ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TRAFFIC SPEED PREDICTION AND CONGESTION LEVEL IDENTIFICATION USING FLOATING CAR DATA

FATEMEH AHANIN, NORWATI MUSTAPHA, MASLINA ZOLKEPLI, NOR AZURA HUSIN

Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Jalan Universiti 1 Serdang, 43400 Seri Kembangan, Selangor, Malaysia

E-mail: fatemeh.ahanin@gmail.com

ABSTRACT

Traffic congestion poses significant challenges to urban mobility and transportation infrastructure worldwide. Accurate prediction of traffic speed and timely identification of congestion levels are crucial for effective traffic management and planning. Owing to the widespread adoption of telecommunication technologies, various traffic datasets have become available, such as Floating Car Data (FCD), which collect real-time information from vehicles in transit, providing a rich and dynamic dataset for analyzing traffic speed. However, predicting traffic speed and identifying congestion levels using FCD remains challenging due to the complexities of traffic dynamics and the non-linear nature of traffic flow. In response, multiple solutions have been proposed using deep learning methods. This study addresses the persistent issue of FCD data sparsity and its limitations in providing consistent, accurate traffic speed predictions. The present work focuses on constructing an LSTM-based method, called LSTM-C, to predict traffic speed. In the proposed LSTM-C method, a new Contrast measure is introduced and incorporated to enhance the prediction of traffic speed across candidate road segments. The LSTM-C model demonstrates a significant improvement in both prediction accuracy and congestion level identification, outperforming existing models such as those by Majumdar et al. and Gao et al. Subsequently, traffic rules are applied to the predicted speeds to determine congestion levels for each segment. The experimental results demonstrate that the proposed model achieves a high level of accuracy, reaching up to 96.697%, which represents an improvement of 1.6% and 1.79% in accuracy compared to the two benchmark LSTM methods employed for speed prediction.

Keywords: *Traffic Speed Prediction, Short-Term Speed Prediction, The Long Short-Term Memory (LSTM), Deep Learning, Data-Driven Traffic Analysis*

1. INTRODUCTION

Today, traffic congestion, as a modern phenomenon, has globally become a serious problem in many urban areas, influencing people's lives economically, culturally, etc. Among problems caused by traffic congestion are people's mobility, timing, life quality as well as traffic planning systems and management. Such an issue is even becoming more critical owing to the growing numbers of vehicles, causing air pollution to be a serious problem in various corners of the world. Different organizations and institutions including governments and universities together with Research and Development (R&D) sectors have dealt with this issue, striving to alleviate the congestion problem through using technologies for monitoring and managing traffic [1]. Machine Learning methods play a key role in Data-driven traffic data analysis. It ought to be pointed out that all machine learning methods demand data for the purpose of training and testing the prediction models [2-6]. Affiliated organizations need to gather an excess of traffic data from different sources for analysis [7,8]. Various technologies such as video cameras, inductive loop detectors and other static sensors may be fitted in at certain fixed places on roads for detecting traffic state (e.g., flow velocity and traffic density) [9]. Whereas such devices provide sufficient and reliable traffic data to be used for managing traffic, these methods fail to cover all roads due to entailing a great amount of infrastructure deployment apart from high maintenance costs. Yet, it is suggested to employ Floating Car Data (FCD), which is a convenient and cost-effective method for collecting traffic data. There is no need to use any specific device since FCD offers good coverage across road networks. However, the limitation of FCD is data sparsity. Despite growing attention to FCD-based traffic analysis, many current models still underperform

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

due to their inability to fully capture abrupt or subtle shifts in traffic speed caused by dynamic, nonlinear flow behavior. This gap necessitates new solutions that enhance pattern recognition without relying on costly hardware-based data sources. Motivated by the significant economic and environmental challenges posed by traffic congestion in urban areas, and the limitations of FCD, this research intends to enhance traffic speed prediction using an improved LSTM-based algorithm (LSTM-C) to better manage congestion using FCD.

This article aims to predict traffic states (in particular traffic speed as it has direct relationship with traffic congestion levels) of road segments using (FCD) of taxis in Beijing, China. Time window of 15-min is applied for the prediction horizon. It should be noted that the traffic speed will be predicted through using an improved LSTM (Long Short-Term Memory) algorithm. In the proposed method, Contrast measure formula is improved by the author and then it is incorporated in the LSTM (called LSTM-C) to enhance the traffic speed prediction. Afterwards, the predicted traffic speed will be employed for identifying traffic congestion level. The accuracy of the proposed method can be compared with the work of Majumdar et al., [10] and Gao et al. [5] as our benchmarks. These (benchmarks) models are re-run employing the same FCD and 15-min time window, given the accuracy of 94.906% and 95.082% respectively. The proposed method accuracy is up to 96.697%, proving LSTM strengths and potential in the traffic prediction field. The prediction of the average speed of the vehicles passing corresponding road segments could be of help for identifying traffic congestion. We hypothesize that incorporating an improved contrast measure into LSTM will enhance short-term traffic speed prediction accuracy using FCD data, and the **research question is whether** this integration significantly improves prediction accuracy compared to standard LSTM architectures.

The contributions of this research are listed as follows:

1. Employing the deep learning model on traffic state prediction.

2. Introducing a new Contrast measure for capturing traffic speed changes more effectively (Contrast measure formula is improved by the author)

3. Construction of a new speed prediction model called LSTM-C that concatenates Contrast measure in LSTM.

4. Validating the proposed LSTM-C model

5. Comparing the model with a latent LSTM model Majumdar et al. [10] and Gao et al., [5].

6. Identifying the traffic congestion level on the basis of the predicted speed.

The remainder (of the paper) is organized as follows: Section 2 reviews the related work on traffic seed prediction. Section 3 describes the model design. In Section 4, model implementation is discussed. This is followed by experimentation results in Section 5. Finally, Section 6 concludes the paper, suggesting the future research direction.

2. RELATED WORK

The leading idea in the present article is to propose an improved LSTM model called LSTM-C for predicting the average traffic speed. This will then be followed by identifying the level of congestion on the basis of the traffic congestion standard in Beijing. In accordance with research available, three main categories are considered to model traffic state prediction employing Artificial Intelligence (AI) with Probabilistic Reasoning, Shallow Machine Learning, and Deep Learning (DL) as shown in Figure 1 [11]. As the proposed method is built upon deep learning (DL), the existing literature predominantly emphasizes DL-related approaches.

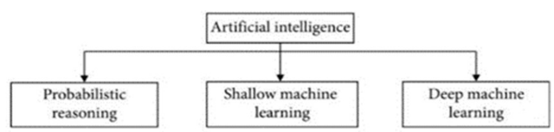

Figure 1: Classification of Artificial Intelligence techniques for traffic state prediction [11].

Probabilistic reasoning are approaches that involve the use of probability and logic to deal with uncertainty. Several relevant algorithms are used for predicting traffic state.

The researchers have employed Fuzzy logic methods to predict traffic state in a number of studies [12-14]. Hidden Markov Model (HMM) is said to be the next algorithm category in probabilistic reasoning. As a model, Markov chain recognizes probabilities of sequences in state variables which is commonly used for modelling time-series data. HMM, applied in various studies, distinguishes traffic patterns in congestion prediction [13, 15]. A hybrid model composed of HMM and contrast measure was suggested to foresee traffic states in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

roads. Whereas contrast measure may be used as a useful statistical technique for capturing traffic state variation, HMM's generalization is believed not to be sufficiently strong. Gaussian processes (GPs), as another group of probabilistic reasoning models, are said to be flexible non-parametric models [16]. Meanwhile, GPs may be employed to model complex time-series. Three data sources including trajectory data and speed data as well as traffic-related tweets were used in a study by [17] for predicting road traffic speed. Incorporating three sources of data seems to be interesting, whereas the model structure is relatively complex. The probability of traffic state distribution was presented in another study [18], applying EM algorithms for choosing variance parameters and mean of Gaussian distribution. Eventually, it is time for Bayesian network (BN), as another category of Probabilistic reasoning algorithms, directed graph models and capable of presenting conditional independencies among random variables. It should be pointed out that Graph Theory and Probability Theory are combined for building BN to direct key issues in engineering and applied mathematics [11,19]. In a study [20], BN is used to predict and detect traffic congestion, where three sources of data (i.e., Loop detector, incident data and weather information) were applied in the proposed method. 40 scenarios were presented on the basis of congestion occurrence probabilities.

In Shallow Machine Learning (SLM) category of algorithm, which is composed of Traditional and basic ML algorithms, features cannot be extracted from the input, and therefore need to be defined beforehand. Once the feature extracted, model training may be used. ANNs, being a type of model for machine learning (ML), are relatively competitive to conventional regression and statistical models in terms of utility [21]. Owing to its ability for efficient forecasting and easy implementation, ANN is employed as one of the most frequently used algorithms in traffic state prediction. In various researchers, ANN-based algorithms such as Feedforward neural network (FNN), and Backpropagation neural network (BPNN) in traffic management were effectively applied. A method was suggested in [22] using BPNN for foreseeing traffic flow and obtaining the congestion grade judgment. The data generated based on SUMO traffic simulation data as well as the proposed congestion evaluation algorithm based on road occupancy (CRO) were compared considering three other evaluation methods including congestion evaluation based on mileage ratio of congestion (CMRC) along with road speed (CRS) and vehicle density (CVD). The results obtained divulged that the congestion degree of roadways was correctly expresses bearing less training cost with low processing time of real-time processing. In another study [23], a hybrid NN was proposed via bringing together an adaptive prediction algorithm (Adaptive RMSE) with BPNN. The data gathered in the present paper is based on real-time GPS data, updating the database, yet the data increment effect in the accuracy was not elaborated. One of the merits of applying ANN algorithms is that the data analysis is flexible and capable of handling multidimensional problems effectively. Each layer in the artificial neurons may be modified on the basis of input data. Enjoying flexible structures for capturing complex nonlinear behaviors, ANNS are believed to be effective in recognizing and modeling patterns for diversity of road types. However, for ANN to be increased in performance, large datasets are required to add complexity. Regression, as a statistical method used in data science and ML for different tasks including prediction, forecasting, and time series modeling, is employed to model the relationship between input and output numerical variables. Numerous types of regression including linear regression and logistic regression are at work. For predicting the traffic flow, such multiple machine learning algorithms as Linear Regression, Gradient Boosting as well as Random Forest, and two DL models were applied [24]. Having carried out on public dataset derived from induction loops, the algorithms obtained similar results, where Linear Regression had the lowest performance accuracy yet requiring less training time compared to other models. In spite of producing good results, the traffic flow prediction accuracy could possibly increase when using additional features. [25] suggested using a framework for foreseeing traffic congestion through exploring correlation among roadway congestion and energy usage. In a study of traffic speed prediction [26], the researchers integrated the spatiotemporal correlation, comparing linear regression model and LSTM model. The results revealed that the LSTM model outdid other models. Regression techniques are believed to have yielded favorable results in predicting time series problems including traffic forecasting and management. Linear time series are effective, yet normally failing to elaborate certain data aspects. Hence, such models do not seem to be reliable for nonlinear datasets. As a supervised learning method, decision tree is applied for prediction and classification, using a set of if-then-else conditions for learning from data as well as employing all features present in data for a series of decision-making. Multiple decision tree (known as Random Forest) proves to occasionally produce effective results. Random Forest provides decisions trees, comprising the results for making the final decision on the basis of the majority. In the study done by [27], a Convolutional Neural Network–Gated Recurrent Unit (CNN-GRU) was introduced to foresee traffic speeds in the three lanes as to the same road segment. Compared to Linear Regression model, this method is said to have yielded better results. A random-forest-based model in a study by [28] was suggested, in which a combination of Classification and Regression Trees, called CART, was chosen for traffic congestion prediction. The researchers utilized a number of variables including weather conditions, time, road quality and holiday in the model. The data is classified by decision tree via learning simple decision rules considering one or more input data. Nevertheless, decision tree normally provides binary results, being not suitable for foreseeing traffic congestion.

DML is in fact the deep ANN, showing several hidden layers on NN. When it comes to discussing SML, there is only one hidden layer, yet with several hidden layers in DML. Such hidden layers, being units of nonlinear process units, are applied for extracting features and transforming data. Note that in DML, diverse traffic data may be transformed into feature vectors or patterns in a certain time limit [11]. As a result, DML enjoying great strengths is the dominant method in TSP with limited collection time horizon into patterns or feature vectors. DML has just recently become popular in studies involving predicting traffic congestion. This section will discuss DML algorithms applied in TSP. By way of example, Convolutional neural network (CNN) can be said to be one of the common DML algorithms in TSP, having wide usages in processing images and computer vision. CNN could process traffic data effectively when taking traffic data as a time-space (2D) matrix. The matrix with time-dimension features is connected with traffic information of the roads. In a timestamp among all roads, the matrix with space-dimension features is linked with the traffic flow information [29]. Therefore, the traffic state variables as to roads may at certain timesteps be foreseen by CNN. Following extracting the time-space features in input data, one can predict traffic state via a full connection layer. The study done by [30] divulged that compared to the first matrix dimension representing the temporal feature, the second shows average traffic speed. In fact, the false-positive rate as well as detection rate was adopted for different dataset partitions. Nonetheless, we can gauge traffic speed through only 3 levels in traffic congestion. They include heavy traffic (0-20), moderate traffic (20-40), and free flow traffic (>40), respectively. Multiple convolutional operations are used by [31] for designing multiscale traffic patterns and temporal dependencies alike by means of video surveillance data. In the paper, the definition of congestion level is provided as the average travel time for that segment at each timeslot, to be compared with the prediction of PCNN congestion level. Recurrent Neural Networks (RNNs), as another significant algorithm in DML, may help to learn Spatiotemporal features. As a result, it is suggested to employ RNN as a DML prediction model, where the output of the previous timestep functions as the input in the next timestep. Backpropagation through time is the basis of how RNN learns, having an input layer and hidden layer as well as output layer. The RNN with two variants is known as LSTM and GRU. In study by [32], a deep stacked bidirectional and unidirectional LSTM named SBU-LSTM was suggested for predicting the traffic speed by means of fixed-position sensors. In spite of providing good accuracy performance for speed prediction, SBU-LSTM fails to identify the traffic congestion levels. [33] presented a Res-RGNN (Residual Recurrent Graph Neural Networks) for predicting traffic speed including loop detector data. Res-RGNN, as a hybrid algorithm of GRU and Graph convolution, models the direct relationships between historical and future time-steps utilizing gating mechanism. Despite Res-RGNN obtained good performance results, it is still felt necessary to investigate the spatiotemporal features learned by MRes-RGNN for finer interpretability. In another study, a Speed Prediction of Traffic Model Network (SPTMN) was proposed based on both Graph Convolution Network (GCN) and Temporal Convolution Network (TCN), where loop detector data was applied to test the model [34]. In another study conducted by [29], traffic speed was projected employing such various algorithms as CNN, LSTM and GRU, where the predicted speed was exercised to recognize the level of congestion. They used HMM-based map-matching to approximate the average traffic speed as a pre-processing step prior to forecast. It ought to be pointed out that the LSTM model accomplished the highest performance in which the best results for window length of 8, MAE, RMSE and MSE include 1.45, 6.08 and 36.97, respectively. Research carried out by [35] put forward a traffic congestion model applying attention-based LSTM via fixed-position sensors, revealing how a specific traffic state is of significance to the entire traffic flow and implying further contextual association. Nonetheless, this model fails to have desired results due to emphasizing on traffic flow rather than speed prediction, as an effective method for predicting congestion. Gao et al. recommended a traffic forecasting model that extracts traffic flow pattern from five loop detectors, which is followed by employing LSTM for prediction [5]. The method proposed is formulated on relationship between flow and speed in traffic flow theory, which is not in the scope of the present study. Another study by [10] described a model for traffic speed forecast, generating the congestion level through LSTM, utilizing loop detector data for experimentation as well as applying RMSE to measure performance, reaching 84–95%. The serous disadvantage is the absence of numerical experiment to identify congestion level; therefore, it just sufficed to present visual congestion propagation. A study by [6] proposed a deep learning model based on LSTM networks for short-term traffic speed forecasting using GPS-monitored data. Although the authors mention tuning hyperparameters, details about the optimization process are limited. Moreover, the exact short-term prediction horizon used for the implementation is not specified. D-LSTM (Long Short-Term Memory with Dynamic Time Warping) model was introduced for short-term road traffic speed prediction [36] using GPS positioning data. While the results achieved good performance details about hyperparameter tuning and sensitivity analysis are not explicitly discussed. A study by [3] presented an attentive graph neural process AGNP method for network-level short-term traffic speed prediction and imputation. The approach combined the strengths of Gaussian processes GPs and graph neural networks GNNs. The performance results of LSTM in comparison to proposed AGNP are 4.084 and 4.013 respectively. The accuracy of the proposed method requires further improvement.

[2] addressed traffic speed forecasting (TSF) using GPS probe data from registered transport vehicles on parallel multilane roads using enhanced LSTM algorithms (PSO-LSTM and GA-LSTM). The accuracy of speed forecasting using MAE reached 6.55, which implies the method requires further improvements.

Despite advances in AI-based traffic speed prediction, many existing models do not adequately leverage simple, interpretable statistical features such as speed variation patterns. Furthermore, most works rely on structured sensor data (e.g., loop detectors), while Floating Car Data (FCD) remains underutilized due to sparsity issues. This study aims to address these limitations by introducing a contrast-based feature to improve learning from FCD. There is limited research on the integration of contrast measures within deep learning models such as LSTM or GRU for traffic speed forecasting. No known models have tested the statistical contrast feature in neural network-based traffic applications, particularly using FCD.

The proposed model in this paper aims to provide an accurate LSTM based model using Contrast measure and speed. Contrast measures have never been utilized in Neural Network and Deep learning models. Moreover, the author has improved the Contrast measure formula in the present study.

3. MATERIALS & METHODS

3.1 Model Design

The present research aims to incorporate Contrast measure into LSTM model to increase the speed prediction accuracy, resulting in the LSTM-C model, which is explained in section 3.3. Subsequently, the level of congestion will be identified} using the predicted speeds.

This study employs a comparative experimental design, building upon methodologies from earlier traffic prediction studies (e.g., Majumdar et al., 2021; Gao et al., 2022), and enhances them with a novel feature (Contrast measure) evaluated on the same dataset to ensure consistent benchmarking. The proposed LSTM-C model is compared with the work of Majumdar et al. (2021) and Gao et al., (2022). Both models are tested using the same FCD dataset, and their results are presented. Meanwhile, in the following sections, it is tried to elaborate the basics of the LSTM model, Contrast measure, and the suggested LCTM-C model.

3.1.1 The basic of LSTM model

The LSTM model falls under the RNN model, in which the output of the previous timestep is considered as the input in the next. How RNN learns is based on backward propagation through time. RNN can comprise three layers including input layer, hidden layer, and output layer. The major refinement in LSTM over RNN is to add gate structure in the cells of hidden layer as shown in Figure 2. In fact, each cell in the hidden layer enjoys three gates viz input gate, forgetting gate, and output gate are incorporated. The input of the current hidden layer is provided with the output produced in the previous hidden layer and the recorded cellular information as well as the input at the current timestamp [12] (see Figure 2). The input information will be logged selectively into the cell state through the input gate. What is required in the cell is forgetting certain state information by the forget gate. The output gate plays a part in selecting the output result in the hidden layer.

The LSTM layer provided in our model is made up of 128 LSTM cells, applied to the model by [10] and [5] as well.

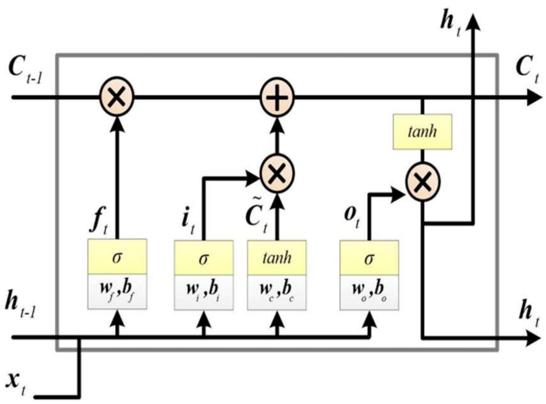

Figure 2: The structure of the hidden layer [29]

How to calculate the hidden layers is illustrated by the equations below, where W is the weight matrix, and b is the constant.

The forget gate is calculated as ft using Eq. 1, where a nonlinear activation function σ, the output of the hidden layer at t – 1 as o_{t-1}, and the input at time t as x_t are used.

$$f_t = \sigma(W_f.[o_{t-1}, x_t] + b_f) \quad (1)$$

The input gate i_t is represented in Eq. 2.

$$i_t = \sigma(W_i.[o_{t-1}, x_t] + b_i) \quad (2)$$

Ĉ is the candidate cell information at time t, described by Eq. 3 and the tanh function is applied as activation function.

$$\hat{C}_t = tanh(W_c.[o_{t-1}, x_t] + b_c) \quad (3)$$

The cell information is described by Eq. 4 as c_t at time t.

$$c_t = f_t.C_{t-1} + i_t.\hat{C}_t \quad (4)$$

The output gate is defined by Eq. 5 as h_t.

$$h_t = \sigma(W_o.[o_{t-1}, x_t] + b_o) \quad (5)$$

The final output is defined by Eq. 6 as o_t.

$$o_t = h_t.tanh(C_t) \quad (6)$$

3.1.2 Definition of contrast measure

The term Contrast can be seen in image processing field to measure the intensity between two consecutive pixels [37]. Calculating the Contrast described by Eq. 8 [38] requires applying the Gray Level Co-occurrence Matrix (GLCM), which contains information as to how frequently the two data points with gray level values c1, c2, and the distance d might take place. The distance in an image could characterize a pixel and its consecutive pixel if d =1 or two pixels apart if d=2 [15]. This will be followed by using the GLCM amount in Eq. 7, calculating Contrast.

$$CON = \sum_{c1,c2} (c_2 - c_1)^2 a_{c1c2} \quad (7)$$

Where:

$$a_{c1c2} = \frac{\#\ of\ pairs\ at\ distance\ d\ with\ gray\ level(c_1, c_2)}{total\ number\ of\ possible\ pairs} \quad (8)$$

Contrast measure formula is improved in this paper by removing power 2 from the Eq. 7. Eq. 9 is the improved Contrast formula which can capture increases and decreases in traffic speed data by employing positive and negative signs of GLCM.

$$CON = \sum_{c1,c2} (c_2 - c_1) a_{c1c2} \quad (9)$$

In the traffic context, c1 and c2 suggest two different speed vectors, of which the total occurrence in each speed pair is determined by ac1c2 in Eq. 8.

Let us provide an example to better understand GLCM matrix. Figure 3 represents a sequence of 9 traffic speeds S = [8 8 9 10 8 8 9 10 10], of which the total occurrence of the pair (8,8) in the GLCM matrix is a8,8 = (0.25).

$$a_{8,8} = \frac{number\ of\ pairs\ at\ distance\ d\ with\ gray\ level(c_1, c_2)}{total\ number\ of\ possible\ pairs} = \frac{2}{8} = 0.25$$

$$a_{8,8} = \frac{2}{8} = 0.25$$

The total Contrast for the sequence reads CON = +0.75.

Journal of Theoretical and Applied Information Technology
15th August 2025. Vol.103. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

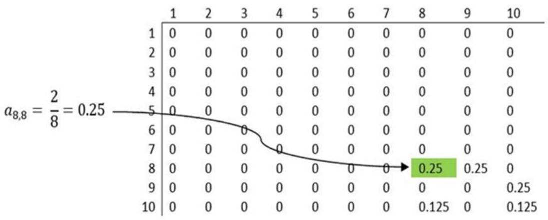

Figure 3. GLCM matrix for a speed sequence [37]

3.1.3 Proposed LSTM-C model

The original idea of the LSTM-C is to incorporate the contrast as a statistical measure in a deep learning model to enhance accurate prediction. In the LSTM-C model presented, contrast is connected to the output in feedforward neural network for foreseeing the speed. The LSTM-C structure is shown in Figure 4.

As can be seen from Figure 4, the concatenation layer is added to the LSTM-C network to predict the traffic speed. Algorithm 1 shows the main steps of the proposed model.

Due to structural similarity between LSTM and GRU, the author has concatenated the Contrast measure in GRU as well in the implementation section. However, the GRU algorithm formulas are not discussed in this paper as it would make this section too long.

Algorithm 1: Traffic Speed Prediction Using LSTM-C
Input: Raw FCD
Output: Speed Prediction
1. Data Pre-processing
Data cleaning, Filter Arterial Road, Parameter Determination
Calculate average speed for every 15-min timeslot
2. Feature Extraction
Create GLCM matrix using Eq (8)
Develop Contrast feature Eq (9)
3. Set the parameter for the network
Set the value of the hidden unit
batch size
Max epoch value
Dropout value
L2 Regularisation value to improve the LSTM network
4. Concatenate Contrast to LSTM
5 Obtain the network output
6 Train the model
7 Apply the fine-tuning strategy
8 Test the model
9 Obtain the prediction
10 Output the prediction report

Figure 4: LSTM-C Network Model

Following prediction steps, the relevant error is reported by means of MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). Eventually, the speeds predicted are utilized for identifying the congestion levels in the corresponding segments as per the traffic index and speed ranges in Table I. If each predicted speed falls under the same category of the actual speed, then the identification of the congestion level based on the predicted speed will be very precise. The overall model workflow is shown in Figure 5.

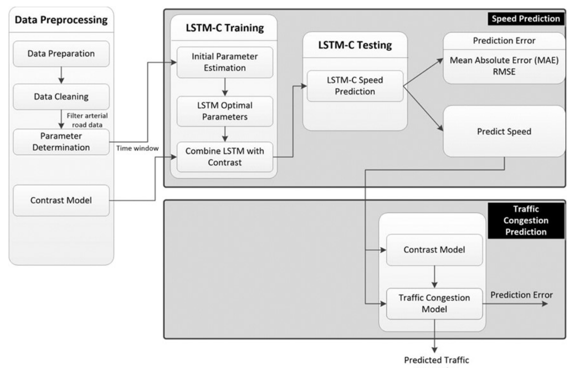

Figure 5. The overall model workflow

4. MODEL IMPLEMENTATION

In this section, the findings are presented, and the performances of the proposed model will be discussed.

4.1 Test Environment

The proposed model is tested using Python programming language in Google Colab which provides a single 12GB NVIDIA Tesla T4 GPU.

4.2 Pre-processing

To predict the average traffic speed, the proposed model is applied. This is followed by considering the traffic conditions in arterial roads in Beijing for 24 hours and dividing the time into 96 time slots at 15-min intervals. The average speed is computed for each timeslot.

The speed limit on the Beijing arterials being 120 km/h, the speed range is $0<v\leqslant 120$ km/h yielding 120 different observations. This leads to computational complexity. Consequently, the speed was split into equally spaced ranges of 5 km/h to bring down the observations and computational

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

complexity. As shown in Table 1, observations reduction into 24 different observations has been provided. As an example, any speed between 25 and 30 km/h is shown by traffic index 6. This step is performed on our LSTM-C model and the model provided by [10] and [5] as well.

Table 1: Traffic Index and Speed Ranges

Traffic Index	Speed Range (km/h)	Traffic Index	Speed Range (km/h)
1	0<v⩽5	13	60<v⩽65
2	5<v⩽10	14	65<v⩽70
3	10<v⩽15	15	70<v⩽75
4	15<v⩽20	16	75<v⩽80
5	20<v⩽25	17	80<v⩽85
6	25<v⩽30	18	85<v⩽90
7	30<v⩽35	19	90<v⩽95
8	35<v⩽40	20	95<v⩽100
9	40<v⩽45	21	100<v⩽105
10	45<v⩽50	22	105<v⩽110
11	50<v⩽55	23	110<v⩽115
12	55<v⩽60	24	115<v⩽120

The taxi trajectory data in Beijing ring road during November 2012 served as the FCD data, obtained from 12,600 taxis. The features which were used for the experiment include speed, latitude, longitude, date and time. A sample of the dataset is shown in Table 2.

The total number of segments, being 7128, is lowered to 6926 following data filtering (Table 3). This is followed by dividing the 6926 roads into two parts for the sake of testing and training during timeslots 28 to 38 and 28 to 42 to roll time window of 8 and 12, respectively. The LSTM model provided by [10] and [5] applied loop detector data; and for comparing accuracy all models will be implemented employing the same FCD, accordingly.

Table 3: Number of Test and Train Samples

Number of samples		
Test	Train	Total
1386	5,540	6,926

4.2.1 Data of speed prediction in a road segment

In order to predict the speed of the next timestep in a single road segment, a time series sliding window is adopted to create overlapping sampling. As demonstrated in Figure 6, the window size is 6, which refers to the average speed of a road segment in 6 timesteps (starting timeslot=85). These speeds are used to predict the speed of the next timestep, which is the traffic speed in the next 5 minutes. In order to evaluate the traffic speed prediction method, the FCD dataset is divided into test set (20% of actual data) and train set (80% of actual data). The rolling window size of 6 and 8 are demonstrated in Figure 6. for two random road segments.

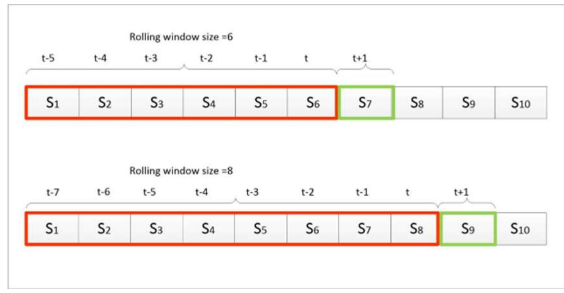

Figure 6: A sample of time series data sliding window (rolling window size of 6 and 8).

In time window size of 6, speed data of previous 6 timesteps are extracted (S_1 to S_6) to predict the speed at next timestep (t+1), which is S_7. Similarly, in time window size of 8 the speed data of previous 8 timesteps are extracted to predict the speed at the next timestep. The same process is done for timesteps 12, and 18.

Figure 7 illustrates the average speed of all road segments in FCD (Beijing) from timeslot 73 (6:00 AM) to timeslot 181 (3:00 PM). It helps to depict the pattern in traffic speed during a specific time.

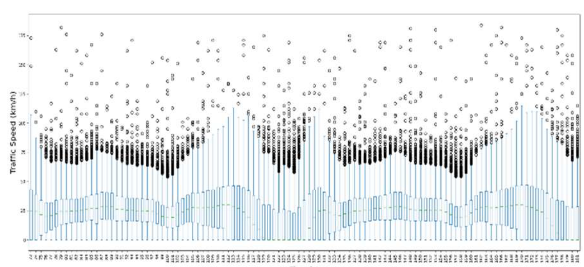

Figure 7: Average of traffic speed of all road segments in FCD (Beijing) from timeslot 73 (6:00 AM) to timeslot 181 (3:00 PM)

The author further illustrates the average speed of random 100 road segments between timeslot 73 to 181 (6:00 AM to 3:00 PM) in Figure 8. The dark red colour indicates low speed traffic, while the green colour indicates higher speed.

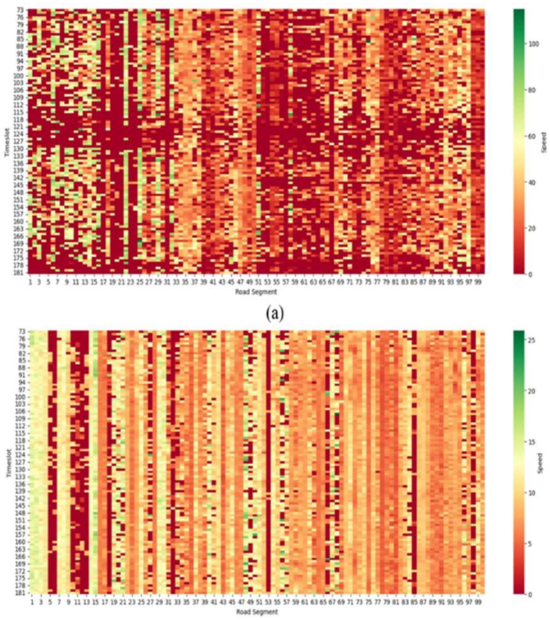

Figure 8: A sample of traffic speed data of random 100 road segments between timeslot 73 to 181 (6:00 AM to 3:00 PM) in two FCD datasets (a) FCD of Beijing, (b) FCD of Xuancheng

4.3 Hyperparameter

In order to obtain accurate predictions, it is required to set up two types of parameters: (1) Parameters for the input preparation, (2) hyper-parameters for the model.

(1) Parameters for the input preparation
The parameters include time intervals such as 5 minutes, and 15 minutes. This study follows [29] and set time intervals, which includes 30 minutes, 40 minutes, 1 hour, and 1 hour and 60 minutes, which refers to rolling window size 6, 8, 12, and 18, respectively.

(2) hyper-parameters for the model
The optimized Deep Learning-based structure is determined by evaluating its performance to identify the configuration which yields the least RMSE. In this study, the decision to opt for an LSTM layer and two FNN layers is based on the consideration that an excessive increase in the number of layers can lead to overfitting problems. To discover the optimized structure, a range of scenarios is defined by varying the number of hidden layers and the number of neurons within each hidden layer. These scenarios are presented in Table 4.

Table 4: Range of the parameters used to find optimal parameters

Hyperparameter	LSTM-C model
Optimiser	Adam
Learning rate	0.001
Activation function	ReLU in Feedforward NN layers Linear activation in output layer
Dropout rate	0.2
Batch size	64
Hidden layer size	128
Epoch	100
Loss	Mean squared error
L2 Regularization	0.08

In order to find the optimal optimizer, a grid search analysis is conducted. The number of neurons in the LSTM hidden layer is one of the following: 16,32, 64, 128, or 256. The following dense layer (First FNN) has one of the number of neurons as 50 or 40, and the second dense layer (second FNN) has a smaller number of neurons than the previous dense layer. The optimized learning rate is one of the following: 0.1, 0.01, 0.001, which are used in similar studies.

For setting batch size, the best value is found based on one of the following numbers: 32, 64, 128. Rectified Linear Unit (ReLU) is utilized as the activation function in the training. ReLU is widely used in ANN and DL applications. It does not require significant computational resources and straightforwardly produces the maximum value between zero and the input value.

Following a series of trial-and-error experiments, the optimal hyperparameters for the models to converge were determined as follows: a learning rate of 0.001, a batch size of 64, and 128, 50, and 30 neurons for each hidden layer, respectively. The dropout rate is set to 100 epochs. Moreover, to find a proper number of epochs early stopping technique is used. Specifically, the stop early strategy will terminate when the validation loss increases in 5 consecutive epochs. In addition, L2 regularization is used to avoid the problem of over-fitting.

The grid search is performed with various optimizers (Table 5). According to the results of grid search analysis, the RMSE of using RMSprop, Adam, Adadelta, and Adagrad as optimizers on the suggested Deep Neural Network structure are presented in Table 4. Based on the analysis, Adadelta and Adagrad have similar performance with RMSE of 6.871, and 6.641, respectively. Since Adam optimizer recorded the least RMSE, this study used Adam optimizer for model development and evaluation. The optimal structure refers to the specific number of neurons present in each hidden layer (LSTM, FNN1, FNN2).

Table 5: Using different optimizers on the model. The optimal structure refers to the number of neurons at each layer.

Optimizer Name	Optimal Structure	RMSE (km/h)
RMSProp	32/40/30	7.473
Adam	128//50/30	4.98
Adadelta	64/50/20	6.871
Adagrad	64/50/30	6.641

The hyperparameters of the proposed model are defined in Table 6.

4.4 Performance Measure

The model suggested is measured in terms of performance employing MAE (Mean Absolute Error), RMSE (Root Mean Square Error) and Mean Square Error (MSE) and Mean Absolute Percentage Error (MAPE). The aforementioned model is weighed up with the model already presented by [10]. Both models are exercised utilizing the same FCD and 15-min time window.
The calculation formula for MAE is represented in Eq. 10.

$$MAE = \left(\frac{1}{n}\right) * \sum_{i}^{n} |y_i - x_i| \qquad (10)$$

Where y_i is the actual speed value for the i^{th} road segment, x_i is the value predicted for the i^{th} road segment, and n is the total number as to the speed values predicted. MAPE is calculated by Eq. 11.

$$MAPE = \frac{1}{n} * \Sigma\left(\frac{|actual - Predicted|}{|actual|}\right) * 100 \qquad (11)$$

The average deviation between the predicted speeds and actual values is considered as MAPE. In the same manner as MAE, the average magnitude of error between predicted values and actual values is calculated. Percentages being simpler for people to understand MAE, MAPE enjoys a clear interpretation as well. Meanwhile, because of using absolute value, MAPE and MAE are resistant to outliers' effects. Eq. 12 describes the calculation of the RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Predicted - actual)^2}{n}} \qquad (12)$$

On the basis of Eq. 12, RMSE is the square root of the average in the squared differences between the suggested speed value and the actual one.

MSE, defined in Eq.13, determines the sum of squared difference between actual values and suggested ones.

$$MSE = \frac{1}{n} * \sum_{i}^{n}(Predicted - actual)^2 \qquad (13)$$

5. RESULTS

The present paper attempts to predict the traffic speed through the proposed LSTM-C model, spotting the level of traffic congestion utilizing the predicted speed. In the following section, the results obtained from predicted speed will be reported.

5.1 Speed Prediction

For the sake of prediction, both the average traffic speed and Contrast measure are provided by the proposed model. Each day (24 hours) will be divided into 96 timeslots starting 00:00 am and each timeslot being 5-min. For LSTM-C sequence, the lengths of rolling window size are set to three values (i.e., 6, 8, 12, and 18). When the rolling window size is set to 6, the previous 6 timesteps (t-6… t) will be utilized for predicting the subsequent timestep (t+1). This is the case with the rolling time window if set to 18 for foreseeing the following timestep (t+1) employing the previous 18 timeslots (t-18, …, t). ACC measure in Table 7 refers to (100 – MAE).

Moreover, the proposed model is applied to GRU (Gated Recurrent Unit) called GRU-C and the results are compared to the other models in Table 7. The proposed LSTM_C is believed to be outperforming the LSTM model presented by [10] and [5]. The results taken from all four models, namely LSTM-C, GRU-C, LSTM [10], LSTM [5] and GRU, are compared considering 6, 8, 12 and 18 rolling window size. In addition, GRU, like LSTM, is an RNN-based model, of which the implementation results are quite similar to LSTM. Mean Absolute Error (MAE) for LSTM-C, GRU-C, LSTM [10,5] and GRU are displayed as 3.303, 3.277, 5.094, 4.918 and 5.199, respectively, with the rolling window size, equaling to 6. As for the rolling window size of 8, the MAE rises to some degree in LSTM-C, GRU-C, LSTM [10], LSTM [5] and GRU up to 3.614, 3.304, 5.308, 5.078 and 5.417, respectively. While the best performance measurement refers to the rolling window size of 6, the results show that once rolling window size rises to 12 and 18, LSTM-C outperforms GRU-C, demonstrating the LSTM-C capability for predicting longer rolling window size. The average of actual and predicted traffic speed is sketched in Figure 9 for LSTM-C, GRU-C, LSTM [10], LSTM [5] and GRU models.

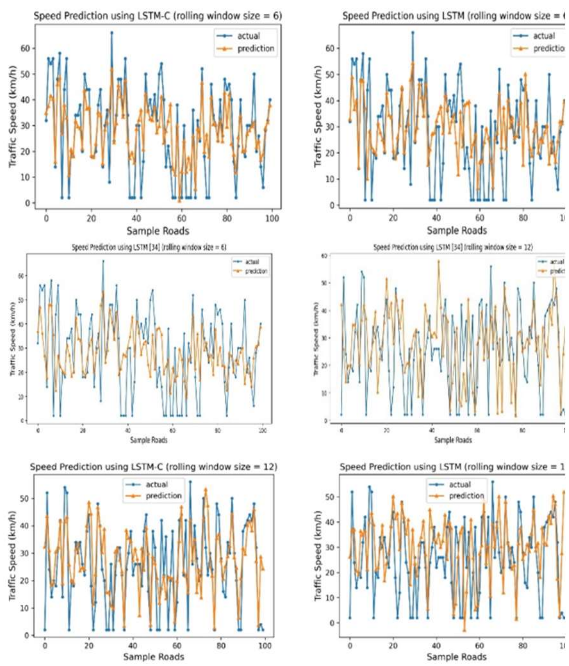

Figure 9: The average speed prediction for rolling window size 6 for LSTM-C, LSTM [10], LSTM [5], GRU-C and GRU

As shown in Figure 9, the accuracy of LSTM-C model for predicting the future speed patterns is to a high degree and mostly identifies the upcoming traffic speed correctly.

5.2. Propagation of Traffic Congestion

In the proposed LSTM-C model, the average traffic speed is predicted accurately. This stage aims to identify congestion level of traffic based on the average predicted speed by the proposed LSTM-C and will be compared with the congestion level of the actual average traffic speed. Table 8 shows the classification of traffic congestion level in Beijing arterial roads.

Table 9 indicates the results for spotting the congestion level of speeds (predicted as well as actual). The LSTM-C model accuracy being significant, the speeds predicted show close proximity to the actual speed, developing the likelihood of putting the predicted speed in the exact class as to the actual speed.

As stated by Table 9, the RMSE, MAE and MSE of LSTM-C are equal to 0.534, 0.259 and 0.285, respectively, with the rolling window size being 6. What is more, the congestion identification for greater rolling window sizes of 8,12 and 18 are determined. Despite the slight rise in measurements as the rolling window size grows, the performance of identifying congestion level is still high. By way of example, the results obtained from RMSE, MSE and MAE suggest 0.625, 0.391 and 0.347 respectively for the rolling window size of 18.

In addition, the speed prediction results produced from LSTM model provided by [10] are employed for identification of congestion level in roads. The model accurately measured by the rolling window of 6 includes 0.723, 0.522, 0. 0.451 for RMSE, MSE and MAE, respectively. Concerning the rolling window of 18, the accuracy measurements for RMSE, MSE and MAE represent 0.782, 0.611 and 0.482, respectively. The outcome achieved from spotting congestion level vividly divulges that the LSTM-C is particularly effective detecting the level of congestion.

5.3. Speed prediction in a road segment

Figure 10 shows the comparison between the real data (actual) and the predicted data of each model. This comparison is formed on the road segment number 1 and number 1000, considering the rolling window size being 6, in which road segment #1000 is specified by little changes in speed, whereas segment #1 suggests large and frequent speed fluctuations.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

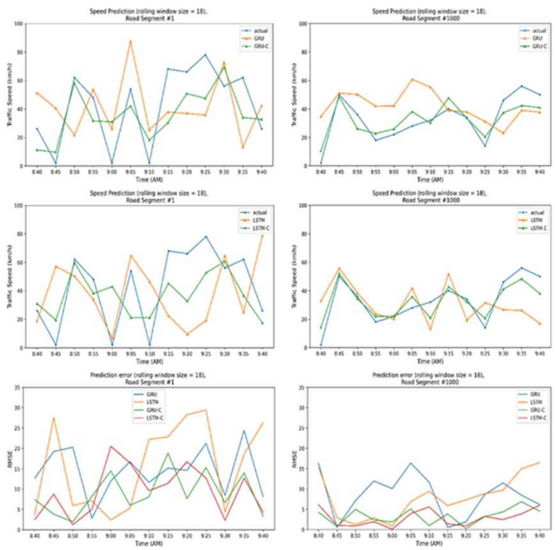

Figure 10. Comparison of predicted speed and RMSE between proposed model and benchmark

To predict the speed at the next timestep (Rolling window size = 6), 6 timesteps are applied, where the left part exhibits Road segment #1 with large and frequent speed fluctuation. Yet, this is not the case with the right part in that speed shows relatively low changes when the road segment is #1000.

5.4 Discussion

The study by [10] proposed single variable (traffic speed) and multi variables (traffic speed and vehicle headway) as features to their LSTM model. They concluded that multivariate LSTM does not add any significant contribution to adequately predicting traffic speed compared to single variate LSTM. However, in this study, concatenating additional features to an LSTM-based model improved the speed prediction, which shows the relevancy of Contrast Measure and its effect in the proposed model which helps the model to gain more insights and improve the accuracy. Concatenating the relevant features in the model can potentially improve the accuracy of model, or it can introduce more noise which leads to degrading the performance of model. Therefore, this study proposed and investigated the use of Contrast Measure and based on the results, the proposed Contrast Measure is able to help model's ability to learn meaningful patterns from changes (increasing and decreasing) in speed.

This result directly addresses the research objectives of improving accuracy through additional feature engineering and enhancing performance using FCD. Each major objective, including the use of LSTM-C, integration of contrast measure, and comparison with existing models, has shown to positively impact the accuracy of predictions. The proposed model consistently outperforms benchmarks across MAE, RMSE, and MAPE, validating the study's hypoth Compared to the state-of-the-art models in literature (LSTM, GRU, CNN-GRU, AGNP, D-LSTM), LSTM-C offers a simpler yet more interpretable framework while achieving superior accuracy with limited input features. This supports the notion that statistical enhancements like contrast can offer performance gains without relying on excessive data fusion.

This study focuses solely on short-term speed prediction using historical traffic speed and contrast features derived from FCD. External influences such as weather, road blockages, and driver behavior were not modeled. Moreover, the model's performance was tested only on Beijing FCD, which may affect generalizability to other cities or countries.

6. CONCLUSIONS AND FUTURE WORK

This paper presented an improved LSTM-based model, called LSTM-C, to predict average traffic speed in a 5-min time window using FCD in Beijing. A total of 6926 road segments were used in the experiment. The contrast measure was incorporated into LSTM to enhance prediction accuracy. In the traffic context, contrast captures increases and decreases in speed using positive and negative signs. MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and MSE (Mean Square Error) were used to evaluate the performance of the proposed model. Experimental results show that LSTM-C achieves up to 96.697% accuracy (based on MAE) with a rolling window size of 6, demonstrating the strong potential of LSTM in traffic prediction. Moreover, the model was also applied to GRU (Gated Recurrent Unit), termed GRU-C, achieving 96.723% accuracy for the same window size. The speed prediction results were compared with the LSTM model by [10], which reached 94.906% under identical settings. Furthermore, the predicted speed from LSTM-C was used to identify congestion levels, achieving high precision with MAE = 0.259 (rolling window = 6).

The proposed method relies solely on location, speed, and time. For future work, we plan to enhance the analysis by integrating multi-source traffic data. Given the volume of updates on social media platforms, user-generated data—such as weather conditions or reported congestion—can improve accuracy. Additionally, incorporating road network information, including speed limits, lane closures, and other restrictions, could further refine predictions. Another potential improvement is exploring longer time intervals by combining LSTM with generative models that simulate varied traffic behavior over time.

This study marks the first known integration of an improved contrast measure within an LSTM-based traffic prediction model using FCD. It shows that statistical contrast features can enhance deep learning performance in traffic forecasting. The model outperforms two established LSTM benchmarks and proves applicable to GRU, validating a novel feature engineering method to address FCD limitations.

We also plan to integrate multi-source data, including weather and social media content. Future exploration may involve graph-based neural networks and hybrid generative models to capture longer time dependencies and contextual patterns.

REFERENCES:

[1] Ahanin, F., Mustapha, N., Zolkepli, M., & Husin, N. A. (2023). A Review of Traffic State Prediction (TSP) Methods in Intelligent Transportation Systems (ITS). International Journal of Academic Research in Business and Social Sciences, 13(3). https://doi.org/10.6007/ijarbss/v13-i3/16683

[2] Do, V. M., Tran, Q. H., Le, K. G., Vuong, X. C., & Vu, V. T. (2024). Enhanced Deep Neural Networks for traffic speed forecasting regarding sustainable traffic management using probe data from registered transport vehicles on Multilane Roads. Sustainability, 16(6), 2453. https://doi.org/10.3390/su16062453

[3] Xu, M., Di, Y., Ding, H., Zhu, Z., Chen, X., & Yang, H. (2023). AGNP: Network-wide short-term probabilistic traffic speed prediction and imputation. Communications in Transportation Research, 3, 100099. https://doi.org/10.1016/j.commtr.2023.100099

[4] Tran, Q. H., Fang, Y.-M., Chou, T.-Y., Hoang, T.-V., Wang, C.-T., Vu, V. T., Ho, T. L., Le, Q., & Chen, M.-H. (2022). Short-term traffic speed forecasting model for a parallel multi-lane arterial road using GPS-monitored data based on Deep Learning Approach. Sustainability, 14(10), 6351. https://doi.org/10.3390/su14106351

[5] Gao, Y., Zhou, C., Rong, J., Wang, Y., & Liu, S. (2022). Short-Term Traffic Speed Forecasting Using a Deep Learning Method Based on Multitemporal Traffic Flow Volume. IEEE Access, 10, 82384–82395. https://doi.org/10.1109/ACCESS.2022.3195353

[6] Tran, Q. H., Fang, Y.-M., Chou, T.-Y., Hoang, T.-V., Wang, C.-T., Vu, V. T., Ho, T. L., Le, Q., & Chen, M.-H. (2022). Short-term traffic speed forecasting model for a parallel multi-lane arterial road using GPS-monitored data based on Deep Learning Approach. Sustainability, 14(10), 6351. https://doi.org/10.3390/su14106351

[7] Malik, A., Khan, M. Z., Qaisar, S. M., Faisal, M., & Mehmood, G. (2023). An efficient approach for the detection and prevention of gray-hole attacks in VANETs. IEEE Access, 11, 46691–46706. https://doi.org/10.1109/access.2023.3274650

[8] Mehmood, G., Khan, M. Z., Bashir, A. K., Al-Otaibi, Y. D., & Khan, S. (2023). An efficient QoS-based multi-path routing scheme for smart healthcare monitoring in Wireless Body Area Networks. Computers and Electrical Engineering, 109, 108517. https://doi.org/10.1016/j.compeleceng.2022.108517

[9] Ahanin, F., Mustapha, N., & Zolkepli, M. (2020). AN EFFICIENT TRAFFIC STATE ESTIMATION MODEL BASED ON FUZZY C-MEAN CLUSTERING AND MDL USING FCD. Journal of Theoretical and Applied Information Technology, 31(14). www.jatit.org

[10] Majumdar, S., Subhani, M. M., Roullier, B., Anjum, A., & Zhu, R. (2021). Congestion prediction for smart sustainable cities using IoT and machine learning approaches. Sustainable Cities and Society, 64(September 2020). https://doi.org/10.1016/j.scs.2020.102500

[11] Akhtar, M., & Moridpour, S. (2021). A Review of Traffic Congestion Prediction Using Artificial Intelligence. Journal of Advanced Transportation, 2021. https://doi.org/10.1155/2021/8878011

[12] Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., & Hellinckx, P. (2020). A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transportation Research Part C: Emerging

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Technologies, 112, 62–77. https://doi.org/10.1016/J.TRC.2020.01.010

[13] Zaki, J. F., Ali-Eldin, A. M. T., Hussein, S. E., Saraya, S. F., & Areed, F. F. (2019). Time Aware Hybrid Hidden Markov Models for Traffic Congestion Prediction. International Journal on Electrical Engineering and Informatics, 11(1). https://doi.org/10.15676/ijeei.2019.11.1.1

[14] Zhang, Y., & Ye, Z. (2008). Short-Term Traffic Flow Forecasting Using Fuzzy Logic System Methods. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 12(3), 102–112. https://doi.org/10.1080/15472450802262281

[15] Zaki, John F., Ali-Eldin, A., Hussein, S. E., Saraya, S. F., & Areed, F. F. (2020). Traffic congestion prediction based on Hidden Markov Models and contrast measure. Ain Shams Engineering Journal, 11(3), 535–551.

[16] Rodrigues, F., Henrickson, K., & Pereira, F. C. (2019). Multi-output gaussian processes for crowdsourced traffic data imputation. IEEE Transactions on Intelligent Transportation Systems, 20(2), 594–603. https://doi.org/10.1109/tits.2018.2817879

[17] Lin, L., Li, J., Chen, F., Ye, J., & Huai, J. (2018). Road traffic speed prediction: A probabilistic model fusing multi-source data. IEEE Transactions on Knowledge and Data Engineering, 30(7), 1310–1323. https://doi.org/10.1109/tkde.2017.2718525

[18] Zhu, Lin, Krishnan, R., Guo, F., Polak, J. W., & Sivakumar, A. (2019). Early Identification of Recurrent Congestion in Heterogeneous Urban Traffic. 2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, 4392–4397. https://doi.org/10.1109/ITSC.2019.8916966

[19] Sun, Shiliang, Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems, 7(1),

[20] Kim, J., & Wang, G. (2016). Diagnosis and Prediction of Traffic Congestion on Urban Road Networks Using Bayesian Networks: Https://Doi.Org/10.3141/2595-12, 2595, 108–118. https://doi.org/10.3141/2595-12

[21] Dave, V. S., & Dutta, K. (2012). Neural network based models for software effort estimation: A Review. Artificial Intelligence Review, 42(2), 295–307. https://doi.org/10.1007/s10462-012-9339-x

[22] Yang, X., Luo, S., Gao, K., Qiao, T., & Chen, X. (2019). Application of Data Science Technologies in Intelligent Prediction of Traffic Congestion. Journal of Advanced Transportation, 2019. https://doi.org/10.1155/2019/2915369

[23] Nadeem, K. M., & Fowdur, T. P. (2018). Performance analysis of a real-time adaptive prediction algorithm for traffic congestion. Journal of Information and Communication Technology, 17(3), 493-511.

[24] Navarro-Espinoza, A., López-Bonilla, O. R., García-Guerrero, E. E., Tlelo-Cuautle, E., López-Mancilla, D., Hernández-Mejía, C., & Inzunza-González, E. (2022). Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms. Technologies, 10(1), 5. https://doi.org/10.3390/technologies10010005

[25] Zhang, P., & Qian, Z. (Sean). (2018). User-centric interdependent urban systems: Using time-of-day electricity usage data to predict morning roadway congestion. Transportation Research Part C: Emerging Technologies, 92, 392–411. https://doi.org/10.1016/J.TRC.2018.05.008

[26] Lonare, S., & Bhramaramba, R. (2020). Traffic Flow Prediction Using Regression and Deep Learning Approach. In New Trends in Computational Vision and Bio-inspired Computing (pp. 641–648). Springer International Publishing. https://doi.org/10.1007/978-3-030-41862-5_63

[27] Chen, C., Li, K., Teo, S. G., Zou, X., Wang, K., Wang, J., & Zeng, Z. (2019, July). Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 485-492).

[28] Liu, Y., & Wu, H. (2018). Prediction of road traffic congestion based on random forest. Proceedings - 2017 10th International Symposium on Computational Intelligence and Design, ISCID 2017, 2, 361–364. https://doi.org/10.1109/ISCID.2017.216

[29] Sun, Shuming, Chen, J., & Sun, J. (2019). Traffic congestion prediction based on GPS trajectory data. International Journal of Distributed Sensor Networks, 15(5). https://doi.org/10.1177/1550147719847440

[30] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. Sensors 2017, Vol. 17, Page 818, 17(4), 818. https://doi.org/10.3390/S17040818

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[31] Chen, M., Yu, X., & Liu, Y. (2018). PCNN: Deep convolutional networks for short-term traffic congestion prediction. IEEE Transactions on Intelligent Transportation Systems, 19(11), 3550-3559.

[32] Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2018). Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.

[33] Chen, C., Li, K., Teo, S. G., Zou, X., Wang, K., Wang, J., & Zeng, Z. (2019, July). Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 485-492).

[34] Hu, Z., Sun, R., Shao, F., & Sui, Y. (2021). An efficient short-term traffic speed prediction model based on improved TCN and GCN. Sensors, 21(20), 6735.

[35] Zhang, T., Liu, Y., Cui, Z., Leng, J., Xie, W., & Zhang, L. (2019). Short-Term Traffic Congestion Forecasting Using Attention-Based Long Short-Term Memory Recurrent Neural Network. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11538 LNCS, 304–314. https://doi.org/10.1007/978-3-030-22744-9_24

[36] Meng, X., Fu, H., Peng, L., Liu, G., Yu, Y., Wang, Z., & Chen, E. (2022). D-LSTM: Short-term road traffic speed prediction model based on GPS positioning data. IEEE Transactions on Intelligent Transportation Systems, 23(3), 2021–2030. https://doi.org/10.1109/tits.2020.3030546

[37] Ishak, S. (2003). Deriving Traffic-Performance Measures and Levels of Service from Second-Order Statistical Features of Spatiotemporal Traffic Contour Maps. Transportation Research Record, 1858, 148–157. https://doi.org/10.3141/1858-22

[37] Qi, Y., & Ishak, S. (2014). A Hidden Markov Model for short term prediction of traffic conditions on freeways. Transportation Research Part C: Emerging Technologies, 43, 95–111. https://doi.org/10.1016/J.TRC.2014.02.007

[38] Y. Chen, C. Chen, Q. Wu, J. Ma, G. Zhang, and J. Milton, “Spatial-temporal traffic congestion identification and correlation extraction using floating car data,” J. Intell. Transp. Syst. Technol. Planning, Oper., vol. 0, no. 0, pp. 1–18, 2020, doi: 10.1080/15472450.2020.1790364.

Table 2: The dataset input features

Vehicle ID	Date & Time	Longitude	Latitude	Speed	Direction	Status
426242	20121101051340	116.3610535	39.6585464	17	342	1
426242	20121101051430	116.3607025	39.6595154	25	166	1
426242	20121101051526	116.3611298	39.6567192	0	162	1
426242	20121101051621	116.3611298	39.6567574	6	342	1

Table 6: LSTM-C, LSTM, and GRU Hyperparameter

Hyperparameter	LSTM-C	LSTM	GRU
Optimiser	Adam	Adam	Adam
Learning rate	0.001	0.001	0.001
Activation function	ReLU in Feedforward NN layers Linear activation in output layer	ReLU in Feedforward NN layers Linear activation in output layer	ReLU in Feedforward NN layers Linear activation in output layer
Dropout rate	0.2	0.2	0.2
Batch size	64	64	64
Hidden layer size	128	128	128
Epoch	100	100	100
Loss	Mean Squared Error	Mean Squared Error	Mean Squared Error
L2Regularization	0.08	0.08	0.08

Table 7: Performance comparison of the proposed LSTM-C model with LSTM [10], LSTM [5], GRU-C and GRU

		Input units (rolling window)			
		6	8	12	18
MAE	LSTM-C	3.303	3.614	3.489	4.075
	GRU-C	3.277	3.304	3.631	4.238
	GRU	5.199	5.417	5.094	6.086
	LSTM [5]	4.918	5.078	4.886	5.536
	LSTM [10]	5.094	5.308	5.227	5.870
RMSE	LSTM-C	4.92	5.614	5.118	5.810
	GRU-C	4.869	5.044	5.319	6.112
	GRU	7.218	8.240	7.441	8.686
	LSTM [5]	7.014	7.641	7.144	7.932
	LSTM [10]	7.11	8.008	7.512	8.427
MSE	LSTM-C	24.227	31.515	26.192	33.759
	GRU-C	23.709	25.442	28.291	37.352
	GRU	52.098	67.891	55.371	75.440
	LSTM [5]	49.194	58.387	51.044	62.911
	LSTM [10]	50.545	64.129	56.424	71.006
MAPE	LSTM-C	1.860	2.113	1.785	2.210
	GRU-C	1.773	2.171	1.868	2.287
	GRU	2.577	3.034	2.609	3.242
	LSTM [5]	2.501	2.965	2.438	3.172
	LSTM [10]	2.564	2.958	2.517	3.290
ACC	LSTM-C	96.697	96.386	96.511	95.925
	GRU-C	96.723	96.696	96.369	95.762
	GRU	94.801	94.583	94.906	93.914

		Input units (rolling window)			
		6	8	12	18
	LSTM [5]	95.082	94.922	95.114	94.464
	LSTM [10]	94.906	94.692	94.773	94.13

Table 8: Classification of traffic congestion level in Beijing arterial roads [38]

Grade	Fast	Smooth	Light Congestion	Medium Congestion	Severe Congestion
Speed	$\bar{v}>85$	$65<\bar{v} \leq 85$	$45<\bar{v} \leq 65$	$25<\bar{v} \leq 45$	$\bar{v} \leq 25$

Table 9: Congestion level identification of the proposed model with LSTM model [10]

		Input units				
		6	8		12	18
MAE	LSTM-C	0.259	0.211		0.307	0.347
	GRU-C	0.253	0.267		0.324	0.347
	LSTM [10]	0.451	0.447		0.453	0.482
RMSE	LSTM-C	0.534	0.475		0.586	0.625
	GRU-C	0.527	0.547		0.602	0.629
	LSTM [10]	0.723	0.765		0.743	0.782
MSE	LSTM-C	0.285	0.225		0.343	0.391
	GRU-C	0.278	0.299		0.324	0.396
	LSTM [10]	0.522	0.585		0.552	0.611