Deep Learning for Traffic Prediction in VANET

Shishir Singh Chauhan, Juhi Singh, Gauri Shanker Gupta, and Mrinal Kumar Pathak

Abstract Due to advancements in computing systems, intelligent transportation networks, and communication technologies, the development of VANETs is possible. Traffic prediction in high mobility and dynamic network topologies is highly challenging in VANETs. It is very important for high accuracy in traffic prediction as it reduces congestion, provides smooth flow of traffic, and ensures road safety. This chapter is going to discuss how deep learning technologies can be used in solving the problems above. It gives importance to different methodologies, architectures, and practical implementations. Deep learning is one of the subsets of machine learning that has gained much popularity due to its excellent ability to predict complex relationships and patterns in large datasets. It checks applicability to the problem of traffic-patterns forecasting and solving time series problems using CNN, RNN, and LSTM. This chapter refers to strategies on data preparation-data cleaning, normalization, and augmentation. Data preparation with its key demands-normalization, augmentation makes the input data correct for the deep learning model application; thus, correctness also relies on relevance. A list of relevant topics incorporates federated learning and transfer learning. Even though it doesn't solve the problem of sparse data, transfer learning helps with the adaptation and adoption of models developed from big sets of data to new and related settings. In federated learning, instead, diversity within centralised collections of data is increased because these models decentralised work to perform computations but differ by their levels of privacy. This comprehensive review of deep learning algorithms for traffic prediction in VANETs can provide much-needed insight to academics, practitioners, and stakeholders interested in intelligent transportation systems.

S. S. Chauhan (✉) · J. Singh
Manipal University Jaipur, Jaipur, India
e-mail: shishir.chauhan@jaipur.manipal.edu

G. S. Gupta · M. K. Pathak
Birla Institute of Technology Mesra, Mesra, India
e-mail: gaurishankergupta@bitmesra.ac.in

M. K. Pathak
e-mail: mrinalpathak@bitmesra.ac.in

J. Bhatia et al. (eds.), *Deep Learning Based Solutions for Vehicular Adhoc Networks*,
Studies in Computational Intelligence 1207,
https://doi.org/10.1007/978-981-96-5190-0_7

Keywords Deep learning · Traffic prediction · VANET · Automated traffic system

1 Introduction

Urbanization and increasing vehicle ownership are the leading causes of traffic congestion, which worsens transportation problems. According to Dechenaux et al. [1], the usual strategies for increasing the number of lanes may not be enough to reduce traffic congestion. The ITS aims at enhancing the efficiency of traffic and the capacity of roads with the reduction of pollution and accidents through the implementation of 5G communication and on-road sensors [2–8]. Accurate prediction of road traffic, which predicts volumes of traffic using historical data along with geographical relationships, is an integral part of Intelligent Transportation Systems (ITS) [9–12]. Such predictions underlie traffic management techniques, that include vehicular clouds (VC) and resource allocation.

Understanding how different traffic patterns interact over time is essential for predicting traffic flow [9]. Long-term trends show consistent rises in the morning and declines in the evening, which are impacted by things like daily working hours. Unpredictable variations are introduced by short-term trends that are impacted by unforeseen occurrences such as weather changes or accidents. These patterns include seasonal changes (regular cycles), random variations (chaotic changes brought on by outside sources), and trend variations (long-term directed changes) [10]. Because of these nonlinear dynamics, which are impacted by both human and environmental influences, traffic flow prediction is difficult.

In their survey of traffic prediction techniques, Nagy et al. [11] categorized predictive models into three main categories: naïve, parametric, and non-parametric. Analyzing several public datasets, their survey shows how the mobility of sensors improves predictive performance. However, Do et al. [12] specifically focused on NN models designed for traffic forecasting. The study gave a comprehensive analysis of the predictions made by neural networks in traffic forecasting, considering a wide range of neural network architectures, activation functions, and layer configurations over various prediction scenarios. All these studies together provide very important insights into the usability and practical implementation of various methods of traffic forecasting.

Recent works on DL models applied in predicting traffic flow include the examination of the structural composition of DL frameworks in NN architectures as well as the evaluation of various DL models for given applications in traffic prediction. For example, in the work that was cited as [13], the authors have considered various DL models and DL techniques applied in designing effective DL prediction systems for traffic. They carried out thorough analyses on numerous traffic status prediction tasks using a common dataset, and their comparative study on the advantages and disadvantages of each DL prediction method shed light on the effectiveness and applicability of DL models in all scenarios of traffic flow forecast. We require more such comparison analyses to gain a better understanding of practical applications of deep

learning with traffic forecasting. It provides sharp prescriptions to researchers and practitioners that will be attempting to improve or implement deep learning-based approaches to achieve higher accuracy and reliability for traffic flow estimates.

The literature clearly indicates that machine learning (ML) methodologies are progressively used in traffic flow prediction jobs. Their ability to adeptly capture complex non-linear connections, handle prediction jobs with flexibility, and use large amounts of traffic pattern data is what attracts them [14]. Collectively, these components provide credibility to their increasing acceptability and efficacy in addressing traffic forecasting challenges.

This study will concentrate on the machine learning models used to traffic prediction challenges. We also consider the relevant cases that the ML model has been used for in the meantime. It is crucial that we evaluate various model types according to their correctness as well as attributes like their efficiency, hardware and data dependence, and capacity to handle certain issues. As is well known, machine learning (ML) is a vast field with a wide variety of categorization techniques for ML models based on various viewpoints. As shown in Fig. 1 in accordance with [15–18], the approaches we studied in this study are grouped by taxonomy of traffic prediction models based on ML and DL algorithms. These are further explained as follows:

Regression model: This model looks at how the dependent and independent variables are related in order to determine how to fit the dataset with a curve or line.

Example-based model: To solve the prediction problem, the example-based model first identified previous data samples that were similar to the input sequence. It then used these samples to determine what its final forecast should be.

Kernel model: The kernel technique employs a kernel function to transform the input data into a high-dimensional linear space, therefore simplifying prediction problems. The primary subjects of the inquiry are the Radial Basis Function and Support Vector Machine models.

NN model: A particular kind of computer model that emulates the information-processing capabilities of biological neurones in the brain is referred to as the neural network (NN) model. The framework has many tiers of interlinked nodes. Each node, or neurone, calculates a weighted sum of the input data before applying an activation function, which then generates an output. The particular neural network model used affects the network's design, including the amount of layers and the quantity of neurones in each layer (e.g., feedforward neural networks, convolutional neural networks, and recurrent neural networks). The input signal received by the neurone is transformed into an output signal via the activation function, which is then used to provide classifications or predictions.

Hybrid model: In forecasting, a hybrid model aggregates forecasts from many separate models to increase precision and consistency. A hybrid model strives to generate more reliable projections than any one model could by combining the advantages of many models, such as statistical, machine learning, or deep learning techniques. This method entails choosing suitable base models, figuring out how to combine their forecasts (for example, by stacking or averaging), and

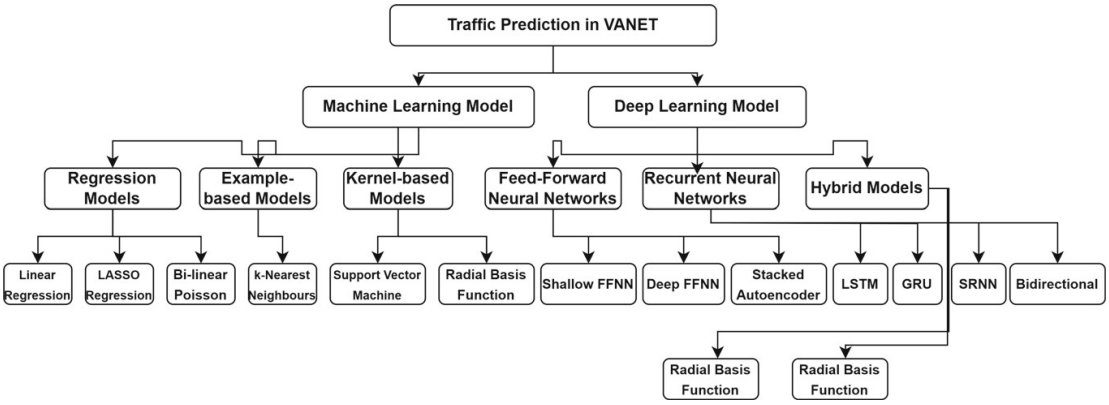

Fig. 1 Taxonomy of traffic prediction models of ML and DL

then fine-tuning the plan in light of performance assessment. Hybrid models, which combine several modelling approaches to improve predictive ability, are frequently employed in many industries where accurate forecasting is essential.

1.1 Motivation

Traffic congestion is a major transportation problem exacerbated by increasing car ownership and urbanization. It is probably that better methods, such as road expansion, are required to mitigate traffic issues. The intelligent transport system attempts to enhance the efficiency of traffic and the capacity of roads while reducing accident and pollution rates by integrating 5G communication with on-road sensors. Intelligent Transportation Systems involve the provision of accurate predictions about road traffic using historical data and geographical correlations in predicting volume hence aiding in resource distribution and management strategies.

It is true that it is only the interaction of multiple traffic patterns over time that actually defines the accurate prediction of a traffic flow. It is therefore true that while long-range patterns have regular fluctuations and are influenced owing to daily working hours, there are short-range trends introducing irregular variability, affected through unexpected events such as changes in the weather or accidents. Prediction of traffic flow is quite a challenge in view of diverse patterns including seasonal fluctuations, irregular changes, and trend reversals. These latter examples of course clearly indicate the inherently nonlinear character of traffic dynamics, induced by both human-induced and environment-induced factors.

Recent works by research studies have kept up its attention on neural network models which have been explicitly engineered with detailed evaluation of architectures, activation functions, and settings of layers. Indeed deep learning has emerged as an interdisciplinary region of machine learning for being a subarea because the subarea has made out successful data management ability in learning subtle nonlinear inter-relations. Comparisons of deep models have proven that these models are robust, and helpful in predicting traffic flows.

The following chapter discusses several machine learning methods used for traffic prediction and evaluates a variety of models based on their accuracy, dependence on data and hardware, performance metrics, and ability to deal with certain challenges. It presents a complete analysis of theories behind several machine learning algorithms along with their practical implementation, giving valuable insights into how these can be applied to traffic flow forecasting within intelligent transportation systems.

1.2 Research Contribution

- This chapter considers how deep learning techniques can be exploited in traffic forecasting with detailed information about complexities introduced by the dynamic network topology and high mobility of vehicular objects.
- The chapter performs an extensive analysis of various deep learning frameworks, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, to prove the significance of their application in traffic prediction within VANET.
- The chapter discusses the techniques of data preparation that are important for ensuring the accuracy and reliability of input data in deep learning models, such as data cleaning, normalisation, and augmentation.
- This would bring into the use sophisticated approaches, such as federated learning, transfer learning, into the realm of VANET in demonstrating how these could offer some mitigation to the existing data privacy issues in particular areas with limited resources.
- It comprises the applications of deep learning on VANETs along with demonstrating how the use of deep learning models might help in fighting traffic congestion and optimizing the traffic flow on roads.
- It focuses on hybrid models that incorporate several deep techniques of learning to enhance both spatial and temporal traffic predictions.
- The future investigations, in conjunction with deep learning models in combination with emerging technologies on the edge computing and autonomous vehicles, will lead the way toward a more scalable, real-time traffic forecasting.

2 Fundamentals of VANETs

ITS require the establishment of VANETs. By establishing communication between vehicles, they improve driving conditions as well as traffic flow besides improving road safety. Some aspects of VANET discussed include communication protocols, architectures, and components of these systems, their characteristics besides their challenges in deployment.

2.1 Architecture and Components of VANETs

Vehicular Ad-hoc Networks are a part of an Intelligent Transportation System as this technology will provide communications between vehicles and roadside infrastructure. Structural framework: improve road safety and optimize traffic efficiency, thus improving the driving experience.

2.1.1 Architecture of VANETs

There are three key components of VANETs: vehicles, roadside units (RSUs), and the backend network or central infrastructure. To better describe that, the very basic architecture of VANETs has been represented diagrammatically in Fig. 2.

1. **Vehicles**: These are equipped with On-Board Units (OBUs) that allow them to communicate with other vehicles (V2V) and with RSUs (V2I). Modern cars also come equipped with sensors, GPS units, and other data-gathering and data-transmitting technologies.
2. **Roadside Units (RSUs)**: These units are stationary and sit beside the infrastructure of the roadway. They allow information about traffic signals, roadway conditions, and other data to be communicated using vehicles. In addition, roadside units have connectivity to the basic infrastructure.
3. **Central Infrastructure/Backend Network**: Data centres, traffic management centres, and other cloud services that gather, handle, and evaluate data from cars and RSUs fall under this category. The backend architecture facilitates sophisticated calculations, long-term data storage, and the distribution of more comprehensive traffic statistics.

2.1.2 Components of VANETs

1. **On-Board Units (OBUs)**: devices that are mounted in cars to facilitate communication. They consist of several sensors, GPS, and communication units.
2. **Roadside Units (RSUs)**: Infrastructure components that communicate with OBUs by offering roadside information services and data relay.
3. **Sensors**: gadgets that gather information on the position, direction, speed, and state of the environment of a vehicle.
4. **Communication Interfaces**: Through the use of wireless communication technologies, they enable data interchange between cars, RSUs, and the central infrastructure.

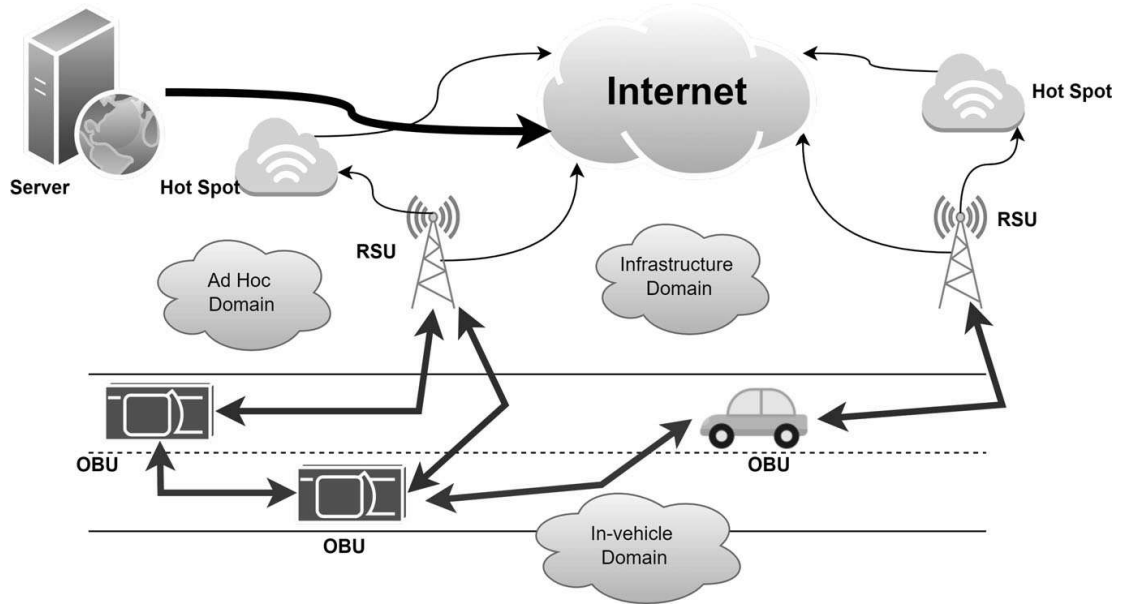

Fig. 2 Basic architecture of VANETs

2.2 Communication Protocols in VANETs

Several protocols are used in VANET communication to provide dependable and effective data sharing. These protocols are intended to meet the particular needs of communication between vehicles.

2.2.1 Types of Communication Protocols

1. **Medium Access Control (MAC) Protocols**: By controlling data transmission across wireless media, these protocols provide unhindered communication between several devices. IEEE 802.11p, which was created especially for VANETs, and DSRC (Dedicated Short-Range Communications) are two examples.
2. **Routing Protocols**: Certain routing protocols, such as AODV, DSR, and GPSR, are essential in vehicular ad hoc networks (VANETs), because cars move quickly and network circumstances change often. By starting route discovery only when required, AODV minimises overhead by creating routes on demand. DSR does not need to keep a global route database; instead, it caches routes and swiftly adjusts to dynamic network changes. GPSR forwards packets to the closest neighbour en route to the destination based on the user's geographic location. These protocols maximise the effectiveness of data transfer, which is essential for secure applications on VANETs and dependable communication.
3. **Transport Protocols**: Data transfer is dependable because to these protocols. With modifications for the high mobility and variable network circumstances of VANETs, TCP and UDP are widely utilised.

4. **Application Protocols**: These specify the precise communication needs for many applications, including entertainment services, traffic management, and safety warnings.

2.3 Characteristics and Challenges of VANETs

2.3.1 Characteristics

1. **High Mobility**: High-speed vehicle movement causes abrupt changes in network architecture.
2. **Dynamic Topology**: Robust and adaptable communication mechanisms are necessary since vehicle locations change often.
3. **Large-Scale Networks**: Vast regions may be covered with VANETs, particularly in metropolitan settings.
4. **Heterogeneous Communication**: Infrastructure components and vehicles may require and have varying communication capabilities.
5. **Real-Time Communication**: Low-latency communication is necessary for many VANET applications, including collision avoidance.

2.3.2 Challenges

1. **Scalability**: ensuring that there is no appreciable performance reduction when a high number of cars are handled by the network.
2. **Security and Privacy**: preventing unwanted access to data and guaranteeing user privacy.
3. **Interoperability**: ensuring efficient communication between various systems and devices.
4. **Reliability**: sustaining constant communication in the face of extreme environmental change and great mobility.
5. **Latency**: reducing data transmission latency to enable real-time applications.
6. **Spectrum Management**: effectively using the little amount of spectrum that is available to prevent interference and congestion.

3 Traffic Prediction Model Based on ML

This section will analyse the various models employed in traffic prediction, specifically focusing on example-based, kernel-based, and regression approaches. This presentation will encompass various forms of machine learning models, emphasising

those that have emerged over the past decade. To enhance the performance of these models in practical applications, we will implement them in prediction scenarios and examine various organisational structures.

3.1 Regression Model

Traffic prediction makes use of regression models, which are frequently connected to parametric approaches. If a technique is parametric, it means that it makes assumptions about a certain distribution of traffic patterns. Because they are easy to develop and operate well for simple traffic network prediction tasks, regression models are used.

According to reference [19], the parametric method entails specifying input and output parameters inside of a mathematical model in which each parameter is directly connected to incoming data. Undefined parameters are found by analysing training data, highlighting how important factor selection and mathematical model selection are to obtaining precise predictions. Research has demonstrated that because parametric approaches have strong theoretical and mathematical underpinnings, they produce projections that are relatively accurate. When it comes to traffic forecasting, linear regression is the most used regression model. To provide accurate prediction results, researchers carefully choose weight parameters and pertinent traffic characteristics. The aforementioned predictions are obtained by linear combinations of obtainable traffic data, so highlighting the effectiveness and resilience of linear regression models in producing accurate traffic forecasts grounded in empirical data.

Rice et al. demonstrated in their work [20, 21] that a linear regression model can be effective even with a small dataset. The method proposed does take into account the fact of temporal variations in the model through the use of time-varying coefficients in adjusting parameters. Results were found to be plausible compared to the k-nearest-neighbors (k-NN) method, which involves two closest neighbors over 20 min. The study discussed that the both predictiveness and usability need to be improved rather than just focusing on the optimization of accuracy measures despite no particular accuracy statistics. This means that linear regression models work well within traffic forecasting contexts, especially within fluid environments that provide adaptations for time-related fluctuations.

Researchers conducted their study by utilizing linear regression analysis [22] to identify spatio-temporal relations that existed in a road network while predicting traffic flow patterns. They wanted to examine whether upstream traffic conditions did impact the flow of downstream traffic on a given link. The results clearly presented that upstream traffic greatly influences downstream traffic flow. The authors integrated the historical averages, upstream traffic data, and real-time traffic conditions using pre-defined weighting factors into three coherent predictive models that they presented as a solution to the problem in question. The authors also designed a method for dynamically adjusting the weight parameters based on the real-time traffic data. Validations with peak-hour traffic data proved the models sound and effective in capturing the complexities of traffic dynamics.

The model beat the historical average (HA) and basic judgement guidelines within the 15-min prediction window, according to the authors' studies. However, the simple linear regression model had a few issues that needed to be fixed. First, over intervals longer than 30 min, prediction accuracy fell short of the historical mean. Second, because these characteristics were limited to certain road segments, it was expensive to determine a set of relevant parameters that would sufficiently describe the link between traffic patterns. Finally, because the model is time-consuming when dealing with intricate road layouts and increasing traffic loads, processing time grew tremendously.

In [23], Kwon et al. coupled linear regression with a tree-based approach and a stepwise variable-selection strategy to speed up the laborious process of choosing appropriate parameters. They produced forecasts using information from occupancy, traffic flow, and historical journey times on the I-880 motorway. The model was evaluated by the authors using cross-validation (CV) over a range of prediction horizons, from 0 to 60 min, in order to solve intersection difficulties. They discovered that while historical data was still helpful for longer-term estimations, current traffic information greatly improved short-term estimates. Changing the depth of the tree significantly improved the efficiency of parameter discovery in the tree-based method. Nonetheless, the model's efficacy was significantly reliant on the initial search range, highlighting the need of a thorough comprehension of historical data unique to the road segment.

The analysed studies highlight the limitations of fundamental linear regression models in accurately capturing the nonlinear variations inherent in traffic flow dynamics. To enhance the accuracy of predicting stochastic fluctuations in traffic patterns, it is common for researchers to integrate regression with alternative methodologies. This approach acknowledges that while linear regression effectively models linear relationships, it may fall short in capturing the complex, nonlinear dynamics inherent in traffic flows. By integrating complementary methodologies, researchers aim to enhance the predictive capabilities of their models, thereby more effectively addressing the intricate and dynamic nature of traffic conditions.

The amalgamation of Granger causality theory along with the dual Lasso regression model address as a nonlinear problems in traffic data by the investigator cited as [24]. Using historical averages, they preprocessed the data to find lingering patterns and used the first Lasso regression model to remove irrelevant data. Through efficient management of both typical and unusual traffic situations, this strategy sought to enhance traffic flow forecast.

In [25], the authors developed a latent component model specifically for short-term traffic flow prediction using bi-linear Poisson regression. The accuracy and robustness of the model were enhanced by including contemporary historical data from critical route portions. In the bi-linear Poisson regression framework, a convolutive mixture was used to address temporal interdependence among different route segments. Furthermore, the authors included a stochastic variational Bayes model, enhancing the system's efficacy for real-time updating and predictive tasks, hence augmenting its applicability in dynamic traffic scenarios.

Table 1 enumerates the regression models used to predict traffic during the last

Table 1 Regression models applied in the last 10 years to forecast traffic

Papers	Work	Regression model	Area	I/O	Comment (if any)
[25]	Flow	Bi-linear	Urban	Multi-multi	Traffic prediction; Bayes method optimisation using stochastic model variations
[24]	Flow	LASSO	Free-way	One-one	Granger causality theory

decade. The table indicates that, in recent years, there have been few models for traffic forecasting using regression analysis. The fundamental nature of the regression model complicates the precise characterisation of the temporal dimensions of continuous traffic data, albeit its straightforward implementation. Regression modelling remains a viable alternative for less computationally intensive traffic prediction tasks in small, simply organised traffic networks.

Nonetheless, it is evident that the regression model has an intrinsic flaw. Regression modelling may be used to create traffic prediction models with a limited amount of historical data. It is suitable for applications requiring fast execution and those with little historical data. Conversely, it reduces the model's accuracy potential due to its vulnerability to overfitting and its inadequacy in delivering a comprehensive analysis of the dataset, including seasonal variations.

3.2 KNN Model

As previously outlined in Sect. 1, our methodology for traffic prediction emphasises example-based techniques, namely the KNN model. This model is preferred for its capacity to delineate spatial linkages among road segments in traffic networks. The k-nearest-neighbor (KNN) model is used for prediction and data filtration, guaranteeing dependable forecasting [26]. KNN functions as a non-parametric regression technique based on data insights, in contrast to formal prediction models. It forecasts future values via the K closest neighbours with analogous variable values, highlighting its data-driven methodology for prediction.

This approach collects the data required for forecasting by using historical data. The historical database encompasses a wide range of trends in addition to normal traffic circumstances, representing all possible patterns of traffic history. The quality of this historical information is a major determinant of prediction accuracy, especially in terms of how comprehensively it captures all potential future traffic situations. The programme depends on the relationships between every traffic system component found in the historical data to guarantee accurate projections.

The conventional KNN model is mostly used for classification jobs; however, modifications may be applied to improve its effectiveness in regression scenarios, as outlined in [27]. The integration of non-parametric regression with K-means clustering forms a predictive approach. K-means clustering organises data points into groups based on closeness, highlighting common attributes within each cluster using distance metrics to evaluate similarity. The first part of the K-means clustering technique involves determining the number of clusters (K). Thereafter, sample sites are randomly chosen to determine the first cluster centres. Subsequent data points are assigned to the next cluster centre according to proximity. Cluster centres are iteratively refined by calculating the mean of the data points allocated to each cluster until convergence is attained. Data points in proximity are chosen for the prediction phase according to their proximity to the current state vector. The neighbouring sites within each cluster are used to predict the subsequent time step. This method utilises interrelated data points inside each cluster for prediction, using clustered data structures to improve forecasting precision.

The authors anticipate traffic flow at 15, 30, 45, and 60-min intervals using KNN [28]. They specifically used previous data collected on Fridays and Saturdays for a particular segment of the route to construct their dataset. The research meticulously investigated the impact of dataset length and neighbour count on prediction accuracy, using various weekdays and prediction intervals.

To improve the process of picking K candidates in the KNN model, the authors of [29, 30] proposed the correlation coefficient distance as a measure for identifying neighbours who are more closely related. The goal of this strategy was to highlight how crucial temporal correlation data is to the prediction model. In addition to the data recordings themselves, the time interval from the present time-which was constrained to a predefined threshold, usually not exceeding one day-also had an impact on the distance between neighbours. The authors substituted a local minimum method for the traditional selection process when choosing prediction candidates in order to address the problem of dimensional reduction of K as a result of overlapping occurrences. The last forecast was refined using a Linearly Sewing Principal Component approach (LSPC), which basically turned it into a minimization issue. Their model fared better than other current models in comparison experiments conducted over different route segments.

The temporal and geographical correlations datasets are important while dealing with traffic flow predictions. For this, the MapReduce-based KNN model was developed by [31]. Unlike all previous KNN techniques, this framework introduces geographical variables. Both the upstream and downstream components constitute the base for distance metric computation. In addition, the authors introduced the Distance Weighted Voting, and this resulted in a decrease in the value of **K** in the KNN framework. This method assigns relevance to the contributions of each neighboring entity according to their proximity toward the target location, taking into consideration the various levels within the predictive framework.

For using the spatial-temporal features of their data in traffic forecasting, the authors in [32] adopted a unique approach. In contrast, in the paper in [31], the considered similar lengths were influenced by the grid distance between selected

Table 2 Progress in KNN applications for traffic forecasting over a decade

Papers	Work	State vector	Distance metric	Method of prediction
[28]	Flow	Historical records target detectors	Euclidean distance	Weighted average
[33]	Flow	Historical records target detectors	Euclidean distance	Weighted average and non weighted average
[29]	Flow	Historical records target detectors	Correlation coefficient distance	LSPC
[31]	Flow	Upstream and downstream historical record and target road	Spatial temporal traffic	Weighted average including trend adjustment
[32]	Speed	The data on grid distance Historical speed records	Gaussian weighted euclidean distance	Gaussian weighted average

road segment and others but the interest in [32] was primarily on segments that lie upstream and downstream. A weighted vector was used to assess the influence of each segment on the concerned road stretch, which enabled a significant reduction in the number of possible predictive options. [32] used a comparison of spatial correlation ratios along with a set threshold to filter out road segments that lacked sufficient spatial connectivity, thereby improving the quality of their selection process. Those candidates for the final forecast were identified by subsequently computing Euclidean distances using a Gaussian methodology. Ultimately, employed a Gaussian weighted average approach to make predictions based on the obtained distances. This approach affords traffic predictions a holistic perspective by gathering critical data across the whole road network, rather than focusing primarily on neighboring road segments.

Table 2 lists the several KNN models that have been applied to traffic condition estimation in the last ten years. These models use traffic data to extract relationships between road segments using the KNN approach. Distances in continuous traffic records or geographic distances are used to characterize interactions, depending on the sort of state vector that is being employed. Adding more historical data to the state vector may result in higher maintenance and computing expenses. Researchers usually use the Euclidean distance metric for state vectors, including single data kinds, such as historical traffic flow records [28, 33]. The weighted distance function would be one of the most effective ways to balance the effects of different kinds of data. It has been observed that the calculation time of KNN models increases with dataset size. While other studies reduce the number of possibilities in the search space by choosing them based on spatial distance, computing demands are still substantial [32]. Although spatial-temporal estimation has advanced, most current forecasts concentrate on individual road segments, which present resource issues when applied to intricate road networks.

3.3 Kernel-Based Model

Vapnik [34] developed Support Vector Machines (SVM), a fundamental component of kernel-based machine learning methodologies. Initially conceived as a versatile statistical learning framework, Support Vector Machines (SVMs) may be used for both regression and classification tasks. Support Vector Machines prioritise structural risk minimisation and are based on statistical theory, distinguishing them from several other nonlinear models and reducing the probability of encountering local optima. Support Vector Regression (SVR) denotes the use of Support Vector Machines (SVM) in regression contexts. Support vector machines (SVR) aim to get maximum generalisation performance by constructing a decision surface, or hyperplane, that maximises the margin between the hyperplane and the nearest training samples while simultaneously minimising the total deviation of training samples from the hyperplane.

Regression problems in traffic prediction frequently call for using the traditional SVM model, which is a non-linear regression technique. In order to map data into a high-dimensional space where a linear hyperplane is constructed, SVM uses kernel functions. This transformation offers a strong foundation to effectively represent complicated interactions in traffic data, enabling researchers to approach traffic prediction challenges similarly to high-dimensional linear regression assignments.

Support Vector Regression (SVR) was initially used in [35] to forecast traffic flow at a particular intersection. With an average inaccuracy rate of 6.03%, predictions were generated for the next hour using data from the five time steps prior. The test data was gathered between 6 and 10 p.m., while the training data was taken between 8 a.m. and 4 p.m. In a different research, cited as [36], SVR was used to estimate aggregate traffic speed during brief periods of time in comparison to a Multi-Layer Feedforward Neural Network (MLFNN) with a Radial Basis Function (RBF) kernel. The past 10 min worth of speed readings were used as input data to forecast speeds for the next 2, 4, and 6 min. When trained on data from the previous two days, SVR performed better than MLFNN, even though MLFNN had greater total accuracy. the gradual increment in the size of datasets signifies the worst SVR's performance and increase in the Mean Absolute Percentage Error however tested in the smaller datasets. A novel parameter, v or v-SVR, was developed by researchers in [37] to improve SVR optimisation for motorway traffic volume prediction. In comparison to MLFNN, our updated SVR model performed better in terms of MAPE (5.5% vs. 8.8%) and Root Mean Square Error (RMSE) (44.7% vs. 64.5%). These studies show how SVR and its variations have evolved and how useful they are for traffic prediction tasks.

A kernel-based model known as Support Vector Machine (SVM) is significantly influenced by the chosen kernel function. The Radial Basis Function (RBF) kernel has been extensively used in research [38, 39]. Research has examined the enhancement of SVM models by including seasonal variations into the kernel function. To include seasonal data from traffic records into SVM predictions, the authors in [40]

used seasonal RBF and seasonal linear kernel functions. These algorithms implemented seasonal interval modifications to temporal data intervals. Comparative trials demonstrated that the seasonal SVM model regularly outperformed non-seasonal kernel SVM models, as well as other models like Artificial Neural Networks (ANN) and ARIMA with Kalman filter. The seasonal SVM model exhibited competitive performance across several time intervals, except the morning peak. It surpassed SARIMA when combined with a Kalman filter. The authors' comparison research demonstrated the seasonal SVM model's competitiveness in both training and prediction phases, despite SARIMA sometimes exhibiting marginally superior accuracy in certain situations and requiring double the training duration.

After choosing a suitable kernel function, parameter optimisation is an important step in SVM models. Low computation time is desired, although convergence to less-than-ideal local minima is to be avoided. The authors of [41] utilised Continuous Ant Colony Optimisation (CACO) as a means of effectively optimising SVM parameters. CACO is more successful than standard Ant Colony Optimisation (ACO) [42] because it works in a continuous search space. The study used SVM with a Gaussian Radial Basis Function (RBF) kernel to anticipate traffic flow during morning (6 am to 10 am) and evening (4 pm to 8 pm) peak hours using data from Taiwanese cities. Based on comparative studies, CACO regularly outperformed SARIMA in traffic flow prediction tasks by facilitating effective parameter adjustment over a wide range of input data features.

Unlike conventional SVM techniques, the model developed by researchers in [43] was designed to handle multi-step prediction problems. By including historical records and upstream data into the input structure, their method improves forecast accuracy. When compared to conventional SVM approaches, the model's overall prediction performance is improved due to this integration, which enables the model to more accurately capture dependencies and trends over numerous time steps. The experiment showed that prediction accuracy was greatly increased by combining upstream data with history records, particularly when prediction intervals were small, like 1. This improvement is ascribed to the significant influence that upstream traffic flow had on the particular road stretch in a little amount of time. All models became less accurate as the forecast interval increased from 1 to 3. The availability of additional historical data was beneficial for the prediction job as the prediction interval grew. The authors also pointed out that, in terms of computing efficiency, using input data that was limited to previous and historical knowledge on the particular road stretch that was being considered was the best course of action. This approach maximised prediction performance while minimising computational resources.

4 Neural Network Models

The distinguishing feature of non-parametric models, particularly those based on neural networks, lies in their ability to learn parameters directly from historical data rather than relying on pre-established values.cite41. Non-parametric models differ

from traditional models with fixed parameters, offering significant advantages in today's data-rich environment due to their ability to adjust to the patterns present within the data. By utilising prior data to enhance prediction accuracy, these models establish a direct connection between predictive objectives and historical data.

Previous research indicates that non-parametric techniques, such as Support Vector Machines (SVM), significantly enhance algorithmic accuracy compared to traditional linear regression models by effectively capturing the non-linear characteristics of traffic patterns. Nonetheless, non-parametric methods are associated with certain disadvantages [44]. A substantial amount of dependable historical data is required, which can prove challenging to locate and maintain current. Moreover, in comparison to alternative methodologies, the training of non-parametric models may incur higher costs.

The 1940s saw the introduction of the neural network (NN) model by Warren S. McCulloch and Walter Pitts [45, 46], which is highly effective at handling complicated linear systems. As a black-box learning model, NN learns and adapts the input-output mapping on its own from large datasets, in contrast to standard empirical formulae. For traffic prediction jobs, its durability, associative memory, and continuous real-time parameter updates make it the perfect choice. To provide precise forecasts, the NN model incorporates important information from the study road segment and historical records with factors like weather, road state, accidents, and the effect of development on the traffic network.

A large quantity of data is needed to train a neural network, and not enough data might result in predictions that are off. The neural network is less adaptive to shifting traffic and road conditions since its applicability is usually restricted to the particular road section it was trained on. Furthermore, choosing the right amount of neurons for each buried layer is difficult yet essential. While too few neurons may reduce prediction accuracy, too many might lead to a huge network topology and longer calculation times. These elements add to the neural network's possible drawbacks in real-world traffic prediction situations.

4.1 FFNN

Neural networks are fundamentally composed of artificial neurons, each of which has a characteristic structure seen in Fig. 2. The input vector to a neuron may be described in the Eq. 1 as $\mathbf{z} = [z_1, z_2, \ldots z_d]$ if we suppose that the neuron receives d inputs $z_1, z_2 \ldots, z_d$. The neuron's output will be:

$$a = f\left(\mathbf{k}^T \mathbf{z} + b\right) \tag{1}$$

where, $\mathbf{k} = [k_1, k_2, \ldots, k_d]$ is a weight vector of dimension d. b is the bias variable and $f()$ is denoted as the activation function. The most commonly used activation functions are described in Table 3.

Table 3 Activation function

Name	Activation function
Logistic	$\sigma(X) = \frac{1}{1+\exp(-X)}$
tanh	$\tanh(X) = \frac{\exp(X)-\exp(-X)}{\exp(X)+\exp(-X)}$
ReLu	$\mathrm{ReLU}(X) = \max(0, X)$

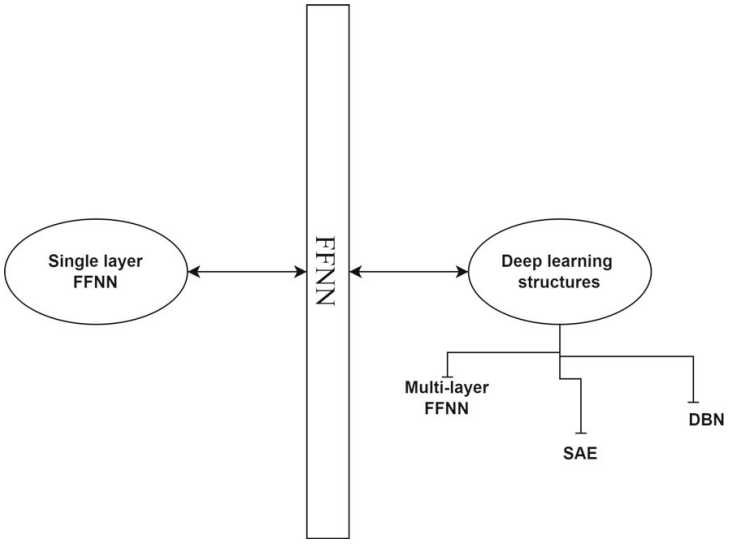

Fig. 3 Advancements in feed-forward neural networks (FFNN): state of the art

Analogous to the brain's layered information processing, the FFNN arranges neurones into consecutive layers for information processing. Each layer utilises the output of the previous layer as its input and then transmits its output to the subsequent layer. Feedforward neural networks (FFNNs) operate with unidirectional information flow, but recurrent neural networks (RNNs) have feedback loops that provide reverse information circulation. Comprehensive calculations demonstrating the internal mechanisms of FFNNs are provided in [16].

Several feed-forward neural network (FFNN) based traffic prediction methods are shown in this section. Additionally, we will look into a number of deep learning frameworks that can greatly raise FFNN's accuracy. A thorough taxonomy of the evaluated state-of-the-art FFNN literature is shown in Fig. 3.

A straightforward and obvious topology for a feed-forward neural network (FFNN) consists of a single hidden layer. In a study analogous to [47], researchers used a feedforward neural network (FFNN) to predict traffic flow with a single hidden layer architecture. The primary objective was to forecast traffic flow both at the present moment and within a specified time range that begins and concludes at a future time T. The MTL FFNN sought to enhance the accuracy of the primary prediction at time T by including supplementary tasks associated with adjacent time periods, leveraging correlations across various time intervals. This enhancement illustrates how the appropriate integration of many interconnected systems may enhance the precision of traffic flow predictions.

In the paper [48], a study has been conducted on short- and long-term traffic forecasting ability of a FFNN. A FFNN has been trained to predict the traffic flow with the help of 20% of the dataset using the input from weekday, month, and previous time step traffic volume data. It is seen that FFNNs are intuitive and fail to predict traffic flow during severe congestion. The failure to identify complex nonlinear patterns may explain this. The need for increasing the model complexity and addressing heterogeneity in the dataset arise in using FFNNs to solve real-world problems such as traffic prediction issues.

In [49], shallow FFNN was used in predicting the traffic flow on a route accommodating different kinds of vehicles. The inputs the FFNN used were numerical data on several vehicle categories, temporal information on the day and time, and speed metrics for each vehicle. Notable conclusions were that sigmoid activation is better than the tanh function, and Levenberg optimizes better than momentum optimization in training the network. Furthermore, the predictive precision enhanced by the number of hidden neurons inside the feedforward neural network, which in turn indicates that the architecture of a network plays an important role in modeling the complex interrelations of the data properly. The findings established that the addition of selected data, being the records of vehicle speed improved overall performance of the model and indicated that full contextual information was vital for efficient modeling of traffic. Findings show how diverse sources of data need to be combined, and parts of the model fine-tuned to facilitate enhancement of FFNN-based systems for traffic in better management and planning of traffic.

4.1.1 FFNN: Deep-Learning Structure

In order to forecast traffic velocity based on observations from spatially coupled sensors, researchers utilized a multilayer Feed-Forward Neural Network (FFNN) as described in [50]. The initial trend of the input was preconditioned using an l_1 trend filter in an effort to reduce noise effects, then the LASSO model was applied for identifying road segments with geometric correlations. The FFNN showed reliable performance in predicting traffic speed in both interrupted and normal conditions due to the efficient application of spatial-temporal filters.

Nevertheless, the model's efficacy decreased as the network's depth increased. In order to get around this, researchers in [51] trained a multi-layer FFNN using an unsupervised training strategy using Wasserstein Generative Adversarial Network (WGAN) techniques [52]. Following the detrending of data using historical averages, the model was created to forecast in both common and uncommon circumstances. The model does not outperform LSTM in common scenarios or SVR and ARMA in exceptional ones, even though it functions satisfactorily in normal conditions. Nonetheless, the writers point out that one of its noteworthy strengths is its adaptability to different settings without depending on a single approach. Despite its complexity, the model's training time is also noticeably less than that of other deep learning techniques.

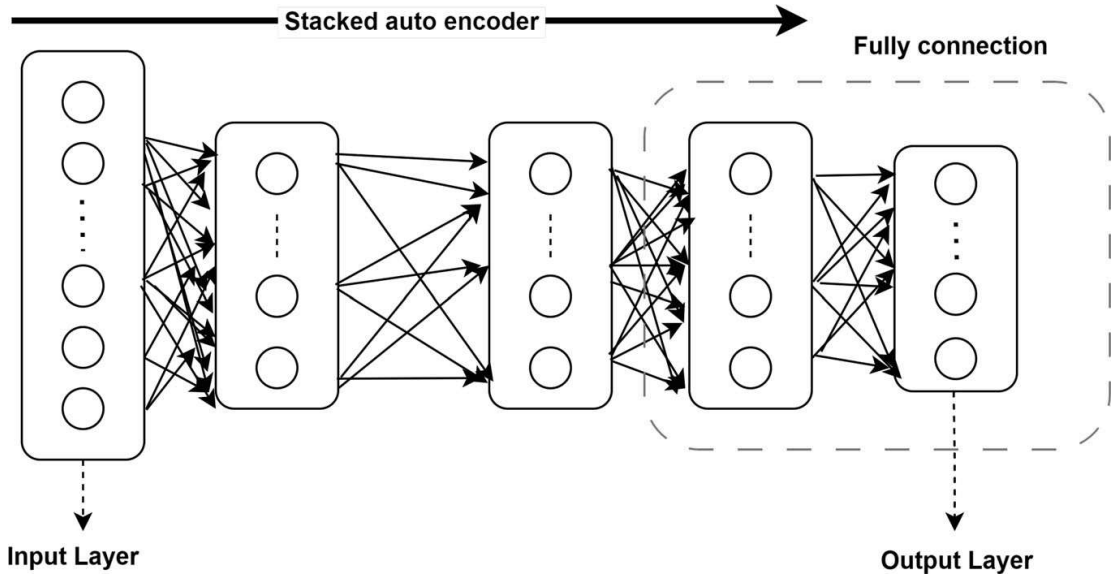

Fig. 4 Structure of SAE

One popular deep learning architecture for traffic prediction problems is the Stacked Autoencoder (SAE). As seen in Fig. 4, it comprises of several layers of autoencoders (AE). In order to extract higher-level characteristics that improve prediction accuracy, each AE layer gradually decreases the dimensionality of the input data. SAE is well-known for its capacity to minimise computing costs and effectively extract significant features from training data [53].

According to [53], autoencoders have three layers: input, hidden, and output. The buried layer receives data like an encoding technique that discards input vector attributes. The autoencoder then copies the input vector from the hidden layer to the output layer to decode these properties. The goal of unsupervised autoencoder training is to minimise the difference between the input and output vectors. This variant shows how AE extracts functional features. Let's analyse the hidden layer's output, Y, and the input vector, $Z = [z_1, z_2, \ldots, z_N]$. Equations 2 and 3 of the Autoencoder (AE) are relevant:

$$M = f\left(G_e Z + b_e\right) \tag{2}$$

$$L = f\left(G_d D + b_d\right), \tag{3}$$

Afterwards, by minimising the cost function, which is defined as follows, we may optimise the AE in Eq. 4 as

$$P(Z, L) = \frac{1}{2} \sum_{i=1}^{N} \|x_i - z_i\|^2 \tag{4}$$

The initial stage in employing the Stacked Autoencoder (SAE) model involves pre-training from the bottom up. Upon receiving the input vector, each layer of the autoencoder (AE) systematically learns to reconstruct it. The subsequent AE layer receives its input from the output generated by the hidden layer of each AE. The weights and biases of the fully linked prediction layer are initialised subsequent to the training of each autoencoder layer. Finally, a top-down backpropagation (BP) method is employed to optimise the entire network.

In [53], SVM, FFNN, and RBFNN were studied to assess the Stacked Autoencoder (SAE) model's ability to estimate traffic flow at 15, 30, 45, and 60 min intervals. The SAE model outperformed other prediction models with a mean relative error of 6.50%, according to the research. The SAE model's accuracy remains mostly intact when the forecast interval is extended. DBN is a deep-learning architecture like SAE for FFNN. The system consists on restricted Boltzmann machines (RBM), which are energy-based undirected graphical models [54, 55]. DBN, unlike SAE, includes non-interconnected units at every tier, providing the image of a completely connected network. DBN and SAE are comparable because RBM layer hidden units are inputs to the next layer. The SAE and DBN training methods are similar.

Huang et al. presented Deep Belief Networks (DBN) in [56], outperforming FFNN, SVR, and ARIMA models with over 87% accuracy over a range of prediction intervals. By fine-tuning DBN's parameters using the FireFly Algorithm (FFA), a different research [57] showed increased prediction accuracy and lower processing costs when compared to previous approaches.

4.2 Recurrent Neural Network (RNN)

Neurons in the same layer of a feed-forward neural network (FFNN) solely forward information from input to output; they do not exchange messages. Because of their construction, FFNNs can only produce output by using the current as their input. Nonetheless, outcomes in traffic prediction tasks frequently rely on both current and previous conditions. This restriction was addressed with the development of Recurrent Neural Networks (RNNs). Sequential relationships in traffic data are well captured by RNNs, enabling more precise modelling of how previous states affect future forecasts [58].

according to [59–61]. Because of their "short-term memory," RNNs are unique in that their neurons can process inputs from both the present layer and layers that came before it. Compared to Feed-Forward Neural Networks (FFNNs), RNNs are superior at detecting temporal correlations in traffic data because of this capacity. A thorough taxonomy of the literature on RNNs is probably depicted in Fig. 5, which also highlights the many designs, uses, and developments in the subject.

The authors of [62] conducted a thorough analysis of LSTM's effectiveness in short-term traffic flow prediction. To optimise hyperparameters such the number of hidden units in the LSTM layer and the duration of input traffic flow records, they used a single hidden layer LSTM NN model and grid search. For each of the thirty

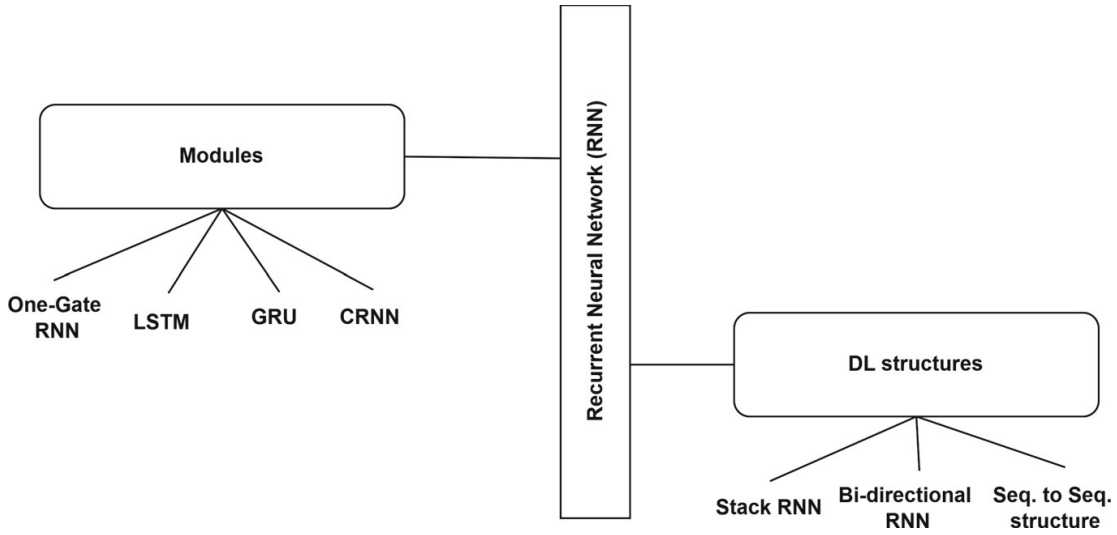

Fig. 5 RNN: state of the art

detectors, they used characteristics like MAPE and RMSE to compare LSTM against models like as Random Walk (RW), SVM, FFNN, and SAE. According to hyper-parameter analysis, the model's accuracy peaked at about 20 hidden units, and as it increased, it became less accurate. Interestingly, the LSTM NN model significantly outperformed earlier models in that it kept consistent accuracy throughout a range of input data lengths. The dataset's seasonal bias, which affected experiment stability because of noticeable seasonal fluctuations in traffic circumstances, was a weakness of the study. Test data concentrated on winter circumstances, whereas training data spanned three different seasons in 2014.

Hochreiter et al. [63] introduced LSTM in 1997 to solve long-term dependence problems in RNNs. In contrast to conventional RNNs, LSTM has input and output gates. Later, Gers et al. [64] observed that the original two-gate LSTM was occasionally ineffective with longer input time series because it oversimplified non-linear characteristics. To overcome this, Gers et al. added a forget gate to the two-gate LSTM design, leading in the creation of a three-gate LSTM module, as shown in Fig. 6.

Cho et al. (2014) created the Gated Recurrent Unit (GRU) [65], which is frequently utilised in traffic prediction because it can manage long-term dependencies with less complexity than LSTM. GRU was used in [66] to anticipate traffic flow at short intervals in both rural and urban locations. The performance of GRU was assessed using MAE and MSE metrics in comparison to RBF after the data were normalised using Min-Max normalisation. The findings demonstrated that, in both urban and rural datasets, GRU consistently outperformed RBF in terms of MSE and MAE values. With a little drop in accuracy seen exclusively in metropolitan areas, the model outperformed RBF and normal iteration approaches when employing the auto-iteration method for multi-step predictions.

In [67], traffic volume information from 30 sensors in the PeMS dataset was used to compare GRU and LSTM. Based on data from each sensor collected over the previous 30 min, both models were trained to forecast traffic flow for the next 5

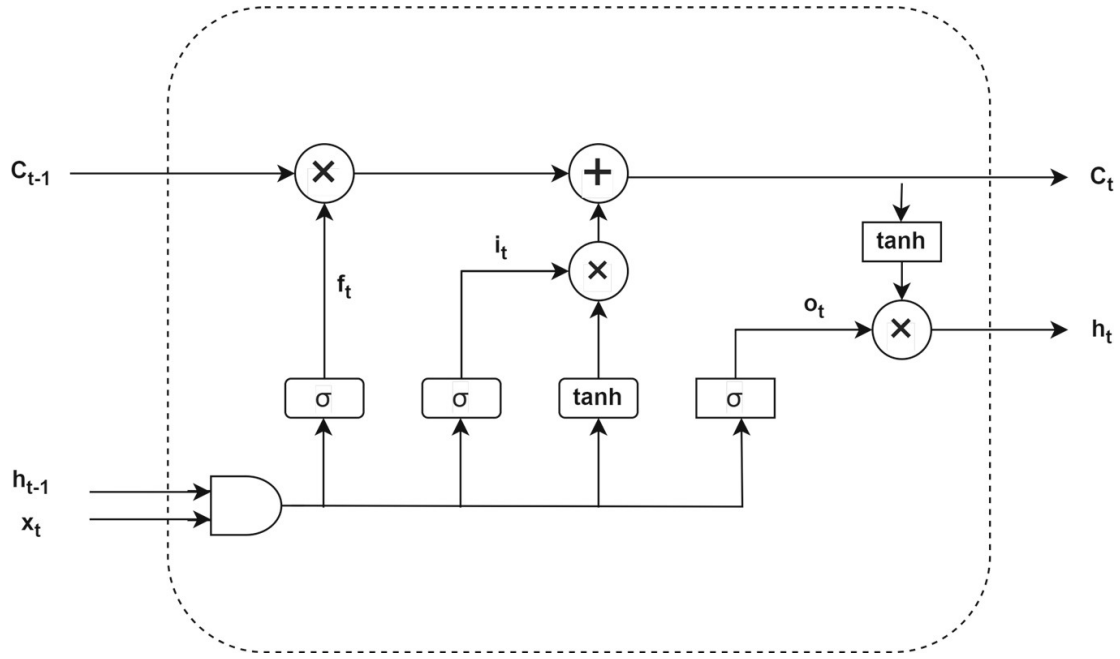

Fig. 6 Repeating module in LSTM

min. The study discovered that for predicting traffic flow, GRU and LSTM performed better than ARIMA. Although there were not many variations in overall performance between the two RNN designs, GRU demonstrated more stability in error rates across sensors when compared to LSTM.

In single-layer designs, a number of well-known RNN modules have undergone substantial testing for traffic prediction. RNNs do, however, also have a deep learning basis. In the next parts, the course will discuss many popular deep-learning RNN topologies.

4.2.1 Deep Learning Architectures for RNNs

RNNs were incorporated into deep-learning frameworks by Pascanu et al. [68] and Hihi et al. [69] in order to fully use their potential. As noted in [58], deepening the RNN structure makes it possible to extract more abstract characteristics and patterns from data. This improvement dramatically improves prediction model accuracy and performance on challenging prediction tasks.

The Stacking RNN (SRNN) is a deep-learning recurrent neural network architecture, as seen in Fig. 7. Information is sent uniformly across each of the several stacked RNN layers of an SRNN. Figure 7 illustrates that N represents the total number of layers in an SRNN, whereas L_i^j denotes the j-th RNN module at timestamp i. Assuming that h_i^j represents the output of the layer at time i and index j, we may calculate h_i^j using Eq. 5 as follows: