Journal of Electrical Systems
and Information Technology

REVIEW Open Access

Artificial intelligence-based traffic flow prediction: a comprehensive review

Sayed A. Sayed*, Yasser Abdel-Hamid and Hesham Ahmed Hefny

*Correspondence:
sayed.ahmed@narss.sci.eg

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

Abstract

The expansion of the Internet of Things has resulted in new creative solutions, such as smart cities, that have made our lives more productive, convenient, and intelligent. The core of smart cities is the Intelligent Transportation System (ITS) which has been integrated into several smart city applications that improve transportation and mobility. ITS aims to resolve many traffic issues, such as traffic congestion issues. Recently, new traffic flow prediction models and frameworks have been rapidly developed in tandem with the introduction of artificial intelligence approaches to improve the accuracy of traffic flow prediction. Traffic forecasting is a crucial duty in the transportation industry. It can significantly affect the design of road constructions and projects in addition to its importance for route planning and traffic rules. Furthermore, traffic congestion is a critical issue in urban areas and overcrowded cities. Therefore, it must be accurately evaluated and forecasted. Hence, a reliable and efficient method for predicting traffic is essential. The main objectives of this study are: First, present a comprehensive review of the most popular machine learning and deep learning techniques applied in traffic prediction. Second, identifying inherent obstacles to applying machine learning and deep learning in the domain of traffic prediction.

Keywords: ITS, AI, Traffic prediction, Traffic congestion, Machine learning, Deep learning

Introduction

In recent decades, the demand for the development of ITS-based solutions for precise traffic prediction and mobility management has increased as cities have gotten increasingly crowded and congested [1]. ITS is an advanced technology for delivering transportation by utilizing advanced data communication technologies through the integration of communications, computers, information, and other technologies and applying them to the transportation industry. This process aims to create an integrated system of people, roads, and vehicles [2]. ITS can construct a comprehensive, real-time, accurate, and effective transportation management system [3]. Furthermore, it has the potential to significantly reduce hazards, high accident rates, traffic congestion, carbon emissions, and air pollution, while also improving safety and dependability, travel speeds, traffic flow, and passenger satisfaction [4].

Precise traffic flow prediction is essential to the ITS as it can help traffic stakeholders (Individual passengers, traffic administrators, policymakers, and road users), shown

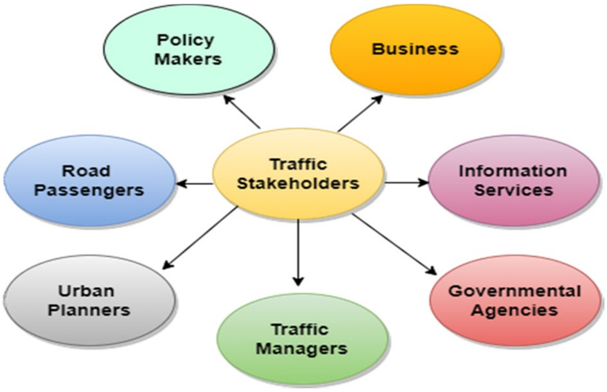

Fig. 1 Traffic stakeholders

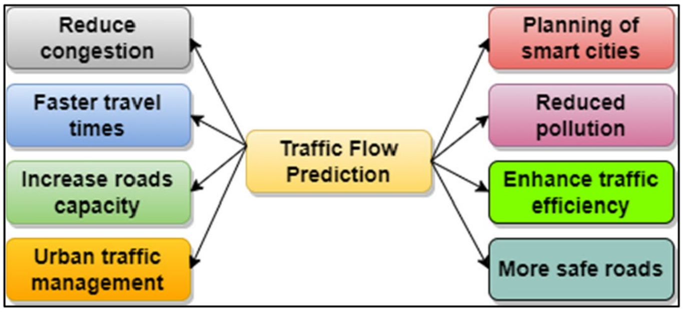

Fig. 2 Benefits of traffic flow forecasting

in Fig. 1, utilize transport networks more safely and intelligently [5, 6]. The efficacy of these systems depends on the quality of traffic data; only then, an ITS will be successful. According to the World Health Organization's (WHO) 2018 report on the universal status of road safety, road traffic deaths continue to rise, with 1.35 million deaths recorded in 2016, making the study of traffic forecasting a valuable method for reducing congestion and ensuring safer, more cost-effective travel [7, 8]. The benefits of traffic flow forecasting are illustrated in Fig. 2.

Historically, traffic flow forecasting was dependent on parametric models such as time series analysis derived from historical data. In time series, a collection of observed readings x is recorded at a specific time t. The objective is to recognize temporal patterns in past traffic data and use these results for forecasting. Another model for mobile stochastic problems capable of resolving regression concerns and minimizing variance to achieve optimal results was the Kalman Filtering method for time-series analysis [9]. Also, the Auto-Regressive Integrated Moving Average (ARIMA) model is a well-known and standard framework for predicting short-term traffic flow [10]. Numerous modifications to the ARIMA model were implemented, and the results ensured an enhanced performance [11–14].

Because traffic flow is stochastic and nonlinear, nonparametric models such as Random Forest (RF) Algorithm, Bayesian Algorithm (BA) approach, K-Nearest Neighbor

(KNN), Principal Component Analysis (PCA), and Support Vector Algorithms [9] have recently been employed in traffic flow prediction. In addition, neural networks became popularly employed for predicting traffic flow [15]. In the era of big data, a shallow Back-Propagation Neural Network (BPNN) [16] showed promising results. Thus, deep learning emerged, employing several layers to extract more complex properties from raw input. Convolutional Neural Networks (CNN) [17], Recurrent Neural Networks (RNN) [18], Long Short-Term Memory (LSTM) [19], Restricted Boltzmann Machines (RBM) [20], Deep Belief Networks (DBN) [21], and Stacked Auto-Encoder (SAE) [22] are some examples of deep learning architectures.

The primary goals of this research are to conduct a comprehensive survey of the key machine learning and deep learning techniques used in forecasting traffic flow in addition to identifying the obstacles and future directions for machine learning and deep learning in this field.

The rest of the paper is organized as follows: Section "Background" gives a theoretical background about traffic prediction problems, machine learning, and deep learning. Section "Survey methodology" outlines the survey methodology and presents a literature review of machine learning and deep learning approaches employed in traffic flow prediction. Section "Challenges" covers the existing challenges in the topic of this survey. Finally, Section "Conclusion" concludes the paper.

Background

ITS provides a bunch of high-resolution traffic data to be used in data-driven-based traffic flow prediction techniques [23]. From this perspective, traffic flow prediction can be considered as a time series problem in which the flow count at a future time is estimated based on data received from one or more observation points during prior periods. Traffic flow forecasting is a major component of traffic modeling, operation, and management. Accurately predicting traffic flows in real-time can give information and recommendations for road travelers to enhance their travel choices and decrease expenses, in addition to supplying authorities with enhanced traffic control tactics to alleviate congestion. Machine learning and deep learning as depicted in Fig. 3 are considered as subsets of artificial intelligence (AI) that have witnessed exponential expansion over years [24]. These approaches have been deemed successful in predicting traffic flow.

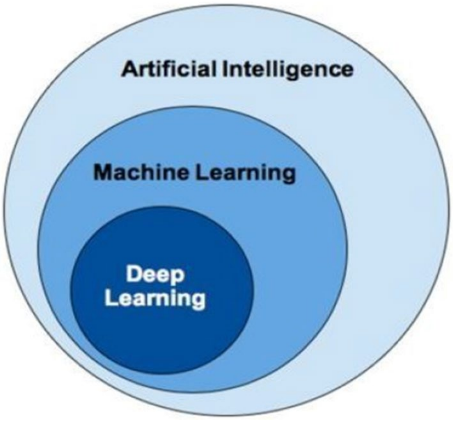

Fig. 3 AI, ML, and DL

Machine learning

Machine Learning (ML) techniques are considered statistical models that are utilized to make classifications and predictions based on the data provided [24]. ML is an area of AI that focuses on the development of prediction algorithms depending on the fair discovery of patterns within huge datasets and without being designed specifically for a particular job [25]. ML models are classified into three categories according to the learning techniques they employ: supervised learning, unsupervised learning, and reinforced learning (RL). In addition, ML algorithms might be further subdivided into several subgroups depending on distinct learning approaches, as shown in Fig. 4 [26].

Supervised learning

In the tasks that depend on supervised learning, a labeled dataset known as feature vectors and their corresponding predicted output labels are supplied to the model. The objective of these models is to create an inference function that maps feature vectors into output labels. When the ML model training is complete, it can make predictions based on new data. Continuous or discrete predictions can be generated using supervised learning algorithms [24]. Support Vector Machine (SMV), KNN, Logistic

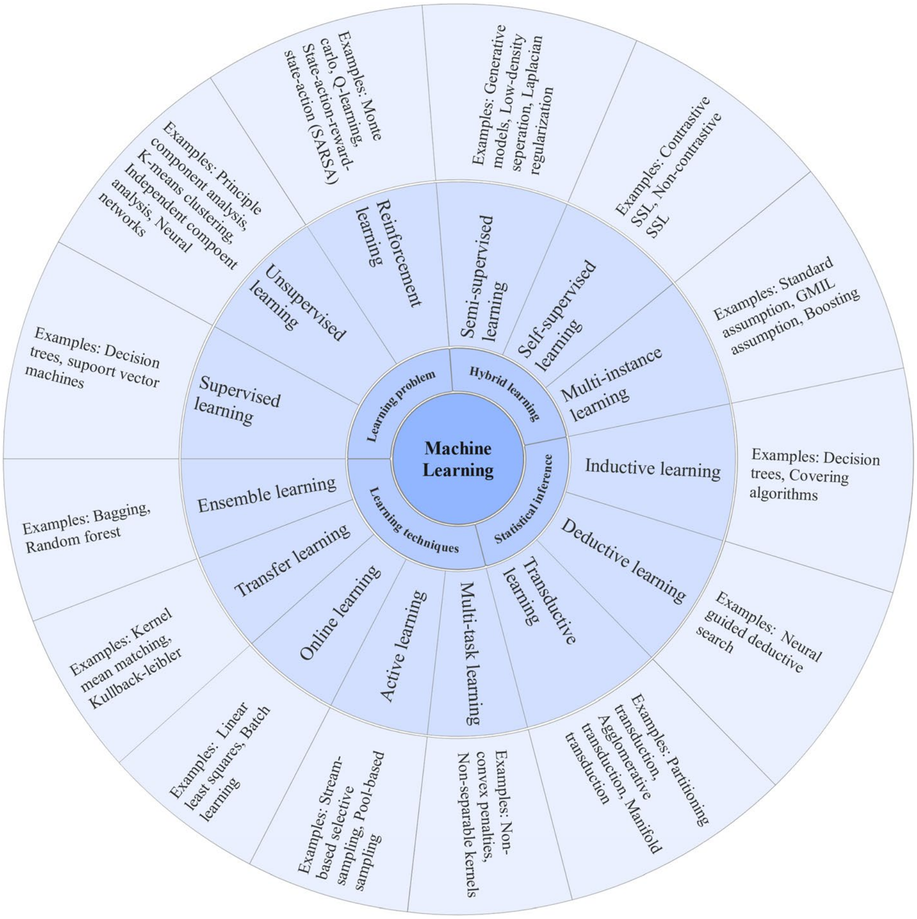
Machine Learning
Learning problem
Hybrid learning
Statistical inference
Learning techniques
Supervised learning
Unsupervised learning
Reinforcement learning
Semi-supervised learning
Self-supervised learning
Multi-instance learning
Inductive learning
Deductive learning
Transductive learning
Multi-task learning
Active learning
Online learning
Transfer learning
Ensemble learning
Examples: Decision trees, supoort vector machines
Examples: Principle component analysis, K-means clustering, Independent component analysis, Neural networks
Examples: Monte carlo, Q-learning, State-action-reward-state-action (SARSA)
Examples: Generative models, Low-density seperation, Laplacian regularization
Examples: Contrastive SSL, Non-contrastive SSL
Examples: Standard assumption, GMIL assumption, Boosting
Examples: Decision trees, Covering algorithms
Examples: Neural guided deductive search
Examples: Partitioning transduction, Agglomerative transduction, Manifold transduction
Examples: Non-convex penalties, Non-separable kernels
Examples: Stream-based selective sampling, Pool-based sampling
Examples: Linear least squares, Batch learning
Examples: Kernel mean matching, Kullback-leibler
Examples: Bagging, Random forest

Fig. 4 Various types of ML algorithms [26]

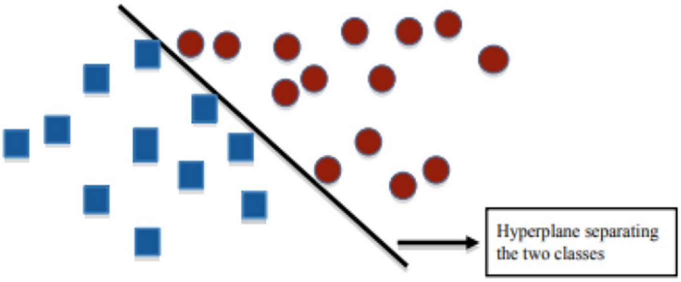

Fig. 5 A hyperplane separating two classes [27]

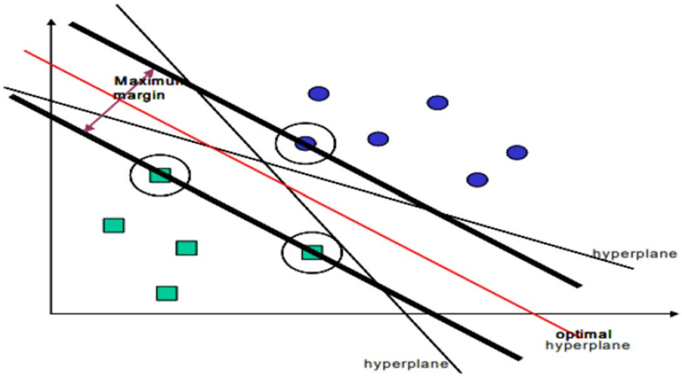

Fig. 6 Maximum Margin [29]

Regression, Linear Regression, Decision Trees (DT), Random Forests (RF), and Naive Bayes are examples of supervised learning approaches [25].

A. Support vector machine

SVM is a supervised learning methodology based on the classification approach. It can be considered a non-probabilistic linear classifier. SVM is regarded as the state-of-the-art machine learning algorithm. Margin calculation is the core concept underlying SVM. In such an approach, each item of data is represented in n-dimensional space as a point, where n is the features count and each feature represents the value of the coordinate. As depicted in Fig. 5, the objective of this strategy is to examine the vectorized data as well as create a hyperplane that distinguishes between the two classes [27]. Various margins are then drawn between several classes, and a hyperplane is built that minimizes the mean-squared error and maximizes the margin-to-class distance [28].

Once an optimal separating hyperplane is identified in the case of linearly separable data, points of data that sit on its boundary are called support vector points, and the solution is introduced as a linear combination of these points alone, as depicted in Fig. 6. The other data values are disregarded [29]. Therefore, the SVM model's complexity is independent of the feature count found in the training data. So, SVMs are ideally suitable for learning missions involving many features relative to the number of training cases.

Despite the greatest margin that enables the SVM to choose through numerous nominee hyperplanes, SVM may be unable to locate any hyperplane that can separate hyperplanes at all due to the misclassified instances contained in the data. One proposed solution to this problem is to utilize a soft margin that allows certain training cases misclassifications [30]. SVMs are binary classifiers, so in the case of multi-class problems, the problem needs to be reduced to a series of several binary classification problems. Categorical data represent another challenge; however, with adequate rescaling, decent results can be obtained [29].

B. K-nearest neighbors

KNN is considered a nonparametric classification technique that makes no assumptions about the basic dataset and is known for its efficiency and simplicity. In KNN, a labeled training dataset is used to predict the class of unlabeled data [31]. KNN is typically employed as a classifier to classify data based on the nearest or most nearby training samples in a specific location. KNN is utilized in datasets where data may be divided into distinct clusters to determine the new input's class. KNN is more significant in case there is no prior knowledge of the data used in the study [31].

KNN typically employs K variable values between 0 and 1 to calculate the number of training data points with the closest distance. KNN employs numerous distance functions, including Manhattan distance, Euclidean distance, Minkowski distance, and Hamming distance. The Euclidean distance is employed to calculate its nearest neighbors in the case of continuous data, but for categorical data, the Hamming distance function is utilized [32].

The most challenging aspect of the KNN algorithm is choosing the K value, as it affects the algorithm's performance and precision. Small K values generate noise in class label prediction, while large K values may lead to excess fitting likelihood. In addition, it increases the computation time and affects the execution speed. The K value is calculated according to (1):

$$K = n^\wedge(1/2) \tag{1}$$

where n is the size of the dataset.

Cross-validation will be applied to training data with varied K values to maximize the test results. The optimal value for test results will be decided based on the optimal precision [32].

The KNN technique has the following benefits: it is a straightforward technique that is simple to apply. It is a very adaptable classification technique that is ideal for multimodal classes.

On the other hand, using the KNN algorithm to classify unknown data is quite costly. It needs to calculate the distance between the k-nearest neighbors. As the size of the training set increases, algorithm computations get increasingly intensive. Noisy or irrelevant characteristics will decrease accuracy. Moreover, KNN does no generalization on the training data and retains them all. Consequently, greater dimensional data will reduce the precision of areas. It computes the distance between k neighbors, so KNN is a lazy learner [33].

C. Logistic regression

Logistic regression is a supervised learning approach used to differentiate between two or more groups [27]. It provides, in terms of 0 and 1, the likelihood that an event will occur based on the values of the input variables (i.e., it gives the binomial outcome). For instance, predicting whether or not an e-mail is categorized as spam is a binomial result of Logistic Regression. In addition, Logistic Regression can produce multinomial outcomes, such as predicting the preferred cuisine (Chinese, Italian, Mexican, etc.). In addition, Logistic Regression can produce ordinal results, such as rating a product from 1 to 5. Therefore, Logistic Regression is concerned with categorical target variable prediction [33]. Logistic Regression provides several benefits, including ease of implementation, computational efficiency, training efficiency, and regularization simplicity. In Logistic Regression, input features do not require scaling. In addition, Logistic Regression is immune to data noise and multi-collinearity. Logistic Regression, on the other hand, is unsuitable for nonlinear problems since its decision surface is linear, and sensitive to overfitting, and all independent variables must be recognized for it to work successfully [33].

D. Linear Regression.

Regression is an example of a supervised learning technique in which the value of the output variable is decided by the values of the input variable and the utilized labeled datasets. Regression can be used to model and predict continuous variables. In linear regression, an attempt is made to fit a straight hyperplane to the data set if the relationship between the variables of a dataset is linear [33]. Linear Regression is calculated according to (2) [32]:

$$F(x) = mx + b + e \tag{2}$$

where x is the independent variable, $F(x)$ is the dependent variable, m is the slope of the line, b is the y-intercept, and e is the error term.

The best prediction accuracy may be achieved using the Linear Regression algorithm if the following steps are followed to prepare the training data [32]:

- Assume that the dependent and independent variables are linear, i.e., apply any of the available data transformation techniques to make the data linear.
- Remove noisy data and outliers using a technique for cleaning data.
- To minimize overfitting, do pair-wise correlation and exclude the most linked variables.
- Apply Gaussian distribution to the training data to generate more accurate predictions.
- Rescale inputs to improve the reliability of the prediction.

From the above discussion, it is clear that the Linear Regression algorithm is straightforward to comprehend. In addition, the ideal linear relationship between dependent and independent variables is demonstrated. In contrast, Linear Regression can only predict the numeric output. It is inappropriate for nonlinear data and highly sensitive to outliers. Also, data must be independent [32].

E. Decision trees

Classifier-generating systems are one of the most popular strategies in data mining [34]. In data mining, classification algorithms are capable of processing vast quantities of data. It can be used to create assumptions about categorical class names, categorize information based on training sets and class labels, and classify newly accessible data [35].

DTs are one of the powerful approaches utilized in numerous domains, including ML, image processing, and pattern recognition [36]. DT is a model that sequentially as well as cohesively combines a set of basic tests in which a numerical characteristic is compared with a threshold value [37]. In addition, DT is a common classification model in Data Mining [38]. Every tree is composed of nodes and branches. Each node represents an attribute inside a group to be categorized, and each branch provides a possible value for the node [39]. Figure 7 illustrates the structure of DT.

DT algorithm is a supervised learning algorithm. It tries to build a training model that may be used to predict the class or value of target variables by employing learning decision rules learned from the training data [41].

The advantages and disadvantages of using the DT algorithm to solve regression and classification problems [42–44] are outlined in Table 1.

F. Random forest

RF is an ensemble classifier since it employs many DTs to compensate for the shortcomings of a single DT [45–49]. The 'vote' of all trees is utilized to determine the final class for each unknown. This eliminates the possibility that a single tree may not be ideal. Therefore, adding numerous trees should result in a global optimum [50]. For the formation of each tree in the "forest", the bootstrap approach is used for resampling. In addition, on each node split, a subset of features is randomly selected, and the split variable selection occurs over this subset. The projected value for classification is the majority vote, and the average, for regressions [51–54]. On RF models, there are two parameters for tuning: *mtry*, which is the number of features that are randomly picked to consider in each split; and *ntree*, which is the trees count in the model. The *mtry* parameter has a tradeoff: large values increase the correlation among trees but improve

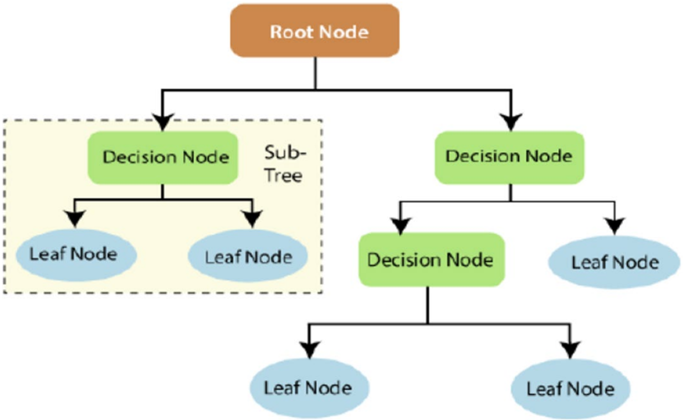

Fig. 7 DT Structure [40]

Table 1 DT benefits and drawbacks

Benefits	Drawbacks
1—Easy to understand 2—Rapidly transformed into a set of production principles 3—Capable of classifying categorical and numerical outcomes, but only capable of generating categorical attributes 4—There are no a priori hypotheses on the validity of the outcomes	1—The ideal decision-making system can be thwarted, resulting in wrong decisions 2—The decision tree has multiple levels, which makes it confusing 3—The complexity of the decision tree's calculations may increase as training samples increase

the accuracy of each tree [51]. The unused elements are called the Out of Bag (OOB) samples, which can be employed for validation in this case, each tree predicts over its OOB samples, and the final result is an average over the outcomes of the trees [55].

There are two options for estimating the relevance of each variable and ranking them accordingly. The initial choice is to utilize the OOB samples. In this option, the accuracy is calculated over the set of each tree and its corresponding OOB samples, a variable is randomly permuted among samples, and the accuracy is recalculated on the new set. Applying this to the set of all trees and average for each variable yields a metric for comparing relevance. This metric for comparison is known as the Permutation Importance Index (PIM) or Variable Importance Measure (VIM). The alternative is to calculate the split improvement for each tree and node using a measure (e.g., the Gini Index) and use these values to compare the significance of the variables [55].

RFs offer high flexibility and prediction rates. It also does not overfit the data when the number of trees is considered. Alternatively, a graphical representation is not feasible as in DTs [55].

G. Naïve Bayes

The Naïve Bayes technique, also known as the Bayes of Idiots, Bayes of Freedom, or basic Bayes, is a fundamental probability-band classifier. Provided the class variable, it is supposed that the existence or absence of a particular class feature has no significance on the existence or absence of any other class feature [56].

The Naïve Bayes technique is straightforward to implement since it does not require complex recursive parameter estimation systems. Consequently, a naive Bayes classifier can be useful for enormous datasets. Also, it requires minimal training data to assess the restrictions. As independent variables rather than the whole matrix of covariance are assumed, only the variances of the variables within each class must be estimated [56].

Unsupervised learning

In unsupervised learning, there is no output label information contained in the dataset. The purpose of these models is to infer the link between data and/or to uncover hidden variables [25]. These strategies are mostly used to reduce the size of a dataset by extracting key features. Reducing the number of features helps prevent problems such as high computational cost and multi-collinearity [57]. Figure 8 depicts unsupervised learning, in which the machine guesses the result according to past experiences

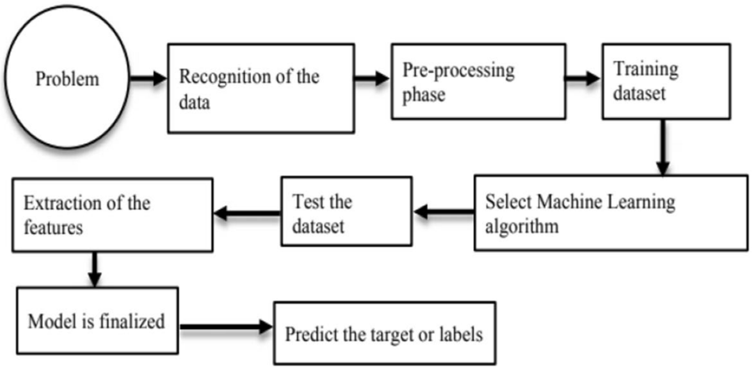

Fig. 8 Unsupervised learning workflow [28]

and learns from information previously provided to anticipate the real-valued outcome. Examples of unsupervised learning-based methods are K-Means Clustering, Principal Component Analysis (PCA), and Latent Dirichlet Allocation (LDA) [25].

A. K-means clustering

K-means clustering is one of the unsupervised learning methods that automatically produces groups or clusters. Data with comparable properties are put into the same cluster. K-means is the name of the method as it forms K different groups [28]. The purpose of the K-means clustering is twofold: (1) to provide K-centroids, one for each cluster, and (2) to minimize the square error function. The mean value is placed in the middle of the cluster [27].

The k-means clustering technique has many advantages. First, it is computationally more effective than hierarchical clustering for enormous variables. Second, it yields tighter clusters than hierarchical clustering with global clusters and small *k*. Finally, ease in implementation and comprehension of the clustering results. The order of complexity of the algorithm is O(K*n*d), so it is computationally efficient [33].

On the other hand, the *K* value is not known and its prediction is complex. Degrades in performance occur when clusters are global and when different beginning partitions result in distinct final clusters. Also, when there is a difference in the size and density of the clusters in the input data, the performance decreases. In addition, the joint distribution of characteristics inside each cluster is spherical (spherical assumption) and cannot be achieved as the correlation between features break it and put extra weights on connected features. K-Means clustering can be susceptible to outliers. Also, it is sensitive to local ideal and initial points, and a unique solution for a specific K value does not exist—so K means needs to be run for a K value lots of times (20-100times) and then, pick the results with the lowest J [33].

B. Principal component analysis

PCA is an unsupervised ML approach that reduces the dimension of the data. Therefore, the computations are more efficient and quicker [27]. The two-dimensional data in PCA are turned into one-dimensional data by transforming the collection of variables into new ones called principal components (PC) which are orthogonal. The data set of the PCA algorithm must be scaled because the results are sensitive to the relative scaling [28].

To explain the PCA mechanism, let us use an example of 2D data. When the 2D data are plotted on a graph, it takes up two axes. Applying PCA to this data will turn it into 1D [27], as illustrated in Fig. 9.

C. Latent Dirichlet allocation

LDA is a statistics-based data mining technique that differentiates between classes of objects in N-dimensional feature space by computing a sequence of $k \leq N-1$ linear discriminant whose values can be used to describe the classes [59]. LDA and PCA are similar [60] in that they describe the "most important" variations in the data and select directions that maximize feature variance. LDA differs from PCA in that LDA makes use of the class labels: it selects directions that can best differentiate the class means relative to the sum of the class variances along that direction. It maximizes the ratio of between-class scatters to within-class scatters. Intuitively, it detects lower-dimensional descriptions of the data which push the class members together and pull members of different classes out [61]. The k linear discriminants that correspond to the eigenvectors are arranged by eigenvalue. The discriminants can be used to group new objects or for dimension reduction [61].

To ensure the discriminant's optimality, the LDA's design makes the following two assumptions: (1) the linear combination of any characteristics is normally distributed, and (2) the classes have equal covariance matrices. Despite the danger of inferior outcomes, LDA has been utilized routinely for dimension reduction and classification when these assumptions are broken [61].

Reinforced learning

Unlike supervised and unsupervised learning, RL is a goal-oriented learning approach. Learning occurs via reacting to the surrounding environments and detecting status changes. RL is strongly tied to an agent (controller) responsible for the learning process

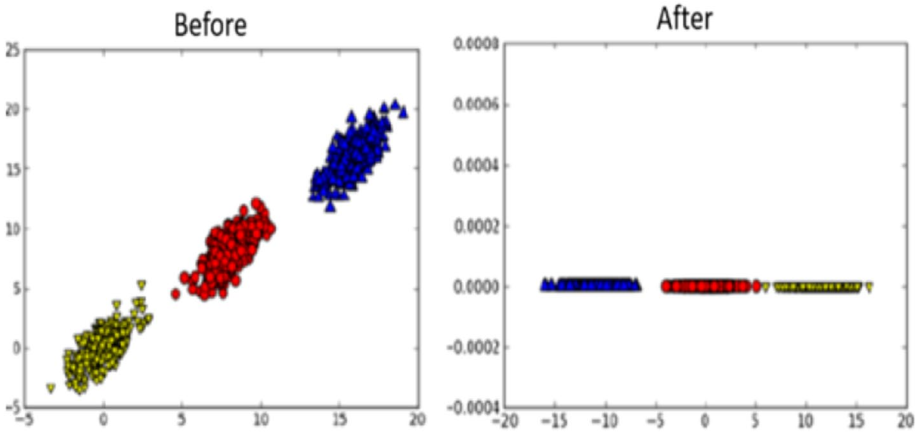

Fig. 9 Data visualization before and after applying PCA [58]

to attain a goal. In particular, the agent takes actions (control signals) and consequently, the status of the environment is changed and rewards, which are special numerical values, are returned either positive or negative. The agent aims to maximize the rewards obtained over time. A task is a full specification of an environment, which determines how the reward is generated [62]. Examples of RL-based techniques are Q-Learning Algorithm and Monte Carlo Tree Search (MCTS).

A. Q-learning

Q-learning [63] is a straightforward way that enables agents to learn how to act optimally in controlled Markovian domains. It represents an incremental approach to dynamic programming which imposes low processing demands. It works by boosting successively its ratings of the quality of specific acts at certain states. It can also be considered an asynchronous Dynamic Programming (DP) approach. It provides agents with the possibility of learning to act optimally in Markovian domains by experiencing the consequences of actions, without requiring them to generate maps of the domains [64].

Q-learning is applied in information theory, and related investigations are underway. Recently, Q-learning and information theory have been applied to various disciplines such as natural language processing, anomaly detection, pattern recognition, and image classification [65–68]. In addition, a framework has been established to provide a satisfying response based on the user's speech using RL in a voice interaction system [69], and a high-resolution prediction system for local rainfall based on DL has been developed [70].

The advantage of the ant Q-learning approach is that it can identify the value of the reward for a specific activity in a multi-agent environment successfully due to the corporation between agents. The drawback of ant Q-learning is that its result can be stuck at a local minimum when agents take just the shortest path [71].

B. Monte Carlo tree search

MCTS is a powerful technique for handling sequential decision problems. The plan relies on a smart tree search that balances exploration and exploitation. Random sampling is employed in MCTS in the form of simulations to save statistics of activities and make more knowledgeable selections in each future iteration [72]. MCTS is a decision-making technique that is utilized in scanning huge combinatorial spaces represented by trees. In such trees, nodes represent states, also referred to as configurations of the problem, whereas edges denote transitions (actions) between states [72].

Formally, MCTS is directly applied to issues that can be described by a Markov Decision Process (MDP). Certain modifications of MCTS make it possible to be applied to Partially Observable Markov Decision Processes (POMDP) [73]. More recently, MCTS paired with deep RL are considered the backbone of AlphaGo developed by Google DeepMind which is documented in [74].

The basic MCTS procedure is conceptually so simple [75], as depicted in Fig. 10. A tree is created in an incremental and unbalanced method. In each iteration, a tree policy is utilized to get the most urgent node of the current tree.

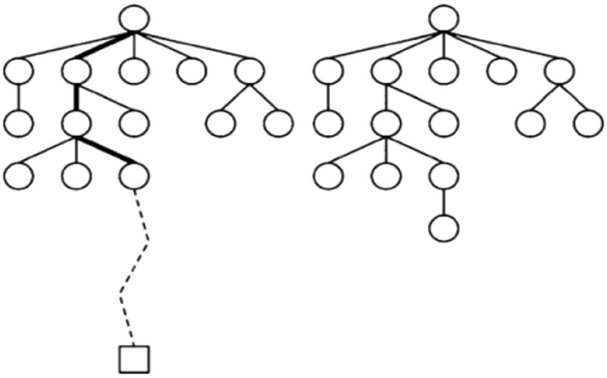

Fig. 10 The basic MCTS process [76]

The tree policy aims to balance the considerations of exploration and exploitation. A simulation is then run from the specified node, and the search tree result is accordingly updated. This involves the insertion of a child node that matches the action taken from the selected node and an update of the statistics of its ancestors. Based on some default policy, moves are being conducted during this simulation which in the simplest scenario aims to make uniform random moves. A notable advantage of MCTS is there is no need for the values of the intermediate states to be evaluated, which extremely minimizes the amount of domain knowledge required [75].

Deep learning

About a decade ago, Deep Learning (DL) emerged as an effective ML technique and achieved good performance in several application fields. The core idea of DL approaches is to learn complicated characteristics extracted from data with low external contribution using Deep Neural Networks (DNN) [77]. These algorithms do not require to be manually provided created features; they automatically learn additional complicated features [78].

DL is an AI paradigm that has gained major interest from the academic community and demonstrated higher potential over conventional methods [79]. DL is a more efficient, monitored, time-consuming, and cost-effective technique than the ML technique. Not only it is a specific approach to knowledge, but also it adapts to various methodologies and topographies that could be beneficial to a wide range of complicated problems. The approach learns the illustrative and differential properties in a relatively varied method [80, 81]. Figure 11 demonstrates the procedure of ML and DL.

To generate high-level abstractions with many nonlinear transformations, DL is based on a collection of ML techniques used to model data. The artificial neural network (ANN) system runs on a DL technology [82, 83]. These networks include many layers for collecting high-level characteristics and for eliminating problematic data, so the performance of DL algorithms is higher than ML algorithms [84].

ML approaches have brought a huge impact on our daily life such as efficient web search, self-driving vehicles, computer vision, and optical character recognition. Also, by implementing ML approaches, the human-level AI has been improved as well [85–87].

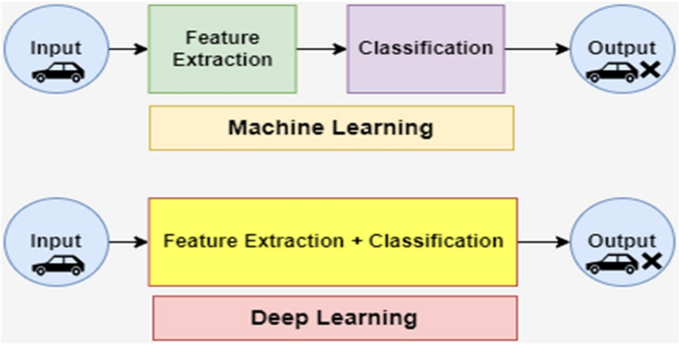

Fig. 11 ML versus DL

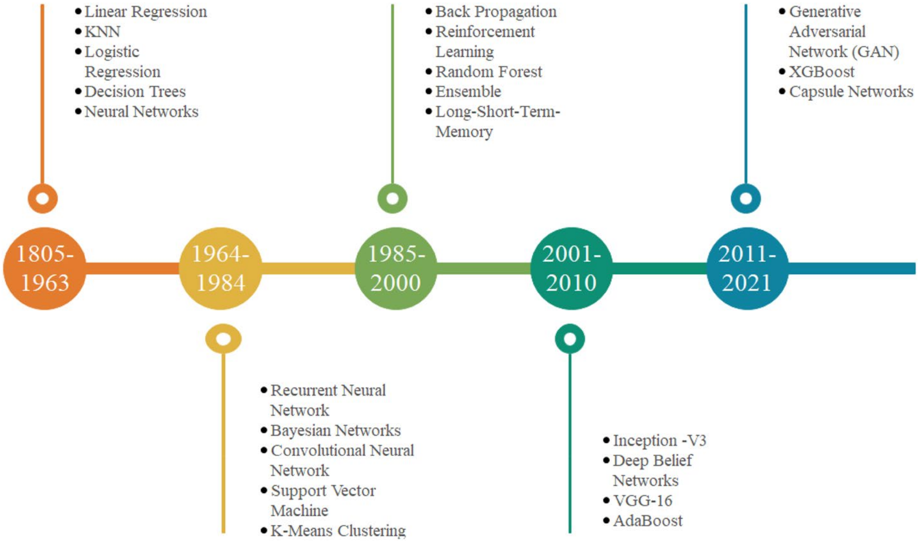

Fig. 12 ML and DL algorithms development timeline [26]

Nevertheless, the performance of classic ML algorithms is far from ideal when it comes to human information processing mechanisms (e.g., voice and vision). The DL algorithms concept was formed in the late twentieth century inspired by deep hierarchical structures of human speech perception and production systems. Figure 12 displays a timeline showing the evolution of deep models along with the classic model [26]. DL has many architectures. Examples of such architectures are CNN, RNN, LSTM, and Recurrent CNN (RCNN).

A. Convolutional neural network

CNNs are a subtype of ANNs and are frequently utilized in face recognition, text analysis, human organ localization, and biological image recognition [88]. CNN structure was first introduced in 1988 by Fukushima [89]. It was not widely employed, however, due to restrictions of computation gear for training the network. In the 1990s, LeCun et al. [90] adapted a gradient-based learning algorithm to CNNs and provided successful results

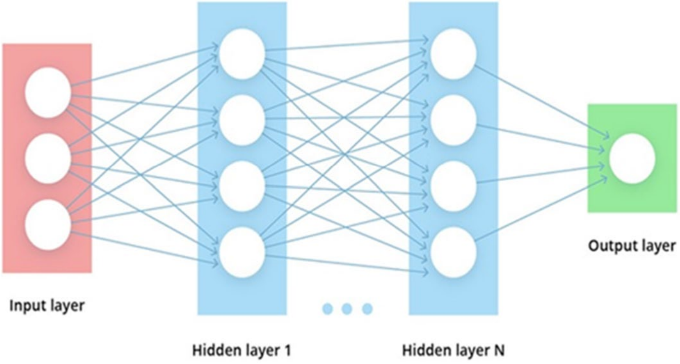

Fig. 13 A CNN architecture

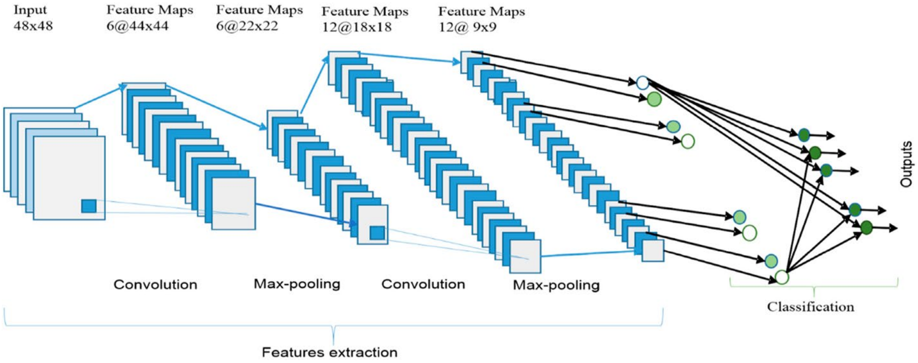

Fig. 14 Feature extractors and classifier parts of CNN [92]

for the handwritten digit classification problem. After that, researchers progressively enhanced CNNs and reported state-of-the-art results in different recognition tasks.

A CNN architecture includes three components: the input layer, hidden layer, and output layer. The intermediate levels of any feedforward network are known as hidden layers, and their number varies based on the network architecture type. Convolutions are executed in the hidden layers, which include dot products of the convolution kernel with the input matrix. Each convolutional layer generates feature maps to be used as input by the subsequent layers [91], as shown in Fig. 13.

In general, CNNs consist of two major components: Feature extractors and a classifier, as shown in Fig. 14. In the feature extraction layers, each layer of the network takes as its input the output of its immediate previous layer and transmits its output to be the input to the next layer. The CNN design involves a combination of three types of layers: Convolution, max-pooling, and classification. In the low and middle level of the network, there are two types of layers: Convolutional layers and max-pooling layers. Convolutions are the even-numbered layers, whereas the odd-numbered layers are for max-pooling operations. The output nodes of the convolution and max-pooling layers are then arranged into a 2D plane named feature mapping. Usually, the plane of each layer is produced by the combination of one or more planes of the previous levels. The nodes of a plane are connected to a small section of each connected plane of the previous layer.

Each node of the convolution layer extracts the features from the inputs by convolution operations on the input nodes. As the features propagate to the highest level, the dimensions of the features are lowered based on the kernel size of the convolutional and max-pooling processes correspondingly.

For ensuring classification accuracy, the number of feature maps is increased for expressing better features of the input. The output of the last CNN layer is used as the input to a fully connected network called the categorization layer. In the classification layer, the extracted features are used as inputs concerning the size of the weight matrix of the final neural network. At the topmost classification layer, and using a soft-max layer, the score of the respective class is calculated. According to the highest score, the classifier produces output for the corresponding classifications [92].

CNNs have various advantages including being more like the human visual processing system, having a highly optimized structure for processing 2D and 3D images, and being effective in learning and extracting abstractions of 2D information. The max-pooling layer of CNNs is successful, particularly at absorbing shape variations. Furthermore, CNNs contain much fewer parameters than a fully connected network of the same size as it is constructed of sparse connections with coupled weights. In addition, CNNs are trained with the gradient-based learning technique that suffers less from the diminishing gradient problem. Given that the gradient-based technique trains the full network to reduce an error criterion directly, CNNs can generate highly optimized weights [92].

B. Recurrent neural network

Developed in the 1980s, RNN is one of the most widely used DL models [93]. These kinds of networks have a memory that stores the information they have seen so far and have various types. Moreover, RNNs are powerful models for time series analysis, and they use the prior output to predict the next output. In this situation, the networks themselves contain repeating loops in the hidden layers, which allow the storing of previous input information for a while, so that the system can predict future outputs. The output of the hidden layer is retransmitted t times to the hidden layer. The output of a recursive layer is only sent to the next layer when the number of iterations is completed. In such a circumstance, the output is more global, and the preceding knowledge is maintained for longer. Finally, the errors are returned backward to update the weights [94]. RNN is employed mostly in the fields of speech processing and Nature Language Processing (NLP) settings [95, 96].

Unlike CNN, RNN employs sequential data in the network. As the embedded structure in the data sequence gives useful information, this property is fundamental to a range of various applications such as NLP. Thus, RNN can be considered as a unit of short-term memory, where x is the input layer, y is the output layer, and s represents the state (hidden) layer [97]. For a specific sequence of input, a typical unfolded RNN diagram is presented in Fig. 15. In addition, a deep RNN was introduced to minimize the learning difficulty in the deep network and brings the benefits of a deeper RNN depending on three different deep RNN techniques, namely "Hidden-to-Hidden", "Hidden-to-Output", and "Input-to-Hidden" introduced by Pascanu et al. [98].

One of the main challenges with RNN is its sensitivity to the expanding gradient and vanishing problems [99]. More specifically, the reduplications of many large or small

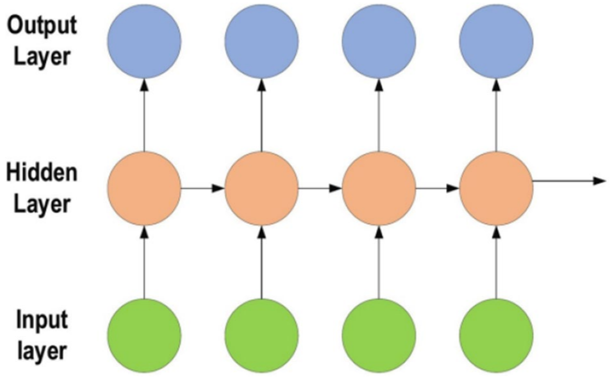

Fig. 15 A typical unfolded RNN diagram [97]

derivatives during the training phase may cause exponentially explode or decay of the gradients. With the introduction of new inputs, the network stops thinking about the original ones; hence, its sensitivity decays over time [97].

C. Long short-term memory

LSTM is a special case of RNN as it has internal memory and multiplicative gates. The diversity of LSTM cell layouts has been described in 1997 when the first LSTM was launched [100]. LSTM contributed to the development of well-known services like Siri, Cortana, Alexa, Google Translate, and Google voice assistant [101]. LSTM is a module in an RNN network that addresses missing gradient problems. Generally, RNN employs the LSTM network to avoid propagation errors. This allows the RNN to learn across multiple time steps. LSTM includes cells that keep information outside of a recurrent network. Like the memory in a computer, the cell is deciding when the data have to be stored, written, read, or erased using the gate [102]. A simple RNN cell depicted in Fig. 16(a) was enhanced by adding a memory block which is controlled by input and output multiplicative gates. Figure 16b shows the LSTM architecture of the jth cell c_j. The main component of a memory block is the self-connected linear unit s_c termed constant error carousel (CEC) which protects LSTM from the drawbacks of regular RNN. An input gate and output gates consist of corresponding weight matrices and activation functions [101].

Generally, it can be concluded that the LSTM cell comprises one input layer, one output layer, and one self-connected hidden layer. The hidden layer may contain 'conventional' units that can be fed into the next LSTM cells. However, a conventional LSTM cell also met some limits due to a linear form of s_c. It was specified that its steady expansion may induce saturation of the function hand converted into an ordinary unit. Therefore, an additional forget gate layer was inserted [103], as illustrated in Fig. 16(b), which permits undesirable information to be wiped and forgotten.

Bidirectional LSTM, Hierarchical LSTM, Convolutional LSTM, Grid LSTM, LSTM Autoencoder, and Cross-modal LSTM are the most advanced network topologies that use the LSTM gating mechanism [104].

Bidirectional LSTM type networks send and receive the state vector in both directions. As a result, bi-directional time dependencies are taken into account. As a result of reverse state propagation, future expected correlations can be included in

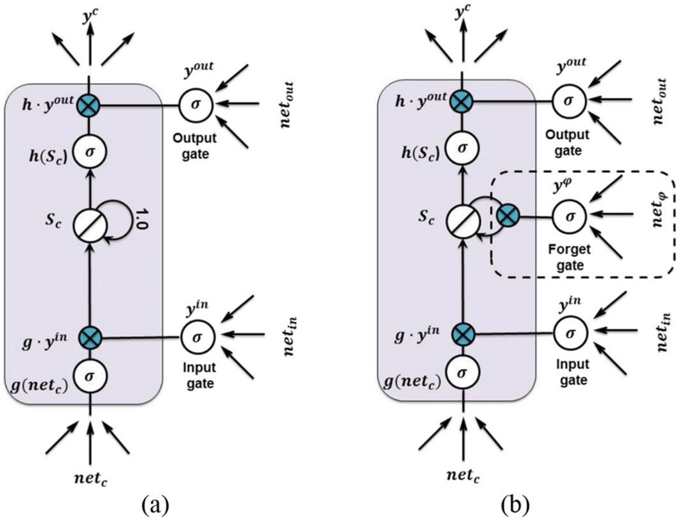

Fig. 16 **a** Original LSTM cell architecture; **b** LSTM cell includes forget gate [101]

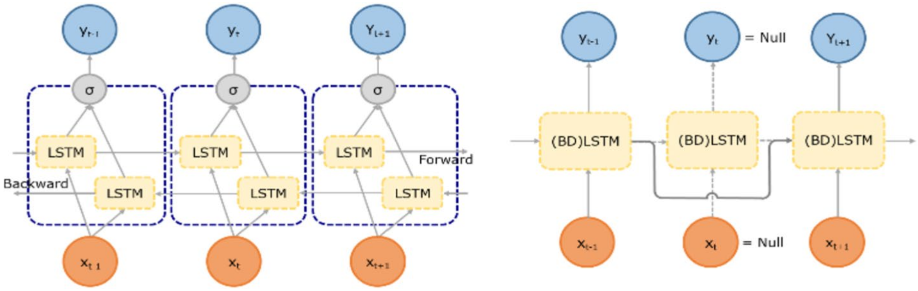

Fig. 17 (Left) Bidirectional LSTM and (right) filter mechanism for processing incomplete data sets [105]

the network's generated outputs. Hence, more time dependencies can be detected, extracted, and resolved using bidirectional LSTM networks more precisely than unidirectional LSTM networks. LSTM networks can encapsulate geographically and temporally dispersed information and harmonize partial data using a flexible connection mechanism for the propagation of the cell state vector [105]. Based on the data gaps discovered, this filter method redefines the connections between cells. Figure 17 depicts the architecture of Bidirectional LSTM.

Hierarchical LSTM networks resolve multidimensional problems by splitting the overall problem into sub-problems and hierarchically structuring them. This is achieved by adjusting weights inside the network which obtains the power to produce a specific degree of attention.

Using a weighting-based attention mechanism that handles and filters input sequences, hierarchical LSTM networks could be utilized to predict long-term dependencies [106]. Convolution LSTM can be used to filter and reduce input information obtained over a longer time period using convolution operations built into

LSTM networks or directly into the LSTM cell structure. Convolution methods that are directly incorporated into the cell can also be used to extend the usual LSTM cell. Correlations are extracted by convolving current input sequences, recurrent output sequences, and weight matrices. The newly created features are received as new inputs by the network gates [107]. Figure 18 depicts this strategy.

Moreover, convolutional LSTM networks are considered ideal for expressing a wide range of quantities, including spatially and temporally distributed relations. Nevertheless, as a reduced feature representation, various values can be collectively forecasted alone. Layers' deconvolving must predict different output quantities based on their original units rather than as features [104]. An autoencoder structure is commonly used to realize information deconvolution and convolving. A layered LSTM autoencoder handles the challenge of high dimensional input data and the forecasting of high dimensional parameter spaces in [108]. In [109], a method for directly integrating an autoencoder into the LSTM cell structure was proposed. This multimodal prediction approach was proposed by extending LSTM. To compress input data as well as cell states, encoders and decoders were integrated directly into the LSTM cell structure. This optimization maximizes information flow in the cell and leads to an enhanced cell state update mechanism for both short-term and long-term dependencies.

Grid LSTM is an LSTM cell with a matrix structure [110]. The Grid LSTM has connections for the input sequences' spatial and temporal dimensions. As a result, connections in various dimensions within cells extend the normal information flow. As a result, the Grid LSTM is appropriate for the parallel prediction of a wide range of output quantities that can be either linearly independent or nonlinearly dependent. Figure 19 compares a two-dimensional Grid LSTM network to a standard stacked LSTM network [110].

Cross-modal LSTM is a modern method for predicting various quantities collaboratively. It combines a number of regular LSTMs that were previously used to separately simulate the individual quantities. The LSTM flows interact via recurrent connections to handle the quantity dependencies. In other streams, the outputs of defined layers are used as extra inputs for previous and subsequent layers. As a result, a cross-modal prediction can be identified. Figure 20 depicts cross-modal LSTM [111].

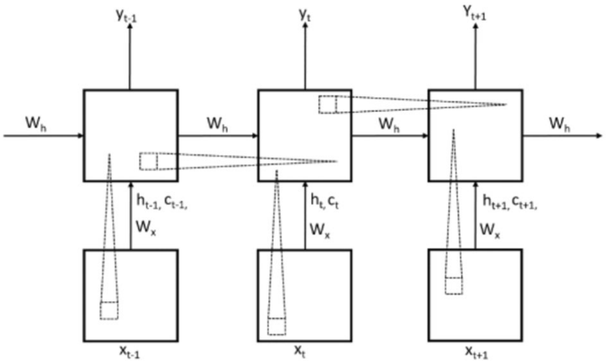

Fig. 18 Convolution operations within LSTM cells [107]

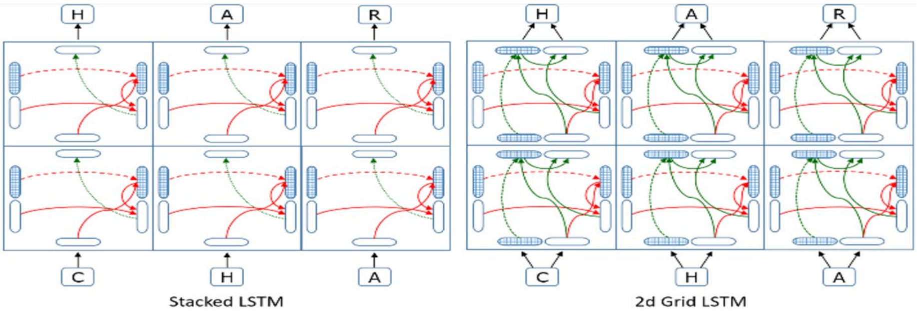

Fig. 19 Grid LSTM (right) versus Stacked LSTM (left) [110]

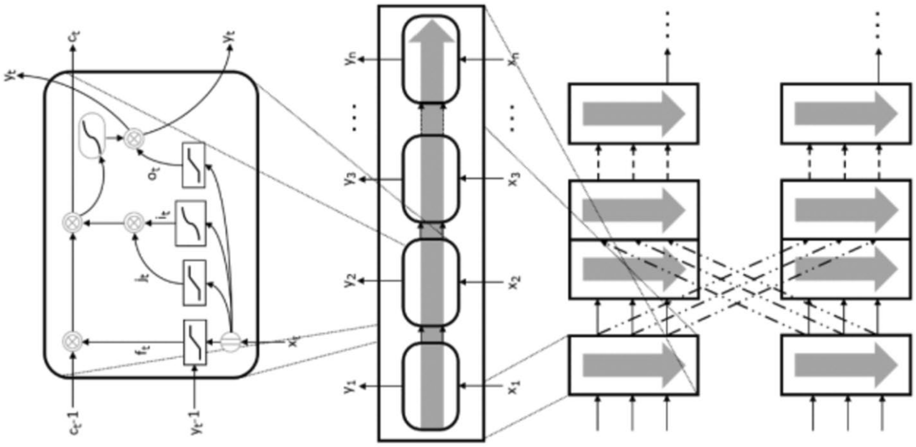

Fig. 20 Cross-modal LSTM [111]

D. Recurrent convolution neural network

In recent years, a new class of CNNs, RCNN, inspired by rich recurrent connections in the visual systems of animals, was introduced. The main component of RCNN is the recurrent convolutional layer (RCL), which integrates recurrent connections across neurons in the normal convolutional layer. With the increasing number of recurrent computations, the receptive fields (RFs) of neurons in RCL expand unboundedly, which is incongruous with biological realities [112]. The traditional RCNN model was proposed in [113, 114]. The RCNN architecture is presented in Fig. 21, in which both feed-forward and recurrent connections have local connectivity and shared weights across distinct locations. This architecture is quite close to the recurrent multilayer perceptron (RMLP) which is generally used for dynamic control [115, 116] (Fig. 21, middle). The main difference is that the full connections in RMLP are replaced by shared local connections, similar to the difference between MLP [117] and CNN.

RCNN integrates a stack of RCLs, optionally interleaved with max-pooling layers, as seen in Fig. 22. Here, layer 1 is the traditional feed-forward convolutional layer without

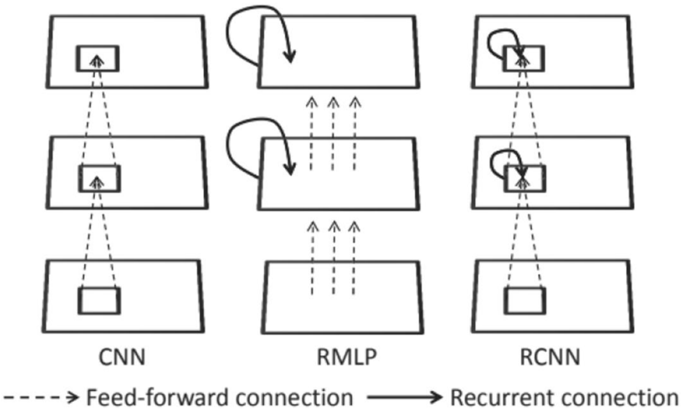

Fig. 21 The CNN, RMLP, and RCNN architectures [113]

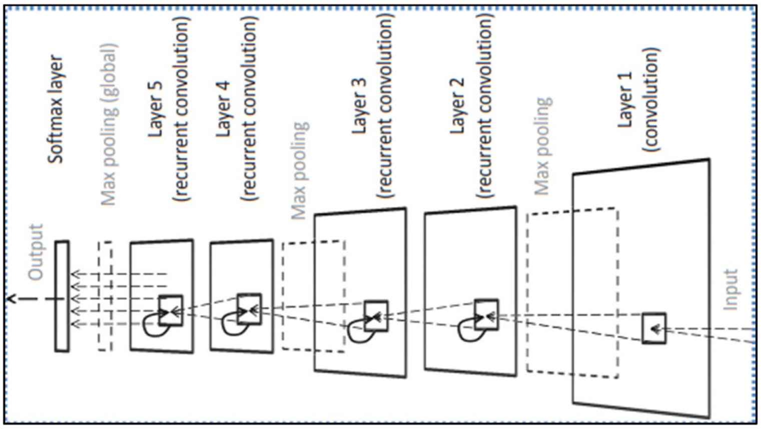
Softmax layer
Max pooling (global)
Layer 5 (recurrent convolution)
Layer 4 (recurrent convolution)
Max pooling
Layer 3 (recurrent convolution)
Layer 2 (recurrent convolution)
Max pooling
Layer 1 (convolution)
Output
Input

Fig. 22 RCNN with one convolutional layer, four RCLs, three max-pooling layers, and one softmax layer [113]

recurrent connections, followed by max pooling. Furthermore, four RCLs are employed with a max-pooling layer in the middle. There are only feed-forward connections among nearby RCLs. Both pooling operations have stride 2 and size 3. The output of the fourth RCL follows a global max-pooling layer, which yields the maximum across every feature map, providing a feature vector describing the image. Finally, a softmax layer is utilized to categorize the feature vectors into C categories. [113].

RCNN has various advantages from the computational perspective. First, the recurrent connections in RCNN allow every unit to include context information in an arbitrarily broad region in the current layer. Second, the recurrent connections improve the depth of the network and at the same time keep the number of changeable parameters constant by sharing weight. This is compatible with the tendency of the current CNN architecture. Third, unfolded RCNN is a CNN with numerous paths from the input layer to the output layer, which facilitate learning. On one hand, the existence of longer paths makes the model capable of learning more complicated features. On the other hand, the existence of shorter paths may improve gradient backpropagation during training [113].

Survey methodology

The articles reviewed in this paper have been published in high-quality conferences and journals of IEEE, Elsevier, Springer, and IOP publishing. Machine learning, deep learning, traffic flow prediction, traffic flow forecasting, traffic speed prediction, short-term traffic prediction, short-term traffic forecasting, and ITS are some of the search terms used to find these articles. The articles examined in this survey are directly relevant to the application of ML and DL approaches in traffic flow prediction. Both empirical and literature reviews on the abovementioned subjects were considered for this work.

Survey organization

This survey compares various forecasting techniques for traffic flow. It follows a dual structure with ML techniques used for traffic flow prediction and DL techniques utilized for traffic flow prediction. This study provides a detailed discussion of the approaches and algorithms which are utilized for predictions, performance measurements, and tools used for these procedures.

The prediction of traffic flow has become one of the primary tasks in the ITS field [118]. Statistical methods, AI, and data mining techniques have been widely employed recently to evaluate road traffic data and anticipate future traffic indicators [119]. Previous findings demonstrated that no single technology could evaluate enormous datasets only by itself. Therefore, according to the data structure and its volume, the proper technology must be applied to extract the best insight from the collected data [120].

ML techniques for traffic flow prediction

In [121], the authors developed an ML-based traffic flow prediction paradigm employing a regression model implemented by several libraries including Pandas, Numpy, OS, Matplotlib, Keras, Sklearn, and Tensorflow. Traffic prediction in this study involves the prediction of next year's traffic data based on previous years' traffic data which eventually offers the accuracy and mean square error. The traffic information was predicated on a basis of 1-hour time gap. Data in this study were acquired from the Kaggle dataset. Two datasets were obtained, in which one is the 2015's traffic data which contains the date, time, number of cars, and number of junctions. The other one is the 2017's traffic data with identical specifications to compare easily without any confusion. This study needs to investigate more aspects that affect traffic flow prediction and employ other prediction approaches like deep learning and big data.

In [122], the authors aimed to address the traffic control problem with the assistance of an ML algorithm to deal with traffic challenges. The authors employed the Q-learning RL technique for managing traffic lights and developed an artificial environment named Simulation of Urban Mobility (SUMO) for simulation purposes. In SUMO, the cars in motion can be watched, the vehicle's delay time can be controlled, and the delay time can be adjusted.

In [123], the aim of this paper was to set the foundation for adaptive traffic control, either by controlling traffic lights remotely or by applying an algorithm that adjusts the timing according to the predicted flow based on the integration of ML (RF, Linear Regression, and Stochastic Gradient Regression) and DL (MLP-NN, RNN) algorithms. The collected findings showed that the proposed ML algorithms had the worst performance.

In [124], the authors concentrated on a critical component of ITSs known as the ability to predict lane changes in vehicular traffic flow. The predictive accuracy to detect changes in lanes was measured using high-fidelity data on vehicular traffic flow gathered by the US Federal Highway Administration (FHWA) for Peachstreet, Atlanta, GA, based on four ML models, namely SVM, NB, RF, and DT. The accuracy and performance measurements revealed that SVM outperforms the other three ML models in terms of precise and accurate prediction of vehicle lane shifts.

In [125], a prediction approach that is based on type-2 fuzzy logic was introduced using the conceptual framework of fuzzy logic and an urban traffic flow time series. The interval type-2 fuzzy system prediction approach was developed, and the Back Propagation (BP) technique was utilized to update the antecedent's coefficients and fuzzy rules' consequent. The effectiveness of the technique proposed in this study was validated using measured data from road networks and compared to other fuzzy approaches. The BP technique and SVM with that type-2 fuzzy logic system have a higher prediction accuracy, according to the testing results.

In [126], the authors investigated the problem of predicting the traffic flow of a road based on historical data. The methodology depended on the decomposition of the canonical polygonal tensor (CP) of the traffic data. This move extracts the normal features of a traffic light on daily and weekly bases in addition to the typical spatial allocation of traffic, while greatly minimizing the amount of data required to represent it. Then, the key elements are extended into the future, and the traffic data are regenerated from the decomposition. The data used here are from the M62 motorway in northern England, from October 1, 2019, to October 28, 2019, at 15-min intervals. These data are reported as the number of passing cars per hour. Using 4 parameters, the prediction captures 90 percent of the signal's power, which exceeds the current rolling average prediction algorithms. The authors indicated that they evaluated 4 variables in traffic flow forecasts but did not mention them.

In [127], the authors developed an intelligent traffic monitoring system based on ML (ML-ITMS) to estimate traffic jams in roadside units to improve ITS performance. A short-term traffic flow ML-based model was developed, and SVM parameters were optimized to enhance traffic flow prediction. In the proposed ML-ITMS, SVM and RF were specifically designed for long-range wide area networks (LoRa) in a single query. The proposed ML-ITMS improved the accuracy estimate for traffic flow and nonparametric processes by using mathematical models. As feedback for the proposed ML-ITMS, a data processing method has been used. The platform was then passed through ML-ITMS services, including public safety and security for cities, medical facility provision, traffic prediction by light and range detection (LIDAR), and parking control. Thus, as the experimental results revealed, the proposed ML-ITMS can improve traffic monitoring to 98.6% and can enhance traffic flow prediction systems better than other existing methods.

In [128], the authors proposed a Gravitational Search Algorithm optimized Extreme Learning Machine, called GSA-ELM. It has been suggested to unleash the performance of short-term traffic flow forecasts. ELM avoids the cumbersome process of BP by defining the best solution analytically. The proposed search technique generally investigates the optimal settings for ELM. The proposed search technique's prediction performance has been measured on four standard data sets by comparing several recent models. The four standard datasets were real-world traffic flow data from the A1, A2, A4, and A8 motorways along the Amsterdam Ring Road. The Mean Absolute Percentage Errors (MAPEs) for the GSA-ELM model on the used data sets are 11.69%, 10.25%, 11.72%, and 12.05%, respectively, while the Root Mean Square Errors (RMSEs) were 287.89, 203.04, 221.39, and 163.24, respectively.

In [120], supervised ML, as a method of Big Data analytics, to forecast various indicators of the traffic volume were examined and conducted through two case studies. In both experiments, for training and testing prediction models, traffic data provided by chosen automatic traffic counters on the roadways in the Republic of Serbia, in the period from 2011 to 2018, were employed.

In [129], the authors proposed reconstructing traffic flows from the expected travel time using an ML method. They examined the capabilities of the Gaussian Process Regressor (GPR) to handle this issue. After obtaining the expected travel time on a specific route, a clustering method shows that travel time profiles in each day can be associated with "different types of the day". Then, various regression factors were trained to estimate traffic flows from the duration of travel. In this study, two situations were studied. In the 'multi-model' variance, the regression factor was trained for each day profile. In the 'Single Model' variation, only one Regressor was trained (the day profile was not considered). The proposed method is a unique method to predict and reconstruct traffic flow in route networks using an ML method from aggregated floating vehicle data (FCD). Two main problems can be identified from this work. The first relates to using non-dispersed algorithms on the input data which can be problematic with longer evaluation sequences, producing a more complex trained model. The other problem is a traditional issue of every ML solution, and it has to do with the dependence on the quality of the input data.

In [130], a hybrid model incorporating ELM and ensemble-based technologies was developed to predict the future hourly traffic on a road section in Tangiers, a city in northern Morocco. The suggested model was built based on a high-speed ML technology that uses a kind of Single-Layer Feed-forward Neural Network (SLFN). The data set in this study was a set of traffic flow recorded over 5 years from 2013 to 2017 from the Moroccan Center for Road Studies and Research. This study needs to consider additional relevant information related to traffic, such as special events, weather conditions, and traffic characteristics on adjacent roads that may affect a particular road.

In [131], the power of various ML techniques was investigated to predict traffic conditions. Preliminary data were collected over two weeks of monitoring in Bandung, Indonesia, to be capable of determining future traffic conditions. The collected features used in the dataset are days, hours, origins, destinations, route view, traffic conditions, weather, and weather locations. The study investigated neural networks, NB, DT, SVM, DNN, and DL. There are two main issues in this work. First, the size of the training data was very small. Second, the change in the training data means that the training process must be reapplied to reflect the newer data set, which takes additional time.

In [132], the prediction accuracy of four ML models was examined using probe data gathered from the road network of Thessaloniki, Greece. The utilized ML models were RF, Support Vector Regression (SVR), Multilayer Perceptron (MLP), and Multiple Linear Regression (MLR). There are two key concerns in this work. First, it has low accuracy in real-time speed prediction. Second, it needs to be tested on different datasets.

In [7], the authors suggested a preliminary method for assessing a realistic data set of road traffic accidents utilizing graphical representations and dimension reduction methods. The data set was subjected to PCA analysis and linear discrimination, and the resulting performance measures provided some comprehensive insights into the patterns of road traffic accidents. The authors developed the preliminary framework by utilizing dimensionality reduction techniques on realistic road traffic accident data from Gauteng Province, South Africa (SA). Furthermore, classification was carried out using the NB, Logistic regression, and K-NN methods. The processed data were post-processed, and model performance measures, precision, and RMSE were used to evaluate each classifier.

In [133], the authors introduced a novel framework for stepwise regression in an idea-drift environment, with ensemble learning as the primary solution for modernizing distribution representation. The regression problem for predicting traffic volume was first converted into a binary classification problem. Second, the Regression to Classification (R2C) method was used to create a more precise classification-type loss function for ensemble learning. Finally, the regression function's incremental learning was modeled as an incremental update to the hyper-resolution level. The proposed R2C architecture for motion volume prediction has the disadvantage of not accounting for motion volume spatial dependencies.

To summarize all previous related works, Table 2 compares among them in terms of methodology, data set, approaches, and their main findings.

DL techniques for traffic flow prediction

In [134], it was proposed to construct a traffic prediction system using four DL approaches namely: Deep Autoencoder (DAN), DBN, RF, and LSTM. This technique is mostly used to estimate the traffic flow in more populated locations. The essential parameters used in this study were zone type, weather condition, day, road capacity, and vehicle types. There is no mention of the used dataset in this work.

In [135], the major objective was to predict trip duration from point A to point B on a route using neural networks. Several DL and neural network algorithms were utilized such as the color clustering algorithm (K-Means algorithm) combined with several parameters to compute and estimate travel duration. The dataset utilized in this study was obtained using Waze Live Map APIs. The authors need to examine other factors such as weather conditions to boost the efficiency and reliability of their job.

In [136], a short-term strategy for traffic flow forecasting based on a recurrent mixture density network, which is a mix of RNN and mixture density network (MDN), was proposed. Traffic flow data generated by sensors placed on road networks in Shenzhen, China, were used as the data set used in this study. It was divided into two periods: from January 1, 2019, to March 31, 2019, and from October 1, 2019, to December 21, 2019. The modest size of the data set used is a critical issue in this study.

Table 2 A comprehensive comparative study of the previous works

Study	Methodology	Dataset	Approaches	Findings
[121]	The authors built a ML algorithm to anticipate next year's traffic depending on data from the previous year utilizing libraries like Numpy, Pandas, Matplotlib, OS, Sklearn, Keras, and TensorFlow	Two Kaggle datasets. The date, time, number of vehicles, and number of intersections are all included in one set of data from 2016. The other is the 2017 traffic data which is identical in every respect to the 2016 data	Regression Model	More factors that affect traffic management should be considered
[122]	The authors aimed to simulate traffic signals using a new environment they proposed and referred to as the Simulation of Urban Mobility (SUMO)	N/A	Q-learning RL method	In SUMO, the vehicles that are currently in motion can have their delay times monitored, controlled, and changed
[123]	The main goal was to propose a mechanism for adaptive traffic control	A Road Traffic Prediction Dataset from the Huawei Munich Research Center	Linear Regression, MLP Regressor, Gradient Boosting Regressor, RF Regressor, and Stochastic Gradient Descendent Regressor	The stochastic gradient had an R^2 value of 0.9, an Explained Variance (EV) value of 0.9, an MAE value of 12.8, a MAPE value of 29%, and an RMSE value of 18
[124]	The authors focused on a key feature of ITSs, the system's capacity to predict lane changes in traffic flows	The U.S. Federal Highway Administration (FHWA) used a program named "Next-Generation Simulation (NGSIM)" to gather a high-fidelity vehicle flow dataset in 2005	SVM, NB, RF, and DT	SVM has the highest accuracy when predicting when a car would switch lanes
[125]	The authors presented a novel fuzzy logic-based framework with a time series analysis for urban traffic volumes forecasting	Data on transportation networks	Interval type-2 fuzzy logic, BP, and SVM	Predictions made with the BP technique and SVM trained using the type-2 fuzzy logic system are more accurate
[126]	The authors aimed at predicting the traffic congestion along a certain road based on past data	Northern England's M62 was the source of 15-min data collection sessions beginning on October 1, 2019 and ending on October 28, 2019	CP tensor decomposition of traffic data	Compared to the cutting-edge rolling-average prediction algorithms, the proposed forecast method significantly beats the competition
[127]	The authors proposed an ML-based Intelligent Traffic Monitoring System (ML-ITMS) to predict congestion at roadside sensors	N/A	SVM and RF	Accuracy 98.6%
[128]	The authors proposed the GSA-ELM to make more precise short-term traffic predictions	Real traffic data for four of Amsterdam Ring Road's freeways: A1, A2, A4, and A8	ELM	The GSA-ELM model obtains MAPEs of 11.69%, 10.25%, 11.72%, and 12.05%, whereas the RMSEs fall in the 288.09%-163.24% range

Table 2 (continued)

Study	Methodology	Dataset	Approaches	Findings
[120]	The authors investigated Supervised ML algorithms as a Big Data based analytics technique for traffic volume prediction	The Number of vehicles moving on the road network in the Republic of Serbia recorded by many automatic traffic counters installed during that period	DT, Lazy IBk, RF, Random Committee, and Random Tree algorithms	In the first case study, the M5P algorithm has the best performance, whereas the Lazy IBk method has the best performance in the second study
[129]	The authors proposed a ML technique for predicting travel times to better organize traffic	N/A	GPR	Using non-dispersed algorithms on the input data produced complex trained model. In addition, the proposed technique has to do with the dependence on the quality of the input data
[130]	Hourly traffic forecasting on a section of road in Tangiers, a city in northern Morocco	a dataset containing annualized traffic numbers from 2013 to 2017 that was compiled by the Moroccan Center for Road Studies and Research	SLFN	The proposed model has been evaluated via a comparison with three industry standard algorithms (MLP, SVR, and ARIMA), all of which have concluded that the proposed model offers higher performance in terms of accuracy and consistency
[131]	The authors investigated the power of various ML techniques to predict traffic conditions	Preliminary data was collected over two weeks of monitoring in Bandung, Indonesia	Neural Networks, NB, DT, SVM, DNN, and DL	The size of the training data was very small. In addition, the training process must be reapplied to reflect the newer data set for every change in the training data
[132]	Evaluate the predictive skills of some ML models	Data from the road network in Thessaloniki, Greece	RF, SVR, MLP, and MLR	The SVR model works best in stable situations with small changes, while the MLP model adapts better to situations with more changes and has the fewest errors that are close to zero
[7]	A preliminary method utilizing graphical representations and dimension reduction methods for assessing road traffic accidents was suggested	Realistic road traffic accident data from Gauteng Province, South Africa (SA)	NB, Logistic regression, and K-NN	The resulting performance measures provided some comprehensive insights into the patterns of road traffic accidents
[133]	To predict future foot traffic, the authors presented a new framework based on stepwise regression inside a drift-free setting	Datasets collected from the Caltrans PeMS	Learn++ for SVR, R2C method,	The R2C architecture's prediction of motion volume suffers from a flaw due to its disregard for the spatial dependency of motion volume

In [137], the authors aimed to enhance the DBN, a DL approach, performance for accurate traffic forecasting under bad weather conditions. First, bad weather and traffic data were gathered from the IoV, rather than from the inductance coils in the usual methods. Subsequently, the SVR technique was utilized to improve the traditional DBN. The optimized DBN consists of two layers: the primary structure is the traditional DBN that unsupervised learning the basic aspects of traffic data, and the topmost layer is an SVR that implements supervised traffic forecasting. Two types of data sets were used in this study. First, traffic data from a highway control center, and second, weather data from local monitoring stations. The main issue in this study was that the computing time of the upgraded DBN requires optimization.

In [138], the authors proposed an urban traffic light control system that combines optimized traffic light scheduling techniques with traffic flow forecasting techniques. The goal was to reduce the number of vehicles that were stopped at all signal intersections on the road network. First, a framework was proposed for an urban traffic control system, which included traffic flow predictions and signal control optimization. Second, to alleviate traffic congestion, an interactive traffic light approach was used. Experiments were carried out on real-world traffic data provided by the Aliyun Tianchi platform to validate the proposed system. The comparison results showed that both the proposed system and the signal control optimization technique work well.

In [139], the authors developed a technique for constructing a traffic congestion index by extracting free-stream speed and flow. The author proposed the Traffic Congestion Index (TCI), which can synthesize changes in traffic flow and speed data to assess traffic congestion, and discussed how it is generated. Considering the correlation properties of road links in the road network, the authors introduced the technique of grouping road links based on the sub-graph to pre-train the DL model and realize information sharing across road links. A traffic congestion prediction model called SG-CNN was proposed by integrating the characteristics of the traffic data and the CNN model, and the training process was improved by the road segment aggregation method. To make the TCI more accurate, the authors must consider more information (such as weather, pedestrians, road conditions, etc.) that affects traffic congestion. Furthermore, designing a more efficient algorithm while accounting for the time complexity of the segment aggregation algorithm is an intriguing topic.

In [140], based on DL, the authors proposed a real-time data-driven queue length prediction technique. They considered a connecting corridor on which information would be transmitted from car detectors (placed at the intersection) to successive intersections. The length of the queue for crossing points in the next cycle was expected to be determined by the length of the queue for the target intersection and two upstream intersections in the current cycle. Data from the adaptive traffic control system InSync were used to train an LSTM neural network model that extracts time-dependent patterns of a signal queue. To reduce overfitting and to select the optimal hyperparameter combinations, the authors used a Sequential Model-Based Optimization (SMBO) technique to determine the appropriate dropout in different stacked layers. For this investigation, they obtained adaptive traffic light data from InSync between December 18, 2017, and February 14, 2018. The Alafaya Trail (SR-434) data for East Orlando, FL, were collected from Lake Waterford. McCulloch Road intersection includes 11 intersections. The

InSync database provides two types of data: (1) Turning Movement Counts (TMC); the number of vehicles per stage and lane per 15 min; (2) Historical data with details of each movement with time, duration, queue, and waiting time for each stage. Due to the lack of data sources, it was not possible to obtain information about the movements of the vehicles in different directions with high accuracy (30–60 s). If this information is available, the performance of the model may improve further.

In [141], the authors presented an Attention-Based Multi-Task Learning (AST-MTL) model for predicting multi-horizon traffic flow and velocity at the road network scale. To learn related tasks while improving generalization performance, this approach integrates a fully connected neural network (FNN) with a multi-headed attention mechanism. To extract the Spatiotemporal aspects of traffic states, the model incorporates graph convolutional networks (GCNs) and GRUs. FNN begins by collecting and analyzing several related functions to derive a common representation. To extract relevant information and empower the model's predictive performance, the attention mechanism also considers task-specific and shared representations. The experiments used new sets of GPS data, called On Board Unit (OBU) data, to make traffic forecasting in highway and urban contexts. This study struggles with finding the right strategy for explicitly maximizing task learning.

In [142], the authors proposed feature-injected RNNs (FI-RNNs), which incorporate temporal-sequential data with contextual elements to extract the potential correlation between traffic context and state. In this model, the stacked RNN was utilized to learn aspects of the traffic data sequence. Meanwhile, a sparse automatic encoder has been trained to increase contextual features, which are high-level abstract representations and coding of contextual elements. Subsequently, a fusion technique was developed that injects contextual information into sequence features to produce fusion features. Finally, new built-in features have been sent to the forecaster to learn traffic patterns and estimate future speed. In this study, the accuracy and performance of the proposed model should be improved by investigating more feature extraction and merging techniques. Also, the examination of other influencing elements is needed.

In [143], a traffic situational awareness array technology was developed, which takes advantage of various core models. In that approach, a graph convolution was implemented on a network of traffic detectors to extract the spatial patterns encoded in the traffic flow. After that, the retrieved features were utilized to build a weight matrix to aggregate the predictions of the base models according to their performance under a given condition. Traffic flow data obtained by Caltrans PeMS were used as a data set for this study. The main observation in this study was the need to improve the network structure and parameter options.

In [144], a traffic congestion model was proposed to predict the traffic of neighborhoods within an area using a DL model. The model was depending on the LSTM and Graph-CNN architectures. It predicts the degree of crowding, defined as the ratio of vehicle accumulation within a neighborhood to the trip completion rate. An abbreviated version of the San Francisco Bay Area Highway Network was used as a data set for this study.

In [145], a strengthened Bayesian Combination Model (BCM) with DL (IBCM-DL) for traffic flow prediction was presented to tackle the error amplification phenomenon of classical summation methods and to improve prediction performance. The revised model was built up on the BCM framework proposed by Wang [146]. Real-world traffic data were obtained by microwave sensors placed on highways in Beijing, China, provided the dataset for this study. Additional information, such as weather conditions, traffic accidents, speed, and occupancy, should be included to enhance the model's reliability.

In [147], the authors addressed the complexity of predicting urban traffic when an FCD is available. Four DL methods have been compared to highlight the ability of a neural network approach (recursive and/or convolutional) in handling the problem of traffic prediction in an urban context. In particular, the authors investigated two RNN approaches (LSTM and GRU), as well as the spatiotemporal RCN (SRCN) model and the High-Order Graph Convolutional LSTM Neural Network (HGC-LSTM) methods. To generate basic FCD inputs, the proposed solutions use a traffic simulation approach. The original FCD was created with Aimsun (2018), a microscopic traffic simulator tool for simulating each vehicle's interactions as well as collecting data from them individually. At each pre-set period, a record (vehicle ID, speed, section, and lane) is collected from the simulation for each associated vehicle. The assembly period was 10 s. In this study, the authors evaluated the performance of prediction models using two distinct urban traffic networks in Spain: Camp Nou, a small area of Barcelona with 4 nodes and 22 divisions, and Amara, a district of San Sebastian with 105 nodes and 192 sections. The results of the experiments conducted revealed that these methods can estimate traffic speeds with good performance. Specifically, recursive algorithms (LSTM and GRU) present fewer errors than convolutional ones (SRCN and HGC-LSTM). On the other hand, FCD can sometimes be insufficient to cover all sections of the network, and ML prediction of a variable without any historical data is meaningless.

In [148], the authors proposed deep artificial neural network (Deep ANN) and CNN traffic speed prediction models for upstream highway segments, including those on connected highways, under work area conditions. The proposed models can recognize congestion on the associated links as well as the upstream mainline segments. The suggested models predict traffic velocity under work zone conditions based on the volume of traffic approaching the work area, speed during normal conditions, work area capacity, distance from the work area, the vertical gradient of the road, downstream traffic volume, and type of highway section. The proposed models utilized a dropout regulation to address the ANN overfitting problems. The generated CNN model to predict traffic velocity under working zone conditions should be improved in the following aspects. Discovering additional sources to update the traffic volume to reflect the real traffic volume would enhance the accuracy of the CNN model. Furthermore, the use of a simulation model to predict the capacity of the working area can advance the generated CNN model. Automating databases via warehouses would facilitate the analysis of data for new goods and developments. Additionally, provided the availability of high-resolution data, the model can be modified to anticipate traffic congestion in the opposite direction of traffic.

In [149], the authors proposed (1) an efficient and city-wide data acquisition scheme by taking a snapshot of the Seoul Transport and Information Service (TOPIS), an open-source web-based traffic congestion map service, and (2) by integrating CNN, LSTM, and Transpose-CNN, a hybrid neural network architecture was created to retrieve Spatiotemporal information from the input image and predict network congestion. In the proposed design, an LSTM network was inserted between the convolutional encoder and the convolutional decoder. The convolutional encoder initially converts the input image sequence into low-resolution latent state sequences. The LSTM network then learns to represent time series from sequences, and the convolutional decoder finally converts the latent state to the original precision. To further enhance forecast accuracy, external factors such as weather information (rain, snow, and fog) must be addressed. Moreover, the performance of the proposed model should be enhanced. Also, more information from many data sources must be added to get more accurate forecasts.

In [150], the authors suggested an LSTM-based traffic jam prediction technique based on correcting missing temporal and spatial information. Before making predictions, the proposed technique performs a pre-processing consisting of extrinsic removal using the average absolute deviation of traffic data and correction of Spatiotemporal values using temporal and geographic trends and pattern data. While data with time-series features are not effectively learned, the suggested prediction technique utilized the LSTM model to learn time-series data to tackle this problem. The precision of forecasting traffic congestion in low-speed areas and urban areas using the proposed technique should be enhanced. Moreover, the authors need to build a model with improved user performance.

In [151], the authors suggested a deep and embedding learning (DELA) technique that could help explicitly learn accurate traffic information, road structure, and weather conditions. The original highway traffic data set contained traffic flow information for approximately 3 months (from July 19, 2016, to October 17, 2016) which was formally provided by Knowledge Discovery and Data Mining Tools Competition (KDD CUP 2017). The proposed model has poor explanatory power for the selected DL models. Also, it has a limited learning ability of the embed component.

In [119], an innovative and comprehensive technique for large-scale, faster, and real-time traffic forecasting has been suggested. It has integrated four complementary advanced technologies: big data, DL, in-memory computing, and graphics processing units (GPUs). Deep networks were trained by employing more than 11 years of data provided by the California Department of Transportation (Caltrans) [152]. The suggested approach has poor prediction accuracy, in addition to the use of a small size data set.

In [153], the authors created a distinctive traffic prediction approach with the least prediction error based on DL and introduced the LSTM model. Real-world traffic big data of performance measurement system (PeMS) were used as the dataset of this research. The count of optimized parameters employed in this study needs to be expanded. Also, the model training time needs to be regulated.

In [154], a pathway-based DL framework was presented. It can provide superior traffic velocity forecasts on a citywide scale. Furthermore, the model was reasonable and interpretable in the urban transportation context. The study area was a road network consisting of 112 road sections. The dataset used was obtained from Automated Vehicle Identification (AVI) detectors in the core area of Xuancheng, China. More essential path selection criteria were investigated. Also, raising the interpretability of a DL model for a transport application is an open topic.

In [155], using refined GPS trajectory data, the level of traffic congestion was forecasted. The Hidden Markov model has been utilized to match GPS trajectory data to the road network. The actual speed of road segments can even be calculated using GPS trajectory data from nearby locations. To predict congestion levels, four DL approaches namely CNN, RNN, LSTM, and GRU in addition to three classical ML models (ARIMA model, SVR, and ridge regression) were used. This study had some limitations that were highlighted. First, the GPS trajectory data collected were insufficient. Also, more GPS data must be considered. In addition, the structure of the CNN network can be altered to improve model performance.

In [156], the authors proposed a spatiotemporal model for the short-term prediction of the level of crowding at each part of the route (CPM-ConvLSTM). The suggested model was developed on a geographical matrix that includes both the congestion propagation pattern and the spatial correlation between road segments. The traffic data set was obtained from Helsinki, Finland. Considering the historical spatiotemporal matrices' time series, the authors applied the newly popular ConvLSTM DL model by using the time series of historical spatiotemporal matrices as input and predicting the future short-range spatiotemporal matrix. To enhance forecasting performance, the authors need to incorporate external parameters, such as points of interest, weather, and the surrounding environment.

In [157], the authors created a DL-based methodology for directly forecasting traffic status based on a time–space diagram using CNN. The time–space diagram is directly fed into the traffic forecasting model, which employs a CNN. This technique has three significant benefits: (1) It allowed the time–space diagram to be used as the input with no need for abstraction or aggregation; (2) This methodology was created through a learning mechanism that focuses on learning the key features of the time–space diagram required for effective forecasting. These features seriously affect the dynamic behavior of traffic flow and vehicle interactions, which may have an impact on future traffic conditions; and (3) This approach addressed the problem of nonparametric models' transferability by introducing location-specific solutions that needed to be re-calibrated for another location. Compared with the existing nonparametric models, that is, SVR, MLP, and ARIMA, the suggested CNN model provided a higher generalization in traffic state prediction in different regions of the main diagram. The suggested CNN model was trained using simulated data and a real-world dataset (NGSIM US-101). However, this study did not look into the effects of lane changes on traffic flow dynamic behavior and prediction accuracy.

In [158], a new method based on fuzzy CNN (F-CNN) was proposed to predict traffic flow more accurately. When uncertain information about traffic accidents is entered into CNN for the first time, a fuzzy approach is used to represent traffic accident features in this method. First, to extract the Spatiotemporal features of the traffic flow data, this study divided the whole region into 32×32 small blocks and created three direction sequences with inward and outward flow types. Second, by applying a fuzzy inference mechanism, the uncertain traffic accident information was derived from the real traffic flow data. Then, the information about the trend sequence, the information of unconfirmed traffic accidents, and the external information can be trained by implementing the F-CNN model. Moreover, pre-training and tuning procedures were designed efficiently to learn FCNN parameters. Finally, the Beijing taxi real route and meteorological data sets were applied to ensure that the proposed method has superior performance compared to the latest methods. The authors need to explore additional influential aspects in traffic flow forecasting and use more efficient DL models.

In [159], a model for short-term traffic forecasting was proposed. This model incorporates Spatiotemporal analysis and the GRU. Before applying an algorithm for spatiotemporal feature selection to determine the ideal input time window and spatial data size, the proposed model applied temporal and spatial correlation analyses to aggregated traffic flow data. Simultaneously, the desired traffic flow information is extracted from the actual traffic flow data and converted into a two-dimensional matrix containing Spatiotemporal traffic flow information. Finally, the GRU was employed to analyze the Spatiotemporal features of the internal traffic flow matrix to achieve the prediction goal. There are some issues with this work, such as other factors (for example, weather conditions) that are not included in the traffic flow, and only the traffic flow is expected for a specific section of the road.

To summarize all previous related works, Table 3 compares them in terms of methodology, data set, approaches, and their main findings.

Challenges

Traffic flows must be carefully anticipated and predicted due to the risk impact of traffic congestion, particularly in populated areas. As a result, realistic and efficient road traffic prediction techniques are required.

The publication gap in traffic flow forecasting addressed in this survey includes a lack of computationally effective methodologies and algorithms. Furthermore, there is a limitation of high-quality training data. Because of using matched city traffic flow statistics, non-exhaustive data contents were used to train network models. These characteristics were discovered to constrain the development of traffic flow prediction using ML and DL approaches.

Because of the complicated link features between road sections and traffic congestion patterns or congested areas, the gap is created by the underutilization of dynamically acquired Spatiotemporal correlations in the DL. Furthermore, a lack of computing power and distributed storage constraints traffic forecasts. A future study should investigate this issue.

The current study has several limitations, including being limited to the approaches and algorithms included in the list of articles investigated. Other strategies that were not addressed in this study could exist. Future research should focus on popularly used DL techniques (CNN and LSTM), which are thoroughly covered in the literature review. This is possible by using traffic data collected in various local urban areas to provide broader data patterns for model training. As a result, traffic forecasting in small cities will improve, as will the accuracy of the ML and DL algorithms used to predict traffic flow. The researchers' biggest challenge will be collaborating with the local urban authority to contribute the volume of vital big data. The rules and regulations for sharing traffic data with local municipal governments will be another impediment.

Table 3 A comprehensive comparative study of the previous works

Study	Methodology	Dataset	Approaches	Findings
[134]	A proposed technique to estimate the traffic flow in more populated locations	N/A	DAN, DBN, RF, and LSTM	LSTM has an accuracy of 95.2%
[135]	Anticipate trip duration between two points on a route using neural networks	Dataset obtained using Waze Live Map APIs	K-Means	Other factors such as weather conditions need to be considered to boost the efficiency and reliability of the proposed technique
[136]	A proposed strategy for short-term traffic flow forecasting	Traffic flow data generated by sensors located on road networks in Shenzhen, China	RNN and MDN	The accuracy of the proposed network is over 92%
[137]	The authors proposed a DL approach for precise traffic prediction under bad weather conditions	Two types of data sets: first, traffic data from a highway control center, and second, weather data from local monitoring stations	Traditional DBN with the topmost layer is an SVR	The proposed DBN controlled the all-time traffic prediction error by 9% and the peak time prediction error by 15% The computation time of the upgraded DBN needs optimization
[138]	The aim was to minimize the number of vehicles stopped at all signal intersections on the road network by proposing an urban traffic light control system	Real-world traffic data provided by the Aliyun Tianchi platform	Cell transmission model (CTM)	The comparison results showed that both the proposed system and the signal control optimization technique work well
[139]	A proposed technique for creating a traffic congestion index by extracting free-stream speed and flow	Caltrans PeMS dataset (http://pems.dot.ca.gov)	SG-CNN	The proposed technique has an MAE of 0.01578, an RMSE of 0.02264, and a MAPE of 0.06148
[140]	The authors proposed a real-time data-driven queue length prediction technique	Data from the adaptive traffic control system InSync	LSTM, SMBO	The LSTM model was compared with other models, such as SVR, KNN, RNN, and FFNN. The RMSE and MAE values for the LSTM model are lower than the SVR and KNN
[141]	The authors proposed an AST-MTL model for predicting multi-horizon traffic flow and velocity at the road network scale	Sets of GPS data called OBU data, which record the distance that a lorry travels within Belgium	FNN, GCN, and GRU	This study struggles with finding the right strategy for explicitly maximizing task learning
[142]	The authors proposed FI-RNNs to extract the potential correlation between traffic context and state	A real-world traffic data set is from the Digital Roadway Interactive Visualization and Evaluation Network (DRIVE Net)	Stacked RNN and Sparse Automatic Encoder	Investigating more feature extraction and merging techniques to improve performance. Also, the examination of other influencing elements is needed

Table 3 (continued)

Study	Methodology	Dataset	Approaches	Findings
[143]	A traffic situational awareness array technology was developed to extract the spatial patterns encoded in the traffic flow	Caltrans PeMS dataset	SVR and LSTM	main observation in this study is the need to improve the network structure and parameter options
[144]	A traffic congestion model was proposed to predict the traffic of neighborhoods within an area	An abbreviated version of the San Francisco Bay Area Highway Network	LSTM and Graph-CNN	The final tests showed that the model is scalable
[145]	A traffic flow prediction technique was presented to tackle the error amplification phenomenon of classical summation methods and to improve prediction performance	Real-world traffic data was obtained by microwave sensors placed on highways in Beijing, China	The IBCM-DL model is built upon the BCM framework proposed by Wang [146]	Extra factors, such as weather conditions, traffic accidents, speed, and occupancy should be considered to improve the model's reliability
[147]	Evaluate the predictive skills of some ML models	The original data was created by Aimsun (2018)	LSTM, GRU, SRCN, and HGC-LSTM	LSTM and GRU presented fewer errors than SRCN and HGC-LSTM
[148]	Traffic speed prediction models for upstream under work area conditions were proposed	Realistic road traffic accident data from Gauteng Province, South Africa (SA)	Deep ANN and CNN	The generated CNN model should be improved in many aspects such as discovering additional sources to update the traffic volume and anticipate traffic congestion in the opposite direction of traffic
[149]	The authors proposed a hybrid neural network architecture to retrieve Spatiotemporal information from the input image and predict network congestion	Datasets collected from the Caltrans PeMS	CNN, LSTM, and Transpose-CNN	Additional factors such as weather information (rain, snow, and fog) must be addressed. Also, more information from many data sources must be added to get more accurate forecasts
[150]	The authors suggested a traffic jam prediction technique based on correcting missing temporal and spatial information	The traffic speed data from the ITS are collected by traffic information collectors installed on the roads or along the roadsides	LSTM	the proposed method was found to achieve better performance with a difference in the MAPE of 3%–17% The forecasting precision should be enhanced Also, the authors need to build a model with improved user performance
[151]	The authors suggested the DELA technique that could help explicitly learn accurate traffic information, road structure, and weather conditions	Traffic flow information for approximately 3 months (from July 19, 2016, to October 17, 2016) provided by KDD CUP 2017	CNN and LSTM	Poor explanatory power for the selected DL models A limited learning ability of the embedded component

Table 3 (continued)

Study	Methodology	Dataset	Approaches	Findings
[119]	A technique for large-scale, faster, and real-time traffic forecasting	11 years of data provided by the California Department of Transportation (Caltrans) [152]	Big Data, In-memory computing, GPUs, and DL	The suggested approach has poor prediction accuracy
[153]	A distinctive traffic prediction approach with the least prediction error	Real-world traffic big data of PeMS	LSTM	The model training time needs to be regulated The number of optimized parameters needs to be expanded
[154]	A pathway-based DL framework that can provide superior traffic velocity forecasts on a citywide scale was presented	Data from Automated Vehicle Identification (AVI) detectors in the core area of Xuancheng, China	Bi-LSTM	The model was reasonable and interpretable in the urban transportation context raising the interpretability of a DL model for a transport application is an issue
[155]	Forecasting traffic congestion level using refined GPS trajectory data and the Hidden Markov model	GPS trajectory data	CNN, RNN, LSTM, GRU, ARIMA, SVR, and Ridge Regression	Insufficient GPS trajectory data was collected. More GPS data must be taken into account. The network structure can be altered to improve model performance
[156]	A spatiotemporal model for the short-term prediction of the crowding level	Traffic congestion data of Helsinki, Finland collected using HERE Traffic API	ConvLSTM	External parameters, such as points of interest, weather, and the surrounding environment should be incorporated
[157]	A DL-based methodology for directly forecasting traffic status based on a time–space diagram	Simulated data and a real-world dataset (NGSIM US-101)	CNN	Compared with SVR, MLP, and ARIMA, the suggested CNN model provided a higher generalization in traffic state prediction
[158]	Predict traffic flow more accurately based on fuzzy CNN	The Beijing taxi real route and meteorological data	F-CNN	Additional influential aspects in traffic flow forecasting need to be explored and more efficient DL models need to be used
[159]	A model for short-term traffic forecasting was proposed	Traffic flow data of the Xiangjiang Middle Road in Changsha City except the weekend of May 20th, 2018, for a total of 40 days	Spatiotemporal analysis and the GRU	Other factors such as weather conditions need to be considered

The installation of sensors to collect traffic data for training ML and DL may result in connected IoT settings that increase cybersecurity risks. A framework should be developed to address the cybersecurity issues of ITS in smart cities. This leaves plenty of room for future investigation.

Conclusion

The present study is aimed to present a comprehensive review of the most significant ML and DL techniques used in traffic forecasting, as well as the problems associated with using ML and DL in traffic forecasting. A total of 40 articles were chosen and thoroughly reviewed after a rigorous selection process. According to the preceding discussion, traffic forecasting is an important task in the transportation industry due to its significant influence on road construction, route planning, and traffic rules. This work advances research in the field of traffic flow forecasting using ML and DL approaches. Contributes to the literature and future studies by serving as a resource for other academics and researchers.

Acknowledgements
Not applicable.

Author contributions
SAS wrote the main text of the manuscript; YA-H and HAH revised the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Received: 27 December 2022 Accepted: 25 February 2023
Published online: 09 March 2023

References

1. Nellore K, Hancke G (2016) A survey on urban traffic management system using wireless sensor networks. Sensors 16:157
2. Patel P, Narmawala Z, Thakkar A (2019) A survey on intelligent transportation system using internet of things. In: Emerging research in computing, information, communication and applications, pp 231–240
3. An S, Lee B-H, Shin D-R (2011) A survey of intelligent transportation systems. In: 2011 third international conference on computational intelligence, communication systems and networks
4. Qureshi KN, Abdullah AH (2013) A survey on intelligent transportation systems. Middle-East J Sci Res 15:629–642
5. Chen C, Li K, Teo SG, Zou X, Li K, Zeng Z (2020) Citywide traffic flow prediction based on multiple gated Spatio-temporal convolutional neural networks. ACM Trans Knowl Discov Data (TKDD) 14(4):1–23
6. Sun P, Boukerche A, Tao Y (2020) SSGRU: a novel hybrid stacked GRU- based traffic volume prediction approach in a road network. Comput Commun 160:502–511
7. Makaba T, Doorsamy W, Paul BS (2020) Exploratory framework for analyzing road traffic accident data with validation on Gauteng province data. Cogent Engineering 7(1):1834659
8. World Health Organization (2018) Global status report on road safety 2018 summary. https://apps.who.int/iris/bitstream/handle/10665/277370/WHO-NMH-NVI-18.20-eng.pdf?ua=1
9. Bengio Y (2009) Learning deep architectures for AI. Now Publishers Inc.
10. Van Der Voort M, Dougherty M, Watson S (1996) Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transp Res Part C Emerg Technol 4(5):307–318
11. Lee S, Fambro DB (1999) Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transp Res Rec J Transp Res Board 1678(1):179–188

12. Williams BM (2001) Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp Res Rec J Transp Res Board 1776(1):194–200
13. Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J Transp Eng 129(6):664–672
14. Chen K, Chen F, Lai B, Jin Z, Liu Y, Li K, Wei L, Wang P, Tang Y, Huang J, Hua X (2020) Dynamic Spatio-temporal graph-based CNNs for traffic flow prediction. IEEE Access 8:185136–185145
15. Kashyap AA, Raviraj S, Devarakonda A, Nayak KSR, Santhosh KV, Bhat SJ (2022) Traffic flow prediction models—a review of deep learning techniques. Cogent Eng 9(1):2010510
16. Smith BL, Demetsky MJ (1994) Short-term traffic flow prediction: neural network approach. Transp Res Rec 98–104
17. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, May 7–9, 2015, San Diego, USA. https://arxiv.org/abs/1409.1556
18. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: International conference on acoustics, speech and signal processing, 26–31 May 2013, Vancouver, Canada
19. Sainath TN, Vinyals O, Senior A, Sak H (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: International conference on acoustics, speech and signal processing, 19–24 April 2015, South Brisbane, Australia
20. Good Fellow IJ, Mirza M, Courville A, Bengio Y (2013) Multi-prediction deep Boltzmann machines. In: Proceedings of the 26th international conference on neural information processing systems, Lake Tahoe, USA
21. Sarikaya R, Hinton GE, Deoras A (2014) Application of deep belief networks for natural language understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22(4):778–784
22. Gehring J, Miao Y, Metze F, Waibel A (2013) Extracting deep bottleneck features using stacked auto-encoders. In: International conference on acoustics, speech and signal processing, 26–31 May 2013. IEEE, Vancouver, Canada
23. Zhang J, Wang F-Y, Wang K, Lin W-H, Xu X, Chen C (2011) Data-driven intelligent transportation systems: a survey. IEEE Trans Intell Transp Syst 12(4):1624–1639
24. Chowdary GJ (2021) Machine learning and deep learning methods for building intelligent systems in medicine and drug discovery: a comprehensive survey. arXiv preprint arXiv:2107.14037.
25. Singh G, Al'Aref SJ, Van Assen M, Kim TS, van Rosendael A, Kolli KK, Dwivedi A, Maliakal G, Pandey M, Wang J, Do V, Gummalla M, De Cecco CN, Min JK (2018) Machine learning in cardiac CT: basic concepts and contemporary data. J Cardiovasc Comput Tomogr 12(3):192–201
26. Ahsan MM, Luna SA, Siddique Z (2022) Machine-learning-based disease diagnosis: a comprehensive review. Healthcare 10(3):541
27. Dey A (2016) Machine learning algorithms: a review. Int J Comput Sci Inf Technol 7(3):1174–1179
28. Dhall D, Kaur R, Juneja M (2020) Machine learning: a review of the algorithms and its applications. Proceedings of ICRIC 2019:47–63
29. Kotsiantis SB, Zaharakis I, Pintelas P (2007) Supervised machine learning: a review of classification techniques. Emerg Artif Intell Appl Comput Eng 160(1):3–24
30. Veropoulos K, Campbell C, Cristianini N (1999) Controlling the sensitivity of support vector machines. In: Proceedings of the international joint conference on artificial intelligence (IJCAI99)
31. Taunk K, De S, Verma S, Swetapadma A (2019) A brief review of nearest neighbor algorithm for learning and classification. In: 2019 international conference on intelligent computing and control systems (ICCS)
32. Obulesu O, Mahendra M, ThrilokReddy M (2018) Machine learning techniques and tools: a survey. In: 2018 international conference on inventive research in computing applications (ICIRCA). IEEE, pp 605–611
33. Ray S (2019) A quick review of machine learning algorithms. In: 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, pp 35–39
34. Kumar R, Verma R (2012) Classification algorithms for data mining: a survey. Int J Innov Eng Technol 1(2):7–14
35. Nikam SS (2015) A comparative study of classification techniques in data mining algorithms. Orient J Comput Sci Technol 8(1):13–19
36. Stein G, Chen B, Wu AS, Hua KA (2005) Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd annual Southeast regional conference, vol 2, pp 136–141
37. Damanik IS, Windarto AP, Wanto A, Andani SR, Saputra W (2019) Decision tree optimization in C4. 5 Algorithm using genetic algorithm. J Phys Conf Ser 1255(1):012012
38. Gavankar SS, Sawarkar SD (2017) Eager decision tree. In: 2017 2nd international conference for convergence in technology (I2CT), Mumbai, April 2017, pp 837–840
39. Mahesh B (2020) Machine learning algorithms—a review. Int J Sci Res 9:381–386
40. Janikow CZ (1998) Fuzzy decision trees: issues and methods. IEEE Trans Syst Man Cybern Part B 28(1):1–14
41. Charbuty B, Abdulazeez A (2021) Classification based on decision tree algorithm for machine learning. J Appl Sci Technol Trends 2(01):20–28
42. Zhao Y, Zhang Y (2008) Comparison of decision tree methods for finding active objects. Adv Space Res 41(12):1955–1959
43. Mittal K, Khanduja D, Tewari PC (2017) An insight into 'decision tree analysis.' World Wide J Multidiscip Res Dev 3(12):111–115
44. Priyanka, Kumar D (2020) Decision tree classifier: a detailed survey. Int J Inf Decis Sci 12(3):246–269
45. Breiman L (2001) Random forests. Mach Learn 54(1):5–32
46. Pal M (2005) Random forest classifier for remote sensing classification. Int J Remote Sens 26(1):217–222
47. Cutler DR, Edwards TC Jr, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ (2007) Random forests for classification in ecology. Ecology 88(11):2783–2792
48. Belgiu M, Drăguţ L (2016) Random forest in remote sensing: a review of applications and future directions. ISPRS J Photogramm Remote Sens 114:24–31
49. He Y, Lee E, Warner TA (2017) A time series of annual land use and land cover maps of China from 1982 to 2013 generated using AVHRR GIMMS NDVI3g data. Remote Sens Environ 199:201–217

50. Maxwell AE, Warner TA, Fang F (2018) Implementation of machine-learning classification in remote sensing: an applied review. Int J Remote Sens 39(9):2784–2817
51. Breiman L (2001) Random forests. Mach Learn 45:5–32
52. Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Comput 9(7):1545–1588
53. Ho TK (1995) Random decision forests. In: 3rd international conference on document analysis and recognition—volume 1 (ICDAR'95). IEEE Computer Society, pp 278–282
54. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
55. Resende PAA, Drummond AC (2018) A survey of random forest-based methods for intrusion detection systems. ACM Comput Surv 51(3):1–36
56. Oladipo ID, AbdulRaheem M, Awotunde JB, Bhoi AK, Adeniyi EA, Abiodun MK (2022) Machine learning and deep learning algorithms for smart cities: a start-of-the-art review. In: IoT and IoE driven smart cities, pp 143–162
57. Dormann CF, Elith J, Bacher S, Buchmann C, Carl G, Carré G, Marquéz JRG, Gruber B, Lafourcade B, Leitão PJ, Münkemüller T, McClean C, Osborne PE, Reineking B, Schröder B, Skidmore AK, Zurell D, Lautenbach S (2013) Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. Ecography 36(1):27–46
58. Harrington P (2012) Machine Learning in action. Manning Publications Co., Shelter Island
59. McLachlan GJ (2005) Discriminant analysis and statistical pattern recognition. Wiley
60. Jolliffe IT (1986) Principal component analysis. SpringerVerlag, New York
61. Gow J, Baumgarten R, Cairns P, Colton S, Miller P (2012) Unsupervised modeling of player style with LDA. IEEE Trans Comput Intell AI Games 4(3):152–166
62. Coronato A, Naeem M, De Pietro G, Paragliola G (2020) Reinforcement learning for intelligent healthcare applications: a survey. Artif Intell Med 109:101964
63. Watkin CJCH, Dayan P (1992) Technical note Q-learning. Mach Learn 8(3):279–292
64. Watkins CJCH (1989) Learning from delayed rewards. Ph.D. Thesis, University of Cambridge, England
65. Achille A, Soatto S (2018) Information dropout: Learning optimal representations through noisy computation. IEEE Trans Pattern Anal Mach Intell 40:2897–2905
66. Williams G, Wagener N, Goldfain B, Drews P, Rehg JM, Boots B, Theodorou EA (2017) Information-theoretic mpc for model-based reinforcement learning. In: IEEE international conference on robotics and automation (ICRA), pp 1714–1721
67. Wilkes JT, Gallistel CR (2017) Information theory, memory, prediction, and timing in associative learning. In: Computational models of brain and behavior, pp 481–492
68. Jang B, Kim M, Harerimana G, Kim JW (2019) Q-learning algorithms: a comprehensive classification and applications. IEEE Access 7:133653–133667
69. An Y, Wang Y, Meng H (2017) Multi-task deep learning for user intention understanding in speech interaction systems
70. Shi X, Gao Z, Lausen L, Wang H, Yeung DY, Wong WK, Woo WC (2017) Deep learning for precipitation nowcasting: a benchmark and a new model. In: Advances in neural information processing systems, pp 5622–5632
71. Juang C-F, Lu C-M (2009) Ant colony optimization incorporated with fuzzy Q-learning for reinforcement fuzzy control. IEEE Trans Syst Man Cybern Part A Syst Hum 39(3):597–608
72. Świechowski M, Godlewski K, Sawicki B, Mańdziuk J (2021) Monte Carlo tree search: a review of recent modifications and applications. arXiv preprint arXiv:2103.04931
73. Lizotte DJ, Laber EB (2016) Multi-objective Markov decision processes for data-driven decision support. J Mach Learn Res 17:211:1-211:28
74. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
75. Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, Perez D, Samothrakis S, Colton S (2012) A survey of monte carlo tree search methods. IEEE Trans Comput Intell AI Games 4(1):1–43
76. Baier H, Drake PD (2010) The power of forgetting: improving the last good-reply policy in Monte Carlo Go. IEEE Trans Comput Intell AI Games 2(4):303–309
77. Alpaydin E (2020) Introduction to machine learning. MIT Press
78. Mikolov T et al (2013) Efficient estimation of word representations in vector space
79. Nguyen G, Dlugolinsky S, Bobák M, Tran V, García ÁL, Heredia I et al (2019) Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52(1):77–124
80. Aggour KS, Gupta VK, Ruscitto D, Ajdelsztajn L, Bian X, Brosnan KH et al (2019) Artificial intelligence/machine learning in manufacturing and inspection: a GE perspective. MRS Bull 44(7):545–558
81. Khan FN, Fan Q, Lu C, Lau APT (2020) Machine learning methods for optical communication systems and networks. Optical fiber telecommunications VII. Academic Press, New York, pp 921–978
82. Pouyanfar S, Sadiq S, Yan Y, Tian H, Tao Y, Reyes MP et al (2018) A survey on deep learning: algorithms, techniques, and applications. ACM Comput Surv (CSUR) 51(5):1–36
83. Dargan S, Kumar M, Ayyagari MR, Kumar G (2019) A survey of deep learning and its applications: a new paradigm to machine learning. Arch Comput Methods Eng 27(4):1–22
84. Lauzon FQ (2012) An introduction to deep learning. In: 2012 11th international conference on information science, signal processing and their applications (ISSPA). IEEE, pp 1438–1439
85. Ling ZH, Kang SY, Zen H, Senior A, Schuster M, Qian XJ, Meng HM, Deng L (2015) Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. IEEE Signal Process Mag 32(3):35–52
86. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117

87. Yu D, Deng L (2010) Deep learning and its applications to signal and information processing [exploratory dsp]. IEEE Signal Process Mag 28(1):145–154
88. Yap MH, Pons G, Marti J, Ganau S, Sentis M, Zwiggelaar R, Davison AK, Marti R (2017) Automated breast ultrasound lesions detection using convolutional neural networks. IEEE J Biomed Health Inform 22(4):1218–1226
89. Fukushima K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw 1:119–130
90. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324
91. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
92. Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Hasan M, Van Essen BC, Awwal AA, Asari VK (2019) A state-of-the-art survey on deep learning theory and architectures. Electronics 8(3):292
93. Apaydin H, Feizi H, Sattari MT, Colak MS, Shamshirband S, Chau KW (2020) Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. Water 12(5):1500
94. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing, pp 6645–6649
95. Baturdinler Ö, Aydin N (2020) An optimal feature parameter set based on gated recurrent unit recurrent neural networks for speech segment detection. Appl Sci 10(4):1273
96. Jagannatha AN, Yu H (2016) 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. vol 2016. NIH Public Access, p 856
97. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8(1):1–74
98. Pascanu R, Gulcehre C, Cho K, Bengio Y (2013) How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026
99. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
100. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
101. Smagulova K, James AP (2019) A survey on LSTM memristive neural network architectures and applications. Eur Phys J Spec Top 228(10):2313–2324
102. Setyanto A, Laksito A, Alarfaj F, Alreshoodi M, Oyong I, Hayaty M, Alomair A, Almusallam N, Kurniasari L (2022) Arabic language opinion mining based on long short-term memory (LSTM). Appl Sci 12(9):4140
103. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: Continual prediction with LSTM. Neural Comput 12(10):2451–2471
104. Lindemann B, Müller T, Vietz H, Jazdi N, Weyrich M (2021) A survey on long short-term memory networks for time series prediction. Procedia CIRP 99:650–655
105. Cui Z, Ke R, Pu Z, Wang Y (2018) Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143
106. Villegas R, Yang J, Zou Y, Sohn S, Lin X, Lee H (2017) Learning to generate long-term future via hierarchical prediction. In: International conference on machine learning. PMLR, pp 3560–3569
107. Chu KF, Lam AY, Li VO (2019) Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions. IEEE Trans Intell Transp Syst 21(8):3219–3232
108. Gensler A, Henze J, Sick B, Raabe N (2016) Deep Learning for solar power forecasting—an approach using AutoEncoder and LSTM Neural Networks. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 002858–002865
109. Hsu D (2017) Multi-period time series modeling with sparsity via Bayesian variational inference. arXiv preprint arXiv:1707.00666
110. Kalchbrenner N, Danihelka I, Graves A (2015) Grid long short-term memory. arXiv preprint arXiv:1507.01526
111. Veličković P, Karazija L, Lane ND, Bhattacharya S, Liberis E, Liò P, Chieh A, Bellahsen O, Vegreville M (2018) Cross-modal recurrent models for weight objective prediction from multimodal time-series data. In: Proceedings of the 12th EAI international conference on pervasive computing technologies for healthcare, pp 178–186
112. Wang J, Hu X (2021) Convolutional neural networks with gated recurrent connections. IEEE Trans Pattern Anal Mach Intell 44:3421–3435
113. Liang M, Hu X (2015) Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3367–3375
114. Liang M, Hu X, Zhang B (2015) Convolutional neural networks with intra-layer recurrent connections for scene labeling. In: Advances in neural information processing systems, 28
115. Fernandez B, Parlos AG, Tsai W (1990) Nonlinear dynamic system identification using artificial neural networks (ANNs). In: International joint conference on neural networks (IJCNN), pp 133–141
116. Puskorius GV, Feldkamp LA (1994) Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks. IEEE Trans Neural Netw 5(2):279–297
117. Rumelhart DE, Hinton GE, Williams RJ (1986). Chapter: learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition, vol 1. MIT Press, pp 318–362
118. Lippi M, Bertini M, Frasconi P (2013) Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Trans Intell Transp Syst 14(2):871–882
119. Aqib M, Mehmood R, Alzahrani A, Katib I, Albeshri A, Altowaijri SM (2019) Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs. Sensors 19:2206
120. Janković S, Uzelac A, Zdravković S, Mladenović D, Mladenović S, Andrijanić I (2021) Traffic volumes prediction using big data analytics methods. Int J Traffic Transp Eng 11(2):184–198
121. Deekshetha HR, Shreyas Madhav AV, Tyagi AK (2022) Traffic prediction using machine learning. In: Evolutionary computing and mobile sustainable networks. Springer, Singapore, pp 969–983

122. Kuamr S (2022) Traffic flow prediction using machine learning algorithms. Int Res J Eng Technol 9(4):2995–3004
123. Navarro-Espinoza A, López-Bonilla OR, García-Guerrero EE, Tlelo-Cuautle E, López-Mancilla D, Hernández-Mejía C, Inzunza-González E (2022) Traffic flow prediction for smart traffic lights using machine learning algorithms. Technologies 10(1):5
124. Upadhyaya S, Mehrotra D (2022) The facets of machine learning in lane change prediction of vehicular traffic flow. In: Proceedings of international conference on intelligent cyber-physical systems. Springer, Singapore, pp 353–365
125. Qu Z, Li J (2022) Short-term traffic flow forecast on basis of PCA-interval type-2 fuzzy system. J Phys Conf Ser 2171(1):012051
126. Steffen T, Lichtenberg G (2022). A machine learning approach to traffic flow prediction using CP data tensor decompositions. In: IFAC world congress 2020. Loughborough Research Repository
127. Wang J, Pradhan MR, Gunasekaran N (2022) Machine learning-based human-robot interaction in ITS. Inf Process Manag 59(1):102750
128. Cui Z, Huang B, Dou H, Tan G, Zheng S, Zhou T (2022) GSA-ELM: a hybrid learning model for short-term traffic flow forecasting. IET Intel Transport Syst 16(1):41–52
129. Li J, Boonaert J, Doniec A, Lozenguez G (2021) Multi-models machine learning methods for traffic flow estimation from Floating Car Data. Transp Res Part C Emerg Technol 132:103389
130. Jiber M, Mbarek A, Yahyaouy A, Sabri MA, Boumhidi J (2020) Road traffic prediction model using extreme learning machine: the case study of Tangier. Morocco Inf 11(12):542
131. Husni E, Nasution SM, Yusuf R (2020) Predicting traffic conditions using knowledge-growing Bayes classifier. IEEE Access 8:191510–191518
132. Bratsas C, Koupidis K, Salanova JM, Giannakopoulos K, Kaloudis A, Aifadopoulou G (2020) A comparison of machine learning methods for the prediction of traffic speed in urban places. Sustainability 12(1):142
133. Xiao J, Xiao Z, Wang D, Bai J, Havyarimana V, Zeng F (2019) Short-term traffic volume prediction by ensemble learning in concept drifting environments. Knowl-Based Syst 164:213–225
134. Ramchandra NR, Rajabhushanam C (2022) Machine learning algorithms performance evaluation in traffic flow prediction. Mater Today Proc 51:1046–1050
135. Pangesta J, Dharmadinata OJ, Bagaskoro ASC, Hendrikson N, Budiharto W (2021) Travel duration prediction based on traffic speed and driving pattern using deep learning. ICIC Express Lett Part B Appl 12(1):83–90
136. Chen M, Chen R, Cai F, Li W, Guo N, Li G (2021) Short-term traffic flow prediction with recurrent mixture density network. Math Problems Eng 2021:6393951
137. Bao X, Jiang D, Yang X, Wang H (2021) An improved deep belief network for traffic prediction considering weather factors. Alex Eng J 60(1):413–420
138. Jiang CY, Hu XM, Chen WN (2021) An urban traffic signal control system based on traffic flow prediction. In: 2021 13th international conference on advanced computational intelligence (ICACI). IEEE, pp 259–265
139. Tu Y, Lin S, Qiao J, Liu B (2021) Deep traffic congestion prediction model based on road segment grouping. Appl Intell 51(11):8519–8541
140. Rahman R, Hasan S (2021) Real-time signal queue length prediction using long short-term memory neural network. Neural Comput Appl 33(8):3311–3324
141. Buroni G, Lebichot B, Bontempi G (2021) AST-MTL: an attention-based multi-task learning strategy for traffic forecasting. IEEE Access 9:77359–77370
142. Qu L, Lyu J, Li W, Ma D, Fan H (2021) Features injected recurrent neural networks for short-term traffic speed prediction. Neurocomputing 451:290–304
143. Chen Y, Lv Y, Ye P, Zhu F (2020) Traffic-condition-awareness ensemble learning for traffic flow prediction. IFAC-PapersOnLine 53(5):582–587
144. Mohanty S, Pozdnukhov A, Cassidy M (2020) Region-wide congestion prediction and control using deep learning. Transp Res Part C Emerg Technol 116:102624
145. Gu Y, Lu W, Xu X, Qin L, Shao Z, Zhang H (2020) An improved Bayesian combination model for short-term traffic prediction with deep learning. IEEE Trans Intell Transp Syst 21(3):1332–1342
146. Wang J, Deng W, Guo Y (2014) New Bayesian combination method for short-term traffic flow forecasting. Transp Res C Emerg Technol 43:79–94
147. Vázquez JJ, Arjona J, Linares M, Casanovas-Garcia J (2020) A comparison of deep learning methods for urban traffic forecasting using floating car data. Transportation Research Procedia 47:195–202
148. Shabarek A (2020) A deep machine learning approach for predicting freeway work zone delay using big data. Doctoral dissertation, New Jersey Institute of Technology
149. Ranjan N, Bhandari S, Zhao HP, Kim H, Khan P (2020) City-wide traffic congestion prediction based on CNN, LSTM, and transpose CNN. IEEE Access 8:81606–81620
150. Shin DH, Chung K, Park RC (2020) Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data. IEEE Access 8:150784–150796
151. Zheng Z, Yang Y, Liu J, Dai HN, Zhang Y (2019) Deep and embedded learning approach for traffic flow prediction in urban informatics. IEEE Trans Intell Transp Syst 20(10):3927–3939
152. California Department of Transportation (Caltrans). Caltrans Performance Measurement System (PeMS) Available online: http://pems.dot.ca.gov/. Accessed 13 May 2019
153. Kong F, Li J, Jiang B, Zhang T, Song H (2019) Big data-driven machine learning-enabled traffic flow prediction. Trans Emerg Telecommun Technol 30(9):e3482
154. Wang J, Chen R, He Z (2019) Traffic speed prediction for urban transportation network: a path based deep learning approach. Transp Res Part C Emerg Technol 100:372–385
155. Sun S, Chen J, Sun J (2019) Traffic congestion prediction based on GPS trajectory data. Int J Distrib Sens Netw 15(5):1550147719847440
156. Di X, Xiao Y, Zhu C, Deng Y, Zhao Q, Rao W (2019) Traffic congestion prediction by spatiotemporal propagation patterns. In: 2019 20th IEEE international conference on mobile data management (MDM). IEEE, pp 298–303

157. Khajeh Hosseini M, Talebpour A (2019) Traffic prediction using time-space diagram: a convolutional neural network approach. Transp Res Rec 2673(7):425–435
158. An J, Fu L, Hu M, Chen W, Zhan J (2019) A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information. IEEE Access 7:20708–20722
159. Dai G, Ma C, Xu X (2019) Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU. IEEE Access 7:143025–143035

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations