Received: December 28, 2023. Revised: January 17, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Prediction of Vehicular Traffic Flow Using Optimized Neural Network

Ahmed Ibrahim Turki[1]* **Saad Talib Hasson[2]**

[1]*Department of Physics, University of Samarra, Samarra, Iraq*
[2]*Information Networks Department, University of Babylon, Babil, Iraq*
* Corresponding author's Email: ahmed.ibrahim@uosamarra.edu.iq

Abstract: Intelligent transport systems (ITS) include a broad range of applications that require proactive strategies and predictive data driven by artificial intelligence and big data. The objective of this paper is to improve the accuracy of traffic flow prediction by utilizing a novel approach that combines feedforward neural-networks with the Quasi-Newton (QN) optimization method. The proposed method decreases the error factor based on the lagrange multiplier and Jacobian vector. This enhancement has resulted in a faster convergence during the learning process. The sample was chosen utilizing the dataset provided by the England Highway (HE) traffic monitoring systems in 2023. In order to assess the proposed model, the research findings are compared to other standard prediction techniques. As a regression model, the proposed optimized multi-layer perceptron neural network method achieves an average root-mean-squared-error of 0.143 compared to 0.319, 0.459, and 0.406 achieved by (random forest, Naïve bays, k-nearest neighbour) respectively. That is, the proposed model achieved an average percentage of improvement in prediction of are approximately 55.17%, 68.81%, and 64.78%, respectively, compared to other standard techniques. Finally, the superiority of the proposed model was evaluated by the coefficient of determination (R^2) and mean-absolute-error measure, and its performance was better than other forecasting techniques as well.

Keywords: Traffic flow, Vehicular networks, Developed neural network, Optimization, Prediction, Regression, Machine learning.

1. Introduction

Intelligent cities can improve the management of resources and the living conditions of citizens, with ITS playing a crucial role [1]. The integration of communication systems and on-road sensors [2], allows for real-time analysis of road conditions, improved efficiency in traffic control [3], and the provision of fast vehicular cloud services. Deploying ITS enhances the safety, sustainability, and convenience of road environments [4], thereby requiring accurate traffic flow prediction systems. These systems provide valuable information for ITS applications such as vehicular ad hoc networks (VANET), vehicular cloud computing, traffic light control, and congestion management [5]. The projected traffic flow is a crucial indicator for designing vehicle routes, thereby aiding travellers in making more informed decisions regarding their chosen paths. Anticipating the timing and location of congestion is highly beneficial for transportation management, as it enables experts to allocate resources to the roads when congestion is likely to occur, ultimately reducing traffic congestion. As a result of its significant superiority over different devices, traffic flow prediction [6, 7] has emerged as a prominent area of study in recent years. In essence, predicting traffic flow involves making assessments about future conditions based on information and expertise derived from related historical data. Consequently, the techniques used for transmitting, collecting, mining, and storing data have a significant influence on prediction methods [8, 9].

There are several conventional traffic forecasting methods, such as the Kalman filtering model, historical average, Gaussian process-based, and others [10]. Nevertheless, the fluctuating patterns of traffic movement and the results achieved through these methods are still inadequate in terms of accurately predicting future outcomes [11]. In recent times, deep learning techniques have demonstrated promising results in a dynamic prediction model. Deep learning models have become popular because of their exceptional capacity to acquire a wide range of feature levels from temporal- spatial traffic data [12].

1.1 Problem statement

The highway England system provides nearly real-time traffic information for the majority of highways, motorways, and major A roads in England. The displayed traffic information provides the average traffic volumes for each intersection in both clockwise and counter clockwise directions on the respective highways. The data sources consist of loop detectors and microwave sensors that are deployed onsite at specific locations along the road. There is an issue implied by the current implemented system:

Currently, the system is unable to make predictions based on the current data. It only shows the average speed in real-time and generates control signals that determine the delays for the electronic signs on the roads. Therefore, it is necessary to utilize advanced AI-based deep machine learning techniques in order to accurately forecast traffic flow by analysing the behaviours of closely related data links. Hence, to alleviate the afore-said issue, this paper proposes construct an elastic forecasting model for highway traffic prediction using the optimized multiple layers perceptron neural networks technique. In order to achieve the aforementioned research goals, utilizing technical indicators commonly used in financial trading management and control to advantage their potential effectiveness in short-term predicting traffic flow. As well as, this study enhances the rate at which standard multilayer perceptron neural networks (MLP-NN) are trained using the back-propagation (BP) algorithm. This is accomplished by substituting the gradient descent method with the Quasi-newton method, resulting in improved convergence speed.

1.2 Contributions

The main contribution of this work is as follows:

i. Develop a novel prediction model that utilizes specific technical analysis indicators as input features to accurately forecast traffic flow within a reasonable timeframe.

ii. This study is the first to utilize technical analysis indicators as inputs for the regression models being proposed. The rationale for incorporating these indicators lies in their robust explanatory capabilities and ability to provide accurate short-term predictions.

iii. The findings of this study have substantial ramifications for transportation agencies and researchers. These findings can be used as a dependable foundation for making well-informed decisions aimed at mitigating traffic congestion within the vehicular network.

The subsequent sections of this document are structured in the following manner. Section two provides an overview of the existing research on different data mining techniques used for predicting traffic flow. Section three presents the proposed research methodology for constructing the regression model. The fourth section provides an explanation of the proposed system. Section five presents and analyses the research findings. Section six ultimately presents the main discoveries of this research.

2. Related work

This paper presents a comprehensive examination of the latest methodologies employed in traffic flow prediction through the utilization of machine learning.

In [13], a fuzzy-oriented traffic flow prediction method was introduced. This study used a fuzzy approach in the deep learning model to reduce data inaccuracies. The fuzzy deep convolutional network (FDCN) was also proposed to improve traffic flow prediction.

One dataset was partitioned into multiple sub-datasets and allocated to different computational nodes in [14]. In addition, "master-slave parallel computing" was implemented, where slave computing nodes acquired features and transmitted them to the master.

An improved scheme was introduced in [15], which established a connection between the significant influence of longer sequence time steps and the current time step. The attention strategy captured traffic flow values with significant impact.

Short-term traffic flow prediction is crucial for many applications, so [16] sought to improve it. They added prediction intervals to the K-NN model to account for uncertainty. They also tested traffic flow rate measurements at 3–30-minute intervals due to traffic stochasticity. K-NN outperformed the benchmark model for point prediction accuracy in short time intervals (less than 10 minutes), suggesting its suitability for traffic flow prediction.

Spatial and temporal attentions predict traffic flow using deep learning [17]. To identify spatial relationships between road segments and temporal relationships, temporal and spatial attention were used.

MIDAS, a reinforcement learning-based method for urban autonomous navigation, was introduced in [18]. To let an Ego agent, influence other vehicles, MIDAS was created. It used an attention mechanism to handle multiple agents and a "driver-type" parameter to improve planning. Extensive experiments validated the method's road scenario adaptability. It demonstrated the ability to generate adaptive Ego policies, remain robust to other agents' behaviour, and outperform interaction-aware decision-making approaches in safety and efficiency.

In [19] presented a traffic flow prediction model that combines denoising techniques with artificial neural network (ANN). The objective of this study was to evaluate the precision of various smoothing techniques in eliminating irregularities in traffic flow data prior to utilizing ANN for forecasting short-term traffic flow progression.

Combined CNN and LSTM networks to create an attention-based convolutional long short-term memory (LSTM) module in [20]. This module extracts short-term temporal and spatial features. To analyse traffic, they added a bidirectional LSTM (Bi-LSTM) module that captures daily and weekly recurring characteristics.

The effects of SVM, ANN and KNN were examined in [21]. This study assessed prediction accuracy using five months of traffic flow data. Variability in flow direction before and after a point.

In [22], time-series models were used to predict future traffic flow using real-time data to minimize prediction errors. Using UK highways data, hybrid models for traffic prediction used autoregressive integrated moving average (ARIMA), multilayer perceptron (MLP), and recurrent neural network (RNN). Multiple metrics showed that these hybrid models were effective (e.g., R2 values around 0.94 for peak hours).

Current techniques face some difficulties which will be reviewed in Table 1.

Within this research domain, the aforementioned constraints serve as a driving force for researchers to propose a novel traffic flow prediction model. A solution to address these challenges involves the development of an improved traffic flow prediction model using O-MLPNN. The proposed model employed RSI, ROC, EMA and ATR features in prediction using O-MLPNN, distinguishing it from other similar studies. In addition, the Quasi-newton optimization approach is employed to expedite convergence in the learning process. This model accurately forecasts traffic patterns in diverse scenarios, including weekdays, holidays, and public events, and is applicable to both large and small road networks.

3. The research methodology

3.1 Technical indicators generation

The key notion behind the technical indicator refers to the results of the mathematical computation that applied on historical flow data to direct researchers towards long or short predicting on different time frame configurations [23]. It is important to observe that the subsequent notations are employed consistently throughout the document:

n: represents the period or number of data points considered.
TR_i: represents the true range at time i.
A_n: Range between the highest and lowest flows for n periods
B_n: Absolute difference between the highest flow and the current flow for n periods
C_n: Absolute difference between the lowest flow and the current flow for n periods
EMA_u: Previous exponential moving average.
R: Current data point
D: Smoothing factor
RSI_n: is the relative strength index with a period of n.
D_n: is a variable used in the calculation of RSI.
$Flowup, Flowdown$: represent upward and downward flows, respectively, in the RSI calculations.
ROC: is the rate of change.

(i) Average true range (ATR)

ATR is computed by taking the average of the actual flow ranges observed during a specific time interval.

ATR measures volatility by incorporating any interruptions in the traffic flow [24]. The ATR can be calculated using Eq. (1).

$$ATR_n = \frac{1}{n}\sum_{i=1}^{n} TR_i \tag{1}$$

Table 1. Traffic flow prediction ML algorithms and methods comparison

Ref.	Prediction Techniques	Dataset	Comb. or single use	Prediction condition	Features	Challenges	RMSE (%)
[13]	FDCN	TaxiBJ	Comb.	Urban	- Minimized data ambiguity - Enhanced precision	External factors had an impact on the predictive capabilities of this model.	0.42
[14]	DBN	Performance Measurement System (PeMS)	Comb.	Freeway	- Decreased margin of error - Minimizes the duration of training	Other works do not support the proposed model's performance.	28.80
[15]	RNN	PEMS and Sensor data from selected location	Comb.	Highways	- Minimal errors - Smooths noise.	Must consider space and time	30.36
[16]	KNN	MIDAS	Comb.	Highway	- Maximize reliability	does not consider the time-dependent correlation of prediction intervals	30
[17]	STANN	Sydney Coordinated Adaptive Traffic System	Comb.	Urban	- Precise forecasts - Examines the time-based and location-based characteristics	Reduced efficacy in a vast and intricate traffic network	8.9
[18]	Reinforcement Learning (RL)	MIDAS	Single	Highway	Safer and more efficient	Overly conservative plans	2.7
[19]	ANN	Data from three loop deductive sensors	Single	Freeway	Precision and efficiency in denoising	Prediction error occurred.	9.16
[20]	AT-conv-LSTM	PeMS	Comb.	Highway	Higher prediction accuracy	This model is for small road networks only.	20.5
[21]	SVM, KNN, ANN	Transportation Infrastructure Ireland	Single	Motorway	KNN predicts sunny and rainy states accurately.	Lower accuracy for ANN and SVM models	N/A
[22]	(ARIMA – MLP), (ARIMA – RNN)	MIDAS	Comb.	Highway	reduce traffic congestion	Future traffic flow extrapolation is difficult.	0.84, 0.81

where:

$$TR_i = Max\{A_n, B_n, C_n\} \tag{2}$$

$$A_n = HighestFlow_n - LowestFlow_n$$
$$B_n = |HighestFlow_n - Flow_n|$$
$$C_n = |LowestFlow_n - Flow_n|$$

(ii) Exponential moving average (EMA):

EMA represents the mean rate of traffic movement [25]. The calculation is performed according to Eq. (3). In (3), R represents the traffic flow for the recent period, D represents the smoothing constant which is equal to 2/(nu+1), nu represents the count of traffic flow in SMA estimated by EMA, and EMA_u represents the EMA for prior traffic flow.

$$EMA = R - EMA_u.\mathrm{D} + EMA_u \tag{3}$$

(iii) Relative strength index (RSI)

RSI is a quantitative tool that oscillates between zero and 100, serving as a technical indicator. RSI can also be used to determine the overall trend [24]. The computation of the RSI can be derived using the Eq. (4) provided.

$$RSI_n = 100 - \left[\frac{100}{D_n}\right] \quad (4)$$

where:

$$D_n = \left[1 - \frac{\frac{1}{n}\sum_{i=1}^{n} Flowup[flow_i - flow_n]}{\frac{1}{n}\sum_{i=1}^{n} flowdown[flow_i - flow_n]}\right] \quad (5)$$

(iv) Rate of change (ROC)

The ROC indicator, also referred to as Momentum, is a momentum oscillator that exclusively quantifies the speed of change. The calculation of the ROC entails comparing the current flow with the flow observed "n" instances ago [25]. The calculation of the ROC can be determined using Eq. (6).

$$ROC = (Current\ flow/flow\ of\ n\ bars\ ago) - 1.0) \,.\, 100 \quad (6)$$

These features are subsequently provided as input to the regression model based on the optimized multilayer perceptron-neural networks.

3.2 Prediction techniques of traffic flow

(i) Naive bayes (NB)

Supervised machine learning algorithm NB categorises observations based on its own rules. The NB classifier is supervised because it assumes the training dataset classes are known. Due to its resilience to irrelevant attributes, ease of implementation, and simple training requirements, the NB method can predict driver behaviour and detect misbehaviour in VANET [26].

(ii) Random forest (RF)

RF is simple, fast to train, and can produce results that can be applied to various situations. RFs have autonomous decision trees. Due to dual random selection, every tree has a fragmented problem view. RF reduces overfitting and eliminates feature selection. It requires few inputs. It solves congestion prediction well due to these advantages [25].

(iii) k-nearest neighbour (KNN)

K-NN is supervised learning. This tool is versatile for classification and regression. The K-NN algorithm predicts using the entire dataset. When predicting an observation not in the dataset, the algorithm finds the K instances most similar to our observation. The algorithm then calculates the estimated output value of the observation using the relevant images of these K neighbours [27].

(iv) Multilayer perceptron (MLP)

MLP uses neurons with differentiable functions. The network has hidden layers. The networks' neurons are highly connected due to synaptic weights. Due to their high connectivity and non-linearity, networks are difficult to analyse [26].

Input layer (1, 2, …, n) is the initial layer that sends signals to network neurons. The neural network's intermediary layer is the hidden layer (1, 2, ..., h). The output layer (indices 1, 2, ..., 0) is the last. The system only computes in the hidden layer. The third layer gives that network results. In addition, the network has bias nodes (b_{hidd} and b_{out}).

The output and hidden layers are linked. The network also has synaptic weight values ($W_{in,out}$) between the input and hidden layers. W_{hi} , out represents the network's hidden and output layers' synaptic weights. Supervised MLP training often uses the BP algorithm. Weight adjustments reduced network error [28]. In this study, error is the mean squared error (MSE) between the network's simulated and original output.

3.3 Quasi-newton (QN)

The optimization algorithm aims to iteratively update a weight vector in a neural network to minimize a loss function. It utilizes the Quasi-newton approach, specifically the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula, to efficiently approximate the inverse Hessian matrix. This method helps mitigate the computational expense associated with evaluating and inverting the Hessian matrix in Newton's approach [29-30]. There are many symbols, variables and functions used in the algorithm as in the notation list:

W_k: Current weight vector in the neural network
CF: Converges factor
k: Iteration counter
$Ridge$: Regularization parameter (e.g., 0.01)
G_k: Jacobian (gradient) vector at iteration k
B_k: Symmetric positive definite matrix at iteration k
d_k: Search direction
a: Step size for line search
DW_k: Change in weight vector
DG_k: Change in gradient vector

The algorithm is divided into 6 main steps:

- **Initialization (Step 1):** Set initial parameters, compute Jacobian vector G_k , and initialize matrix B_k.
- **Convergence check (Step 2):** Stop if the norm

of G_k tends to 0.

- **Search direction (Step 3):** Determine search direction $d_k = -B_k^{-1} G_k$
- **Line search (Step 4):** Find step size α using line search and update weight vector.
- **Update matrices (Step 5):** Update G_{k+1} and matrices B_{k+1}, DW_{k+1}, and DG_{k+1}.
- **Iteration (Step 6):** Increment k and return to Step 2.

The QN equation is expressed in Eq. (7) [29].

$$w^{(i+1)} = w^{(i)} - \left(B^{(i)} \text{ x } g^{(i)}\right)\eta^{(i)} \quad (7)$$

where i=0,1,2, …

The learning rate (**η**) can be either pre-set or determined through the process of line minimization.

Finally, the algorithm utilizes the Quasi-newton approach with the BFGS formula to optimize a neural network's weight vector. It efficiently approximates the inverse Hessian matrix, mitigating the computational cost. Parameters such as the learning rate (η), convergence factor (CF), and regularization parameter (Ridge) play crucial roles. The algorithm iteratively updates the weight vector, incorporating line search and matrix updates to achieve convergence.

3.4 Methods of evaluation

This study employs three evaluation methods to assess the prediction models: Mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R2). The formulas for the evaluation metrics [31] are presented in Eqs. (8-10).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Actual_i - predicted_i)^2} \quad (8)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|Actual_i - Predicted_i| \quad (9)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}^2)^2} \quad (10)$$

$\hat{yi}$ is predicted value of y. $\overline{yi}$ is mean value of y.

4. The proposed model

The architecture of the proposed model for traffic flow prediction is depicted in Fig. 1, illustrating its simplicity and efficient performance. The comprehensive depiction of the proposed model is presented in the following subsections:

4.1 Data description

Some data sources share parameters like log timestamps and vehicle flow. Sensors at model sites record data. Highway England harvested loop-based traffic monitoring unit (TMU) data and inferred journey time using automatic number plate recognition (ANPR). ANPR camera recognition measured travel times between two points, while road surface sensor loops measured speeds, vehicle flows, and occupancy. In case of a loop failure on site, the flow value was imputed from previous values, not vehicle category or speed [32]. The following dataset is suitable for testing and validating our proposed network methodology, which is detailed below.

- **MIDAS dataset**

This study relies on an empirical examination using a collection of historical data for ten of the most heavily trafficked motorways in the United Kingdom. Highways England has jurisdiction over the majority of Class A motorways and major roads in England, specifically focusing on UK data [24]. The study period spanned from January 1, 2022, to December 30, 2022, encompassing all highways. The majority of data collection sites are equipped with inductive loops, although a small number of sites are also utilized for experiments involving radar technology. The motorway incident detection and automatic signaling (MIDAS) original gold dataset is recorded at a frequency of one entry per minute. The site had specific rules for logging the gathered data, including publication time, speed (with a threshold of 240 km/h), vehicle flows (with a threshold of 120 veh/min), and reporting occupancy and headway on a per lane basis. The classification of vehicle flows is based on the length of each vehicle and is determined by the traffic monitoring equipment installed on the roadside. The vehicle flows, which were classified into categories, were converted into volumetric measurements of vehicles per minute for each lane. These measurements were then combined to obtain readings for the entire carriageway. Monthly files are generated for each model site [32]. Table 2 displays the significant data fields within the MIDAS traffic flow dataset.

4.2 Feature extraction stage

The initial step involved extracting

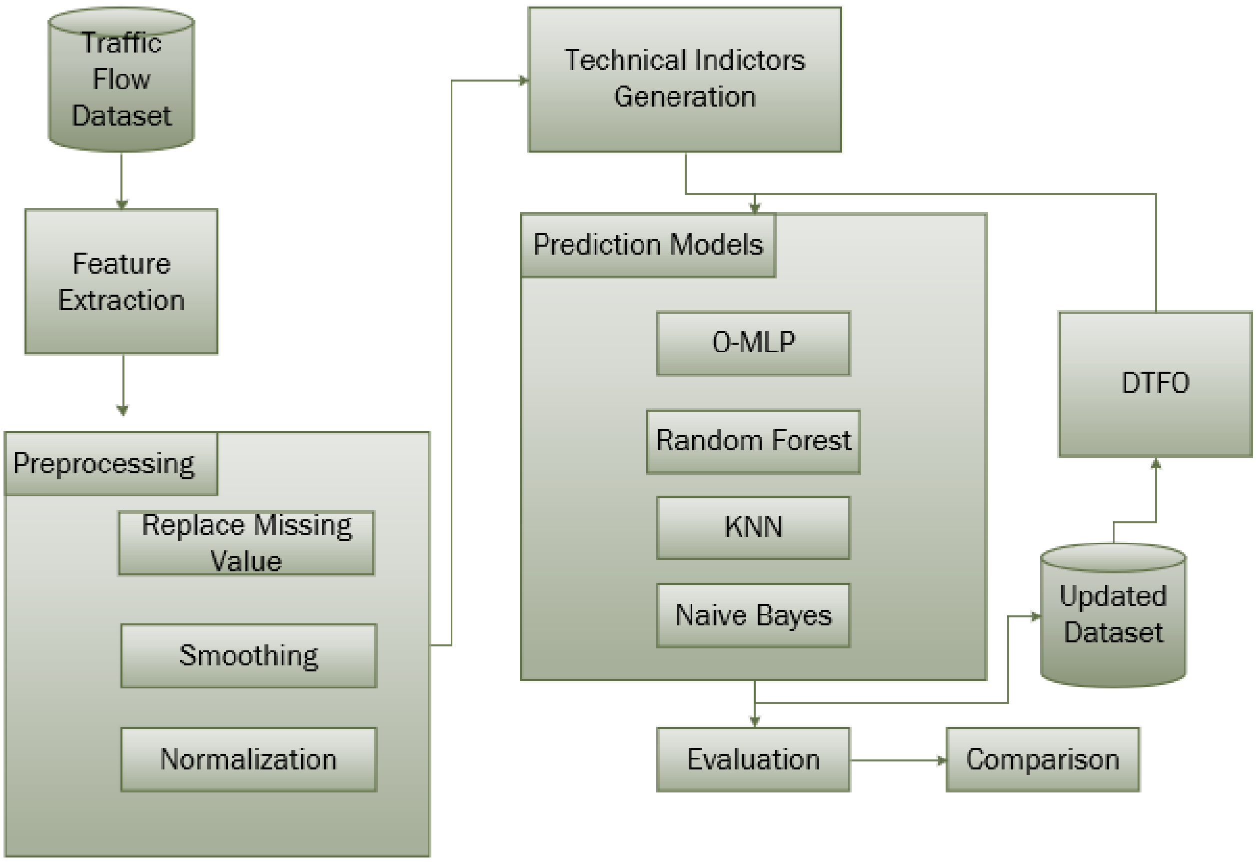

Figure. 1 The proposed system

conventional features from the MIDAS dataset. Table 2 displays the site name, local date, local time, day type, and total carriageway flow.

4.3 Pre-processing stage

During this phase, the data underwent processing and was made ready for the prediction stage. The missing values were imputed using the historical average of flows observed at corresponding times throughout the rest of the year. Moreover, smoothing is employed to decrease the impact of noise and variance in the data sets. As stated in reference [25], it improves the depiction of basic patterns and trends in the data, thereby assisting machine learning algorithms in recognizing and understanding important patterns. The computation of the exponentially smoothing metric for the Y series can be performed iteratively, as

$$t > 0, S_t = \alpha \,.\, Y_t + (1 - \alpha) + S_{t-1} \tag{11}$$

Let α denote the smoothing factor, where α is a value ranging from 0 to 1.

Ultimately, all features were standardized using the Z-score function as described in Eq. (12).

Table 2. The MIDAS Dataset [24] includes unique field names and description features related to traffic flow

Data Field	Description
Site Name	An illustration of the site.
Local Date	Provide the current date adjusted for British Summer Time.
Local Time	The time intervals within the British Summer Time region are denoted in 15-minute increments.
Day Type	The system categorizes the days into three distinct sections: normal workdays, holidays, and weekends.
Total Carriageway Flow	The count of vehicles observed on a specific lane during a 15-minute interval.
Speed Value	The site measured the average velocity, expressed in kilometres per hour, of all vehicles across all lanes during the 15-minute interval.
Network Link Id	A distinct identifier specific to the NTIS link.

$$v' = \frac{v - \mu_A}{\sigma_A} \tag{12}$$

Let A denote the original dataset, μ represent the mean value of the feature, and σ denote the standard deviation of the feature [33].

4.4 Technical indictors generation stage

Technical indicators (TIs) are useful for accurately forecasting short-term vehicle flow and can be integrated into traffic management systems. Hence, the aim of this stage is to enhance the accuracy of forecasting by employing the standard features, which comprise four attributes, as inputs for the models. The outcomes derived from this phase comprise the four TIs characteristics. Algorithm 1 contains supplementary details about the TIs generating stage. The formulation of these indicators relies on Equations (1 - 6), which are computed using historical flow data.

Algorithm 1: The TIs generation stage from dataset

Input: The dataset is represented as an array S of dimensions n×m×r, where n represents the number of data points, m indicates the number of standard technical attributes, and r represents the historical data size.

Output: Array **O1** ($n{\times}j{\times}r$), where j is the number of Technical Indicator features.

Begin

1. **Let** k represent the period sliding window.
2. **Initialize** k ← 4.
3. **Initialize** an empty array O1 of size ($n{\times}j{\times}r$) to store the Technical Indicator features.
4. **For** each data point t in S do
5. **For** i ← 0 to m do
6. **Execute ATR**(Density[t], High[i], Low[i], Density, t, i, k) in parallel and save the output in the **ATR(i).**
7. **Execute EMA** (Density, t, i, r) in parallel and save the output in the **EMA(i).**
8. **Execute RSI** (Density, t, i, r) in parallel and save the output in the **RSI(i).**
9. Execute **ROC** (Density, t, i, r) in parallel and save the output in the **ROC(i).**
10. **End** For
11. **End** For
12. **Merge** array **S** of size (n×m×r) with array **O1** of size (n×j×r) to create Input2 of size (n×|S+O1|×r).

End

4.5 Prediction model

This paper employs three widely-used prediction techniques to compare with the proposed prediction model. Initially, the random forest algorithm was utilized to forecast forthcoming traffic patterns. The squared error is employed as the function to evaluate the quality of the split in each node. Additionally, the K-NN algorithm was utilized to forecast forthcoming traffic patterns. Through experimentation, it was determined that a value of K equal to five yielded the highest level of prediction accuracy. Furthermore, the Naive Bayes algorithm was utilized to forecast forthcoming traffic patterns.

The proposed model was derived from a well-established prediction model. The task was accomplished by employing O-MLP neural networks. These networks were trained using the BP algorithm with one hidden layer. The training process involved minimizing the specified loss function, which was the squared error. Instead of using the standard gradient descent method, the Quasi-newton method based on projected gradients was utilized. The proposed model was derived from an established prediction model. The task was accomplished by employing O-MLP neural networks, which were trained using the BP algorithm with one hidden layer. The training process involved minimizing the specified loss function (squared error) using the Quasi-newton method, which relies on projected gradients instead of the standard gradient descent approach. Two neurons were identified in the hidden layer, along with a bias. The penalty for exceeding the weight limit is 0.02, and the tolerance value is 1.0E-5. Eq. (13) describes the input-output relationship:

$$Y_t = F\left(w_0 + \sum_{a=1}^{h} w_a \, . \, F\left[w_{0,a} + \sum_{b=1}^{m} w_{b,a} . x_t\right]\right) \quad (13)$$

Where $w_{b,a}$ (a=1, 2, …, h; b=1, 2, …, m), w_a (a=0,1, 2, …, h), $w_{0,a}$,and w_0 are the weights of the network, m: input neurons, h: hidden neurons. $w_{b,a}$ represents the weights between input and hidden layer, w_a represents the weights between hidden and output layer, $w_{0,a}$ represents the weight of input bias neuron, w_0 represents the weight of hidden bias neuron, and finally $F()$ represents a non-linear activation function. Furthermore, a tanh function approximation of the logistic function was employed represented in Eq. (14) as the activation function for the hidden layer in order to enhance

Received: December 28, 2023. Revised: January 17, 2024.

Algorithm 2. Optimization method

Input: Initial Parameters (Q_{vk}), Converges Factor (CF)
Output: Final Weight Vector

Begin
Step 1: **Initialization**.
- Ridge = 0.01
- k=1
- Set the initial values of variable W_k
- Compute the Jacobian vector G_k by utilizing W_k.
- Initialize a B_k matrix

Step 2: If the norm of G_k tends to 0, then cease.
Step 3: Determine the search direction in the form of:
- $d_k = -B_k^{-1} G_k$

Step 4: Utilize line search to find an appropriate step size α and thereafter modify the weight vector.
$$W_{k+1} = W_k + \alpha d_k$$
Step 5:
- Determine G_{k+1} by utilizing W_{k+1}:
$$G_{k+1} = G_k + Ridge\ .W_{k+1}$$
- Calculate changes in the weight and gradient vectors:
$$DW_k = W_{k+1} - W_k$$
$$DG_k = G_{k+1} - G_k$$
- Update the B_k matrix using the BFGS update formula:
$$B_{k+1} = B_k + \frac{DG_k\ .\ DG_k^T}{DG_k^T .\ DW_k} - \frac{B_k\ .DW_k\ .DW_k^T .B_k}{DW_k^T .B_k .DW_k}$$
Step 6: k = k + 1 and then return to Step 2.
End.

the execution speed.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{14}$$

A linear function was employed represented in Eq. (15) in the output layer to forecast the flow value.

$$f(x) = x \tag{15}$$

Furthermore, the suggested approximation activation function can be applied to these layers as well. The concept involves approximating the hyperbolic (tanh) activation function by utilizing Taylor's or Maclaurin's formula to determine an approximate value for (e^x). The method of approximation is demonstrated in Eqs. (16-17).

$$e^x = 1 + \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{16}$$

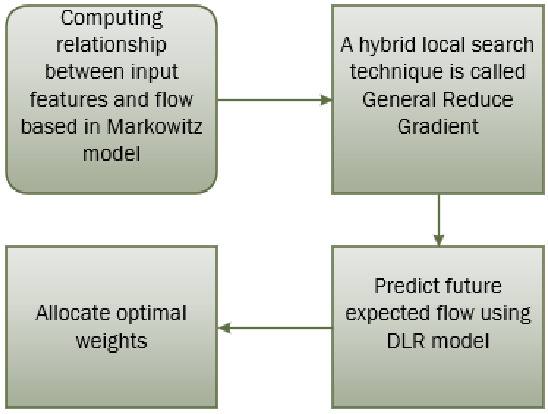

Figure. 2 The proposed methodology for traffic flow optimization

The expression for the approximation function of tanh function is given by:

$$F(net_a) = (z - x)/(z + x) \tag{17}$$

Finally, the proposed model was evaluated and compared to RF, KNN, and Naive bayes techniques using MAE, R2, and RMSE. The optimization model comprises numerous steps, as depicted in Algorithm 2. It aids in error reduction by softening the loss function (the curvature of f), thereby accelerating convergence.

4.6 Deep traffic flow optimization (DTFO) model

The DTFO model is characterized as a hybrid. The diversification principle from the Markowitz model, which calculates the correlation coefficient, local search (exploitation) from the GRG method, and deep learning are integrated into this model. DTFO significantly improves traffic flow optimization, prediction accuracy, and reduces implementation time, as outlined in Fig. 2.

5. Experimental results and discussion

The performance of O-MLP, RF, KNN, and Native Bayes was evaluated by training them on a complete year of data comprising 34,560 data points. Each data point corresponded to a 15-minute interval for each selected road. The relevant parameters of Root Mean Squared Error Table 3 demonstrates that the average percentage of RMSE for the proposed model is 0.14, which is significantly higher than the other models, particularly O-MLP. Table 4 demonstrates that the proposed model has a lower average MAE of 0.307. The proposed model has an average R^2

Table 3. Comparative RMSE for the busiest 10 roads

Road	Junction	RF	Naïve Bays	KNN	O-MLP
M1	J7 – J8	0.30	0.50	0.48	0.16
M60	J13-J12	0.37	0.39	0.41	0.21
M25	J13-J14	0.29	0.62	0.32	0.09
M25	J18-J19	0.27	0.30	0.44	0.11
M1	J6A-J7	0.20	0.49	0.23	0.07
M6	J20-J21	0.29	0.43	0.50	0.09
M8	J16-J15	0.33	0.31	0.34	0.13
M56	J4-J3	0.36	0.54	0.41	0.17
M6	J15-J14	0.42	0.50	0.48	0.18
M25	J23-J24	0.36	0.51	0.45	0.22

Table 4. Comparative MAE for the Busiest 10 roads

Road	Junction	RF	Naïve Bays	KNN	O-MLP
M1	J7 – J8	2.48	10.56	6.11	0.05
M60	J13-J12	4.58	12.82	5.25	0.5
M25	J13-J14	4.19	13.44	2.0	0.15
M25	J18-J19	3.84	12.27	3.41	0.62
M1	J6A-J7	3.81	14.52	3.5	0.25
M6	J20-J21	3.31	12.09	9.44	0.3
M8	J16-J15	3.43	7.84	11.24	0.35
M56	J4-J3	2.72	9.16	12.61	0.22
M6	J15-J14	4.55	11.72	8.46	0.18
M25	J23-J24	6.58	12.52	5.29	0.45

Table 5. Comparative R^2 for the Busiest 10 roads

Road	Junction	RF	Naïve Bays	KNN	O-MLP
M1	J7 – J8	0.821	0.571	0.653	0.971
M60	J13-J12	0.731	0.473	0.788	0.987
M25	J13-J14	0.688	0.842	0.809	0.924
M25	J18-J19	0.672	0.671	0.760	0.984
M1	J6A-J7	0.969	0.797	0.613	0.925
M6	J20-J21	0.826	0.565	0.759	0.992
M8	J16-J15	0.886	0.538	0.803	0.927
M56	J4-J3	0.805	0.588	0.638	0.946
M6	J15-J14	0.737	0.712	0.489	0.919
M25	J23-J24	0.851	0.394	0.746	0.911

value of 0.948, surpassing that of all traditional models.

This suggests that the proposed model exhibited superior efficiency. Despite the high traffic volume on the selected motorways in the UK, the proposed model yields favourable

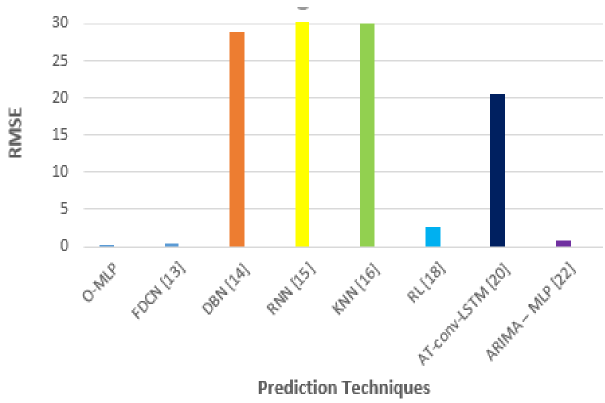

Figure. 3 RMSE of Table 1

outcomes. This validates the model's exceptional capacity to adapt to the significant fluctuations in the quantity of compounds during free flow periods and peak periods.

This demonstrates that the model effectively executed the task of making predictions during this timeframe, encompassing various flow conditions including normal work days, holidays, and weekends, where there is significant variation in vehicle traffic. Upon examining the results, it is apparent that the proposed model demonstrates superior performance when all evaluation measures are applied. were summarized in Table 3. The results of the mean absolute error (MAE) comparison are summarized in Table 4. Table 5 presents a summary of the comparison between the proposed model and traditional models using the R^2 measure.

Fig. 4 illustrates the error analysis of the proposed model in comparison to other models mentioned in Table 1, which utilized identical or similar datasets. Upon examination of the obtained outcomes, the O-MLP model exhibits minimal error when assessed using conventional methodologies. When examining Figure 3, the proposed model demonstrates significant improvements in error values, with percentages of 66.67%, 99.51%, 99.54%, 99.53%, 94.81%, 99.32%, and 82.72% respectively, surpassing traditional FDCN, DBN, RNN, KNN, RL, AT-conv-LSTM, and ARIMA-MLP models. Therefore, the effectiveness of the improved model is confirmed.

6. Conclusions

This paper introduced an optimal model for predicting traffic flow. The model utilizes four input features, namely ATR, ROC, EMA, and RSI, to take advantage of their explanatory capabilities. The OMLP framework utilized these features as inputs to directly forecast the corresponding traffic flow. In order to enhance the accuracy of the prediction, the OMLP was fine-tuned using a novel Quasi-newton (BFGS) algorithm. This optimization technique ensures quicker convergence during the optimization process and improves the training speed and solution quality for MLPs. Subsequently, the correlation coefficient was computed to determine the association between technical indicators and traffic flow, employing the diversification principle from the Markowitz model and the local search (exploitation) principle from the GRG method. Consequently, this method was employed to adjust the weights and optimize the objective function, which is measured by the sharpe ratio. Furthermore, the accuracy of the proposed model is demonstrated by utilizing UK highway sector data as a case study. The proposed model can serve as a valuable tool for transportation agencies and researchers to make informed decisions regarding traffic management. The justification for this approach is based on the utilization of historical traffic flow data to identify any valuable predictive insights. According to the experimental results, the proposed model achieves a RMSE, MAE and R2 (0.14, 0.307, 0.948) as average respectively. Therefore, when compared to the existing methods, the suggested system is superior in terms of efficiency and accuracy of outcomes. As well as, the proposed model's results are optimal, as determined through comparison with other prediction models such as Naive Bayes, KNN, and RF.

Regardless the important contributions and findings of this research, it is not without limitations. First, this study does not delve into external influences (such as weather factors, accidents, etc.) and their impact on traffic flow. Moreover, the proposed model did not take density and velocity into account. Because of these limitations, future work lies in building an integrated traffic flow system by organizing an integrated data set for all relevant variables.

Conflicts of interest

There are no conflicts of interest declared by any of the authors.

Acknowledgment

I acknowledge every one for supporting me for doing research.

Author contributions

Conceptualization, and Methodology, S. T. Hasson; Software, A. I. Turki; Validation, A. I. Turki and S. T. Hasson; investigation, A. I. Turki; resources, A. I. Turki; data curation A. I. Turki; Writing-original draft preparation, A. I. Turki; Writing-review and editing, S. T. Hasson; supervision, S. T. Hasson; project administration, S. T. Hasson.

Hyperparameters tuning

The table below displays the optimal model hyperparameters selected through grid-search. It should be noted that scikit-learn [34] is utilized for executing linear and ensemble models, while Keras [35] is employed for running deep models. Unless stated otherwise in the table below, the default model hyperparameters of either scikit or Keras are used.

Model	Hyperparameters
O-MLP	neuron1: 6, neuron2: 3, loss:MSE, learning rate: 0.001, metrics: ['mae', 'mse', 'mape'], Hidden layer activation function: tanh, Output activation function: linear, EPOCHS : 500, batch_size = 32

Abbreviations

The manuscript employs the following abbreviations:

ARIMA-autoregressive integrated moving average
BFGS – Broyden fletcher goldfarb shanno
BP - Back-propagation
CNN - Convolutional neural network
K-NN - K-nearest neighbours
MLP-NN - Multilayer perceptron neural network
OMLP - Optimize multilayer perceptron
PSO - Particle swarm optimization
RF – Random forest
RNN - Recurrent neural network
TI - Technical indicators
WNN - Wavelet neural network

References

[1] A. I. Turki and S. T. Hasson, "A Markova-Chain Approach to Model Vehicles Traffic Behavior", In: *Proc. of the 2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, Denpasar, Bali, Indonesia, pp. 117-122, 2022.

[2] A. J. Siddiqui, A. Mammeri, and A. Boukerche, "Real-Time Vehicle Make and Model Recognition Based on a Bag of SURF Features", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 11, pp. 3205-3219, 2016.

[3] M. B. Younes and A. Boukerche, "Efficient traffic congestion detection protocol for next-generation VANETs", In: *Proc. of IEEE International Conference on Communications (ICC)*, Budapest, Hungary, pp. 3764-3768, 2013.

[4] C. Rezende, H. S. Ramos, R. W. Pazzi, A. Boukerche, A. C. Frery, and A. A. F. Loureiro, "VIRTUS: A resilient location-aware video unicast scheme for vehicular networks", In: *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, pp. 698-702, 2012.

[5] C. Rezende, A. Boukerche, H. S. Ramos, and A. A. F. Loureiro, "A Reactive and Scalable Unicast Solution for Video Streaming over VANETs", *IEEE Transactions on Computers*, Vol. 64, No. 3, pp. 614-626, March 2015.

[6] S. Deng, S. Jia, and J. Chen, "Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data", *Applied Soft Computing*, Vol. 78, pp. 712-721, 2019.

[7] U. Ryu, J. Wang, T. Kim, S. Kwak, and U. Juhyok, "Construction of traffic state vector using mutual information for short-term traffic flow prediction", *Transportation Research Part C: Emerging Technologies*, Vol. 96, pp. 55-71, 2018.

[8] A. Emami, M. Sarvi, and S. A. Bagloee, "Short-term traffic flow prediction based on faded memory Kalman Filter fusing data from connected vehicles and Bluetooth sensors", *Simulation Modelling Practice and Theory*, Vol. 102, p. 102025, 2020.

[9] F. Kong, J. Li, B. Jiang, and H. Song, "Short-term traffic flow prediction in smart multimedia system for Internet of Vehicles based on deep belief network", *Future Generation Computer Systems*, Vol. 93, pp. 460-472, 2019.

[10] F. Li, S. S. Liao, and M. Cai, "A new probability statistical model for traffic noise prediction on free flow roads and control flow roads", *Transportation Research Part D: Transport and Environment*, Vol. 49, pp. 313-322, 2016.

[11] H. J. Yang and X. Hu, "Wavelet neural network with improved genetic algorithm for traffic flow time series prediction", *Optik*, Vol. 127, No. 19, pp. 8103-8110, 2016.

[12] W. Li, J. Wang, R. Fan, Y. Zhang, Q. Guo, C. Siddique, and X. J. Ban, "Short-term traffic state prediction from latent structures: Accuracy vs. efficiency", *Transportation Research Part C: Emerging Technologies*, Vol. 111, pp. 72-90, 2020.

[13] W. Chen, J. An, R. Li, L. Fu, G. Xie, M. Z. A. Bhuiyan, and K. Li, "A Novel Fuzzy Deep-Learning Approach to Traffic Flow Prediction with Uncertain Spatial–Temporal Data Features", *International Journal of Intelligent Engineering and Systems*, Vol. 1, No. 1, pp. 78-88, 2018.

[14] L. Zhao, Y. Zhou, H. Lu, and H. Fujita, "Parallel computing method of deep belief networks and its application to traffic flow prediction", *Knowledge-Based Systems*, Vol. 163, pp. 972-987, 2019.

[15] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic Flow Prediction Using LSTM with Feature Enhancement", *Neurocomputing*, Vol. 332, pp. 320-327, 2019.

[16] Z. Liu, X. Qin, W. Huang, X. Zhu, Y. Wei, J. Cao, and J. Guo, "Effect of Time Intervals on K-Nearest Neighbors Model for Short-term Traffic Flow Prediction", *Promet - Traffic and Transportation*, Vol. 31, No. 2, pp. 129-139, 2019.

[17] L. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An Effective Spatial-Temporal Attention-Based Neural Network for Traffic Flow Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 1, No. 1, pp. 123-456, 2019.

[18] X. Chen and P. Chaudhari, "MIDAS: Multi-agent Interaction-aware Decision-making with Adaptive Strategies for Urban Autonomous Navigation", In: *Proc. of 2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, pp. 7980-7986, 2021.

[19] X. Chen, S. Wu, C. Shi, Y. Huang, Y. Yang, R. Ke, and J. Zhao, "Sensing Data Supported Traffic Flow Prediction via Denoising Schemes and ANN: A Comparison", *IEEE Sensors Journal*, Vol. 20, No. 23, pp. 14317-14328, 2020.

[20] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A Hybrid Deep Learning Model with Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 11, pp. 6910-6920, 2021, doi: 10.1109/TITS.2020.2997352.

[21] F. I. Rahman, "Short Term Traffic Flow Prediction Using Machine Learning - KNN, SVM, and ANN with Weather Information", *International Journal for Traffic & Transport Engineering*, Vol. 10, No. 3, pp. 123-456, 2020.

[22] V. Rajalakshmi and S. G. Vaidyanathan, "Hybrid Time-Series Forecasting Models for Traffic Flow Prediction", *Promet - Traffic and Transportation*, Vol. 34, No. 4, pp. 537-549, 2022.

[23] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets", In: *Proc. of International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, pp. 4368-4374, 2016.

[24] A. I. Turki and S. T. Hasson, "Study estimating hourly traffic flow using Artificial Neural Network: A M25 motorway case", *Samarra Journal of Pure and Applied Science*, Vol. 5, No. 1, pp. 47-59, 2023.

[25] A. I. Turki and S. T. Hasson, "Using a New Algorithm in Machine Learning Approaches to Estimate Level-of-Service in Hourly Traffic Flow Data in Vehicular Ad Hoc Networks", *Tikrit Journal of Pure Science*, Vol. 28, No. 3, pp. 74-83, 2023.

[26] S. Kumar S and B. K. Raghavendra, "An Efficient Approach for Coffee Leaf Disease Classification and Severity Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 5, pp. 123-456, 2023, doi: 10.22266/ijies2023.1031.59.

[27] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal Traffic Flow Prediction with KNN and LSTM", *Journal of Advanced Transportation*, Vol. 2019, Article ID. 4145353, pp. 1-10, 2019.

[28] S. P. Chelluboina and K. N. Rao, "CATDSNet: Computer Aided Tongue Diagnosis System for Disease Prediction Using Hybrid Extreme Learning Machine", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 1, 2023, doi: 10.22266/ijies2023.0228.01.

[29] J. E. Dennis, Jr, and J. J. Moré, "Quasi-Newton methods, motivation and theory", *SIAM Review*, Vol. 19, No. 1, pp. 46-89, 1977.

[30] K. L. Du and M. N. S. Swamy, "Neural Networks and Statistical Learning", *Springer Science & Business Media*, 2013.

[31] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Real-Time Indoor Air Quality Index Prediction Using a Vacuum Cleaner Robot's AIoT Electronic Nose", *International Journal of Intelligent Engineering & Systems*, Vol. 16, No. 5, pp. 263-274, 2023, doi: 10.22266/ijies2023.1031.23.

[32] https://www.trafficengland.com/resources/cms-docs/user-guide.pdf

[33] P. Muthulakshmi and M. Parveen, "Z-Score Normalized Feature Selection and Iterative African Buffalo Optimization for Effective Heart Disease Prediction", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 1, pp. 25-37, 2023, doi: 10.22266/ijies2023.0228.03.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and É. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, 2011.

[35] F. Chollet, *Keras*, Available online: https://keras.io, Accessed on: 29 Jun 2023.