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. *Neural Comput*. 2000;12(10):2451–71. <https://doi.org/10.1162/089976600300015015>.
60. Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*. 2021;99:650–5. <https://doi.org/10.1016/j.procir.2021.03.088>.
61. Wang J, Hu X. Convolutional neural networks with gated recurrent connections. *IEEE Trans Pattern Anal Mach Intell*. 2021. <https://doi.org/10.1109/TPAMI.2021.3054614>.
62. Liang M, Hu X. Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015; pp. 3367–3375.
63. Liang, M., Hu, X., & Zhang, B. Convolutional neural networks with intra-layer recurrent connections for scene labeling. *Adv Neural Inform Process Syst*. 2015; 28.
64. Naidu G, Zuva T, Sibanda EM. A review of evaluation metrics in machine learning algorithms. In: Computer science on-line conference. Cham: Springer International Publishing; 2023. p. 15–25.
65. Jierula A, Wang S, Oh TM, Wang P. Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Appl Sci*. 2021;11(5):2314.
66. Di Bucchianico, A. Coefficient of determination (R²). *Encyclopedia of statistics in quality and reliability*. 2008.
67. Kim S, Alizamir M, Zounemat-Kermani M, Kisi O, Singh VP. Assessing the biochemical oxygen demand using neural networks and ensemble tree approaches in South Korea. *J Environ Manage*. 2020;270: 110834.
68. Hyndman RJ. Another look at forecast-accuracy metrics for intermittent demand. *Foresight: Int J Appl Forecast*. 2006;4(4):43–6.
69. Kim CH, Kim YC. Application of artificial neural network over nickel-based catalyst for combined steam-carbon dioxide of methane reforming (CSDRM). *J Nanosci Nanotechnol*. 2020;20(9):5716–9.
70. Sanders NR. Measuring forecast accuracy: some practical suggestions. *Prod Invent Manag J*. 1997;38(1):43.
71. Deng X, Liu Q, Deng Y, Mahadevan S. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Inf Sci*. 2016;340:250–61.
72. Lippi M, Bertini M, Frasconi P. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Trans Intell Transp Syst*. 2013;14(2):871–82. <https://doi.org/10.1109/TITS.2013.2247040>.
73. Aqib M, Mehmood R, Alzahrani A, Katib I, Albeshri A, Altowaijri SM. Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs. *Sensors*. 2019;19:2206. <https://doi.org/10.3390/s19092206>.
74. Janković S, Uzelac A, Zdravković S, Mladenović D, Mladenović S, Andrijačić I. Traffic volumes prediction using big data analytics methods. *Int J Traffic Transp Eng*. 2021. [https://doi.org/10.7708/jtte2021.11\(2\).01](https://doi.org/10.7708/jtte2021.11(2).01).
75. California Department of Transportation (Caltrans). Caltrans Performance Measurement System (PeMS) Available online: <http://pems.dot.ca.gov/> (accessed on 5 November 2022).
76. Kong F, Li J, Jiang B, Zhang T, Song H. Big data-driven machine learning-enabled traffic flow prediction. *Trans Emerg Telecommun Technol*. 2019;30(9): e3482. <https://doi.org/10.1002/ett.3482>.
77. Wang J, Chen R, He Z. Traffic speed prediction for urban transportation network: a path based deep learning approach. *Trans Res Part C: Emerg Technol*. 2019;100:372–85. <https://doi.org/10.1016/j.trc.2019.02.002>.
78. Redhu P, Kumar K. Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM. *Phys A: Stat Mech Appl*. 2023;625: 129001.
79. Slimani N, Yousfi S, Amghar M, Sbity N. Towards a new intelligent traffic system based on deep learning and data integration. *Int J Electr Comput Eng (IJECE)*. 2023;13(4):4649–60.
80. Mao Y, Qin G, Ni P, Liu Q. Analysis of road traffic speed in Kunming plateau mountains: a fusion PSO-LSTM algorithm. *Int J Urban Sci*. 2022;26(1):87–107.
81. Ramchandra NR, Rajabhushanam C. Machine learning algorithms performance evaluation in traffic flow prediction. *Mater Today: Proceed*. 2022;51:1046–50.
82. Pangesta J, Dharmadinata OJ, Bagaskoro ASC, Hendrikson N, Budiharto W. Travel duration prediction based on traffic speed and driving pattern using deep learning ICIC express letters. *Part B Appl Int J Res Surv*. 2021;12(1):83–90. <https://doi.org/10.24507/ijicelb.12.01.83>.
83. Chen M, Chen R, Cai F, Li W, Guo N, Li G. Short-term traffic flow prediction with recurrent mixture density network. *Math Problems Eng*. 2021. <https://doi.org/10.1155/2021/6393951>.
84. Bao X, Jiang D, Yang X, Wang H. An improved deep belief network for traffic prediction considering weather factors. *Alex Eng J*. 2021;60(1):413–20. <https://doi.org/10.1016/j.aej.2020.09.003>.
85. Jiang CY, Hu XM, Chen WN. An Urban Traffic Signal Control System Based on Traffic Flow Prediction. In: 2021 13th International Conference on Advanced Computational Intelligence

- (ICACI). 2021; (pp. 259–265). IEEE. <https://doi.org/10.1109/ICACI52617.2021.9435905>.
86. Rahman R, Hasan S. Real-time signal queue length prediction using long short-term memory neural network. *Neural Comput Appl*. 2021;33(8):3311–24. <https://doi.org/10.1007/s00521-020-05196-9>.
 87. Buroni G, Lebichot B, Bontempi G. AST-MTL: an attention-based multi-task learning strategy for traffic forecasting. *IEEE Access*. 2021;9:77359–70. <https://doi.org/10.1109/ACCESS.2021.3083412>.
 88. Qu L, Lyu J, Li W, Ma D, Fan H. Features injected recurrent neural networks for short-term traffic speed prediction. *Neurocomputing*. 2021;451:290–304. <https://doi.org/10.1016/j.neucom.2021.03.054>.
 89. Chen Y, Lv Y, Ye P, Zhu F. Traffic-condition-awareness ensemble learning for traffic flow prediction. *IFAC-PapersOnLine*. 2020;53(5):582–7. <https://doi.org/10.1016/j.ifacol.2021.04.146>.
 90. Mohanty S, Pozdnukhov A, Cassidy M. Region-wide congestion prediction and control using deep learning. *Transp Res Part C: Emerg Technol*. 2020;116: 102624. <https://doi.org/10.1016/j.trc.2020.102624>.
 91. 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*. 2020;21(3):1332–42. <https://doi.org/10.1109/TITS.2019.2939290>.
 92. Wang J, Deng W, Guo Y. New Bayesian combination method for short-term traffic flow forecasting. *Transp Res C, Emerg Technol*. 2014;43:79–94. <https://doi.org/10.1016/j.trc.2014.02.005>.
 93. Vázquez JJ, Arjona J, Linares M, Casanovas-García J. A comparison of deep learning methods for urban traffic forecasting using floating car data. *Transp Res Proced*. 2020;47:195–202. <https://doi.org/10.1016/j.trpro.2020.03.079>.
 94. Shabarek A. A deep machine learning approach for predicting freeway work zone delay using big data (Doctoral dissertation, New Jersey Institute of Technology). 2020.
 95. Ranjan N, Bhandari S, Zhao HP, Kim H, Khan P. City-wide traffic congestion prediction based on CNN, LSTM, and transpose CNN. *IEEE Access*. 2020;8:81606–20. <https://doi.org/10.1109/ACCESS.2020.2991462>.
 96. Shin DH, Chung K, Park RC. Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data. *IEEE Access*. 2020;8:150784–96. <https://doi.org/10.1109/ACCESS.2020.3016469>.
 97. An J, Fu L, Hu M, Chen W, Zhan J. A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information. *IEEE Access*. 2019;7:20708–22. <https://doi.org/10.1109/ACCESS.2019.2896913>.
 98. Dai G, Ma C, Xu X. Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU. *IEEE Access*. 2019;7:143025–35. <https://doi.org/10.1109/ACCESS.2019.2941280>.
 99. Bao Y, Huang J, Shen Q, Cao Y, Ding W, Shi Z, Shi Q. Spatial-temporal complex graph convolution network for traffic flow prediction. *Eng Appl Artif Intell*. 2023;121: 106044.
 100. Zhuang W, Cao Y. Short-term traffic flow prediction based on cnn-bilstm with multicomponent information. *Appl Sci*. 2022;12(17):8714.
 101. Tu Y, Lin S, Qiao J, Liu B. Deep traffic congestion prediction model based on road segment grouping. *Appl Intell*. 2021;51(11):8519–41. <https://doi.org/10.1007/s10489-020-02152-x>.
 102. Zheng Z, Yang Y, Liu J, Dai HN, Zhang Y. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Trans Intell Transp Syst*. 2019;20(10):3927–39. <https://doi.org/10.1109/TITS.2019.2909904>.
 103. Sun S, Chen J, Sun J. Traffic congestion prediction based on GPS trajectory data. *Int J Distrib Sens Netw*. 2019;15(5):1550147719847440. <https://doi.org/10.1177/1550147719847440>.
 104. Di X, Xiao Y, Zhu C, Deng Y, Zhao Q, Rao W. Traffic congestion prediction by spatiotemporal propagation patterns. In: 2019 20th IEEE International Conference on Mobile Data Management (MDM) 2019; (pp. 298–303). IEEE. <https://doi.org/10.1109/MDM.2019.00-45>.
 105. Khajeh Hosseini M, Talebpour A. Traffic prediction using time-space diagram: a convolutional neural network approach. *Transp Res Rec*. 2019;2673(7):425–35. <https://doi.org/10.1177/0361198119841291>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.