International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I5P216

Volume 71 Issue 5, 156-165, May 2023

Original Article

Long-Term Traffic Flow Prediction using Hybrid Deep Learning Technique

Mohandu Anjaneyulu[1], Mohan Kubendiran[2]

[1, 2]*School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology (VIT), Vellor, India.*

[1]*Corresponding Author: mohan.k@vit.ac.in*

Received: 17 February 2023 Revised: 07 May 2023 Accepted: 13 May 2023 Published: 25 May 2023

Abstract *- Smart city traffic regulation relies heavily on advanced traffic management systems (ATMS), a key component of the broader intelligent transportation system (ITS). Traffic flow forecasting is a crucial aspect of transportation that aids in traffic planning, control, management, and information dissemination. Although there are a great variety of models whose primary focus is on the development of short-term traffic flow predictions, making credible long-term traffic flow (LTTF) forecasts has become an increasingly difficult topic in recent years. To solve this problem, this paper proposed a novel hybrid model called the autoencoder gated recurrent unit (AEGRU) that can accurately predict long-term traffic flow for the next 24 hours. Firstly, the autoencoder (AE) will take the raw data and pick out the most important features before doing dimensionality reduction. Secondly, the gated recurrent unit (GRU) uses the information given by the AE to make predictions about how much traffic volume there will be in the future. The outcomes of the evaluation show that the proposed AEGRU model is much better than compared approaches in terms of root mean square error (RMSE) of 1.6% mean absolute percentage error (MAPE) of 2.3% and mean absolute error (MAE) of 1.9%.*

Keywords *- Traffic flow prediction, Long-term traffic flow prediction, Autoencoder, Gated recurrent unit, Neural networks, Deep learning.*

1. Introduction

A well-informed transport system enables efficient route planning, estimates travel duration, and mitigates traffic congestion. Typically, external factors and spatiotemporal data are the cause of the erratic and unpredictable movement of traffic. One of the most difficult problems in transportation studies is predicting traffic flow prediction (TFP). Precise and reliable traffic flow prediction is essential for the scientific management and control of the highway, allowing drivers to plan their routes and times of travel better, make more informed decisions about how and when to travel, save money, and help the government implement cutting-edge traffic management strategies to alleviate gridlock [1].

The art of foretelling the future is one of the most fascinating aspects of analysing human nature, and the same is evident in the administration of transit systems. Realizing how traffic changes across the whole roadway, not just on a single road, is very important and interesting. This will help people who have more traffic information choose better routes and help traffic managers run a road network and allocate resources in a logical manner [2]. Traffic flow is the number of automobiles that use a road at frequent intervals. The objective of TFP is to forecast how traffic will be in the forthcoming by looking at data on traffic flow. Most of the time, short-term traffic forecasting is applied to control traffic in the actual moment. In most scenarios, it just isn't up to the task of making decisions. Better traffic management and service quality are possible thanks to early judgments, measures, and overall arrangements informed by accurate and timely long-term forecasting. The information it gives can be used to manage traffic better and help drivers plan for places where there is a lot of traffic [3]. Current technological innovation and transport systems, like a strong backup for the Global Positioning System, Internet of Things, wireless sensing, and mobile communications, make urban areas increasingly connected, digitalized, and thus smart.

The flow of traffic in cities is one of the most interesting applications of smart cities. During a certain time period, the traffic flow is calculated by tallying the total number of vehicles, pedestrians, taxis, buses, etc., passing through a specific area. Advanced learning algorithms for predicting traffic patterns have been proposed [4]. The widespread adoption of sensor technologies has just ushered in the era of big traffic data. The dominant way for making predictions now is based on data rather than models; it's called "deep learning." Deep learning has grown in reputation because of the fast growth of artificial intelligence. It has been used successfully to classify images, recognize speech, and

process natural language. Congestion, accidents, and delays in urban areas all come from the ever-increasing car population, putting a heavy burden on the city's mass transit. The best way to deal with traffic jams is to create an effective strategy for managing traffic according to projections of how traffic will move [5]. In the past, single-time-step methods for predicting traffic flow were the main focus. However, multi-time-step models for predicting traffic flow have become very important for many reasons, including their scientific value. It is crucial for the transportation management method and assessment of traffic flow as part of the intelligent transport system, and there is an early warning system that can accurately predict how traffic will move in the long run. [6]. Furthermore, most of the current approaches can only provide short-term traffic flow prediction, and reliable long-term traffic flow prediction methods are still in short supply.

Due to how quickly errors spread, it's harder to make long-term traffic predictions than short-term traffic predictions. Using long-term forecasts rather than short-term projections can reduce average travel times. Thus, it is of major importance for scientific management to give medium- and long-term forecasts with improved precision so that more data may be used to support decision-making. The major issue with making long-term predictions is that mistakes are easy to make and add up over time. The iteration method is used to make many model predictions. This means that each model prediction will return the input as a new input. By moving the window, the actual information will get smaller, the forecast data will get bigger, and the error desire will add up and get bigger, which will affect the algorithm's ability to make long-term predictions. Minimizing the effect as a result of these errors, the forecast can be more precise and lengthen the moment it takes to make a prediction [7].

Recent progress in this area of artificial intelligence research has made it more common to use deep neural network models to predict how traffic will move [8]. Deep learning methods are preferred to make predictions and conduct time series data analyses without using predetermined features. Using multiple hidden layers in an artificial neural network, deep learning is a subset of machine learning able to learn the crucial information needed to make a response. Extracting features is the primary distinction that can be made between DL and ML. In addition to discovering features through model training, deep learning is significantly more accurate at predicting future outcomes than competing algorithms. Deep learning, on the other hand, does more than just find features through model training [9].

This paper follows the structure described below. Section 2 presents the literature review. In Section 3 talks about the Materials and Methods that make up the proposed AEGRU framework. The Implementation work is covered in Section 4. Experiments were conducted to verify the suggested method, and the results and analysis of those experiments are reported in Section 5. Section 6 contains the Conclusion and recommendations for future work.

2. Literature Review

One of the biggest challenges of advanced traffic management is predicting traffic flows reliably. Both the total area of land in metropolitan areas and the number of cars on the road have increased to the point where they are at capacity. Because of this, road transit became extremely difficult, which led to an ecological imbalance as well as a threat to human life on the road. Some of the problems have been solved with the help of modern innovations. Predicting traffic conditions in advance aids in journey planning, trip time estimation, and gridlock avoidance. In most cases, external factors are responsible for the unpredictable and disorderly behaviour of traffic. Some of the problems were brought under control when various forms of technology were introduced. The necessity of traffic forecast has led to the development of numerous methods, which can be grouped into two broad categories: parametric and nonparametric. Van et al. [10] proposed the KARIMA method, a hybrid approach to performing short-term traffic forecasting. Autoregressive integrated moving average (ARIMA) is a type of parametric model. The method employs Kohonen's self-organizing map while its foundational overview; furthermore, the ARIMA model is fine-tuned for each class separately.

If you use a Kohonen map with a hexagonal layout, specifying the classes becomes much simpler. Forecasting performance is significantly enhanced in comparison to the ARIMA framework or backpropagation neural network since categorization and functional approximation are explicitly separated. This study shows how the model works by predicting traffic on a French highway for the next 30 minutes and the next hour. Chan et al. [11] proposed a new Levenberg-Marquardt (LM) approach, and the hybrid exponential smoothing approach are applied to train neural networks (NNs) with the goal of enhancing the generalization capabilities of existing NN training methods for short-term traffic flow forecasting. The neural network method's weights are trained with a modified version of the LM technique after the data has been pre-processed with exponential smoothing to remove the lumpiness of acquired traffic flow data. Kumar [12] proposed ARIMA or seasonal ARIMA model being, forecasting traffic flow requires a lot of traffic data to build a methodology.

However, it could fail to work if there isn't enough traffic data. In order to solve this issue, a forecasting method that is fixed, the Kalman filtering method (KFT), was recommended. This method makes less information than what is often used; therefore, it is more efficient. The only data that is used by the KFT-based prediction system to make an estimate of the flow values for the next 24 hours with a certain degree of accuracy is data from the previous two days.

Zhao et al. [13], an abundance of traffic data has shown the limitations of many tried-and-true techniques when applied to real-world scenarios with unpredictable traffic and intricate road layouts. When the traffic flow changes randomly and nonlinearly, it's easy to see where the parameter model's prediction went wrong. Nonparametric techniques have grown in popularity as a means of addressing this issue. Some important discoveries from the past few years have been put into models that predict traffic. Hou et al. [14] proposed the transit time in a network can be predicted using a random forest estimator. Two years worth of journey time data were used to train the estimator, and both temporal and spatial impacts were taken into account during modelling. The random forest models gave accurate estimates of how long trips would take in both heavy and light traffic situations.

Mingheng et al. [15] proposed a support vector machine (SVM)-based, an accurate multi-step model that may be used to anticipate the flow of traffic in which the input vectors were made up of the real traffic volume and four distinct types of input vectors were contrasted to determine how well they predicted the same quantity of actual traffic. Sun et al. [16] came up with a Bayesian network-based way to estimate traffic flow between two nearby roads that meet in a transportation system. For instance, in a Bayesian network, the joint probability distribution among the cause nodes alongside the effect nodes is expressed as a Gaussian mixture model (GMM), and its variables are computed via the competitive expectation maximisation (CEM) method. Sun et al. [17] offer a dynamic procedure-k-nearest neighbour (DP-KNN) that allows the KNN features to be adaptable as well as powerful, lacking the need for any sort of training or pre-existing models. Isravel et al. [18] proposed a multivariate time series framework that analyses and forecasts SDN traffic flow.

The proposed architecture modifies randomised singular value decomposition (RSVD) with multivariate singular spectrum analysis (MSSA) through increased traffic flow prediction accuracy. From observed traffic traces, the proposed technique predicts long-term traffic fluctuations. Future traffic flows are anticipated using the SDN controller's traffic records. Peng et al. [19] came up with a way to predict long-term traffic flow using dynamic graphs. Dynamic traffic flow probability graphs represent the traffic network. These graphs undergo graph convolution to learn spatial characteristics, and long short-term memory (LSTM) units are then used a focus on learning temporal attributes. When the dynamic graphs aren't complete, this article suggests using a graph convolutional policy network trained using reinforcement learning to fill in the gaps as a result of the data sparsity issue. Belhadi et al. [20] proposed a technique called recurrent neural networks with long-term traffic flow (RNN-LF) to anticipate future flows from a variety of data sources. Parallel implementation details for the suggested solution architecture using graphics processing units are also provided, which makes it possible to improve the performance of RNN-LF. Qu et al. [21] proposed a technique for forecasting daily traffic based on past traffic patterns and other environmental factors using a deep neural network. Points are made in the following directions regarding the research gaps found in the literature review.

- Lack of a recent survey in LTTF forecasting considering vital external factors like weather information and special events.
- Limited exploration of capturing the extreme values in time series data.
- Accuracy problem when dealing with temporal and spatial information.
- Limitations of extracting the best features from the dataset using advanced learning techniques.

2.1. Contributions

The most important contributions of this paper are summed up in the following points.

- This work provided an AE for extracting meaningful attributes from unprocessed data.
- This paper suggests GRU uses the information from the AE to predict how much traffic there will be in the future.

3. Materials and Methods

As a result, in this section, an AEGRU approach was developed for conducting a long-term traffic flow prediction analysis over the next twenty-four hours, as depicted in Figure 1. Additional information on the proposed AEGRU framework is outlined in the following subsections.

3.1. Autoencoder

AE are a form of artificial neural network (ANN) used to learn unsupervised data encodings. Simple learning circuits called autoencoder to convert inputs to outputs with little noise. Autoencoders were initially used by Hinton and the PDP group during the 1980s to address the issue of "backpropagation." The goal of an autoencoder is to learn a pattern (encoding) for data with higher dimensions, often for the purpose of dimension reduction. Autoencoders are a key part of both unsupervised learning and deep architectures that use transfer learning and network training. A dimensionality reduction is the same thing as an autoencoder. Dimensionality reduction is apparently used in data pre-processing (reduce or compress). The number of dimensions in a dataset is drastically reduced by the process of "dimensionality reduction." There could be a lot of information in the dataset being looked at. It doesn't use all of those features and uses some of them and then needs to figure out which features [22]. Autoencoder has the three-part architecture depicted in above Figure 1.

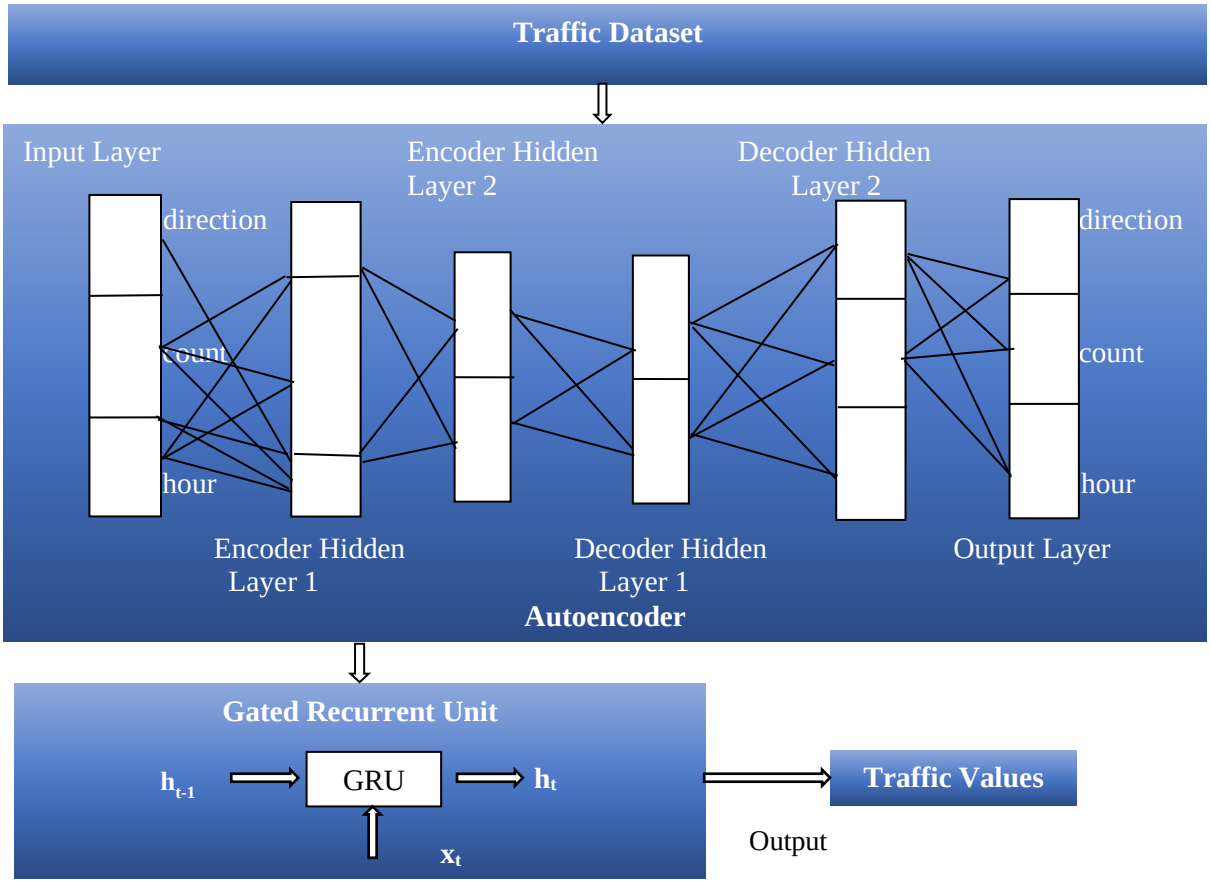

Fig. 1 Autoencoder gated recurrent unit

3.1.1. Input Layer (Encoder)

A component that reduces the size of the input data for the train, validate, and test sets by encoding it into a format that is often many orders of magnitude smaller than the original data.

3.1.2. Hidden Layer (Bottleneck)

A component that stores the condensed representations of the accumulated body of knowledge is; as a result, the most vital component of the network.

3.1.3. Output Layer (Decoder)

A module that assists in "decompressing" the knowledge representations held inside the network and reconfiguring the data from its original encoded state. After that, the results are contrasted with the ground truth. Training autoencoder.

3.2. Hyperparameters to Train an Autoencoder

3.2.1. Code size

When tuning an autoencoder, the code size (or bottleneck size) is the most crucial hyperparameter. The size of the bottleneck will determine how much data needs to be compressed. Moreover, it can be used as a regularization phrase.

3.2.2. Number of Nodes per Layer

Autoencoders are neural networks; hence, the depth of the neural network used for both the encoder and the decoder is a crucial hyperparameter for tuning. The model becomes more complicated as its depth increases, although the model may be processed more quickly at lesser depths.

3.2.3. Reconstruction Loss

To train an autoencoder, first, choose a loss function that best fits the input and output data types. Both MSE losses and L1 losses have become standards when it comes to reconstructing the data shown in Equation 1. The reconstruction loss is expressed in terms of the binary cross entropy with input and output in the range [0, 1].

$$(X, Y\hat{}') = \|Y - Y\hat{}'\| = \|Y - \sigma\hat{}'(W\hat{}'(\sigma(W\_r + b)) + b\hat{}')\|\hat{}2 \quad (1)$$

where

Y – original input,
Y' – reconstructed input,
σ – activation function,
σ' - loss function,
W – weight,
b – bias.

3.3. Gated Recurrent Unit

After RNN and LSTM, the Gated Recurrent Unit is the most recent method for modelling sequences. Because it is the newest competitor, it offers an upgrade over the other two techniques. In 2014, Kyunghyun Cho proposed GRU. GRU doesn't have a specific cell state called Ct. Instead, it has a hidden state called H_t. Because of the less complex architecture, GRUs can be trained much more quickly. The gates are to receive a value that is between zero and one. The fact that the gate returned a 0 indicates that the data is insignificant. One states that it is essential (closer to 0, it is unimportant, closer to 1, it is important). GRU comprises two gates, which are labelled the reset gate and the update gate, shown below in Figure 2.

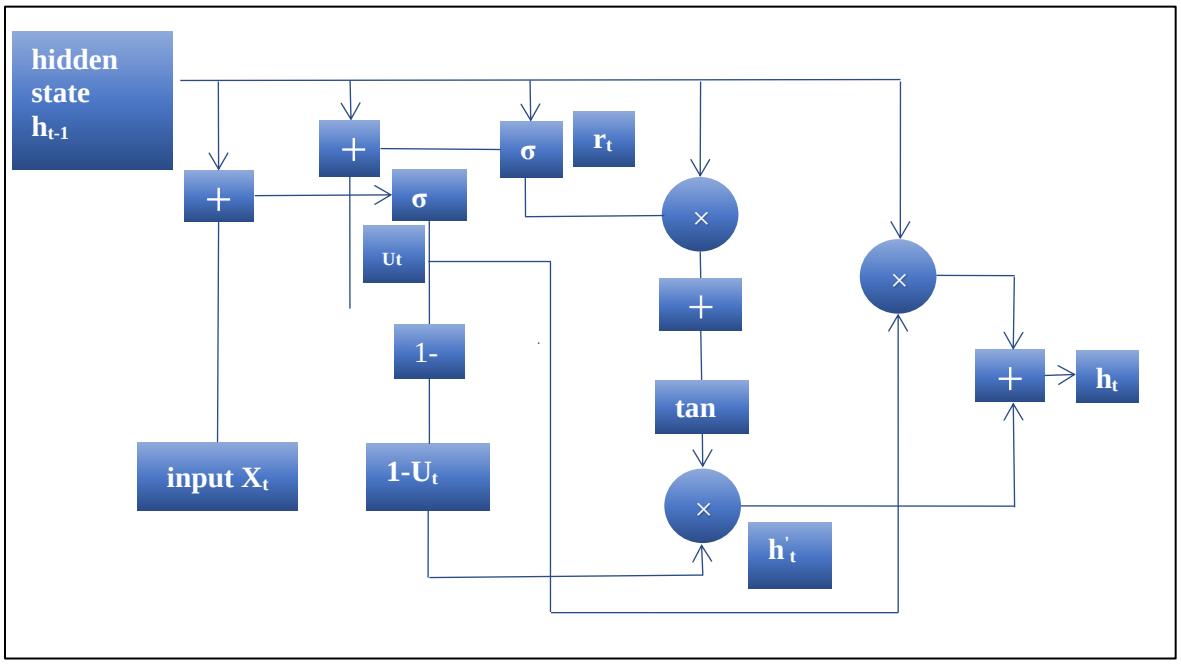

Fig. 2 Gated recurrent unit

3.3.1. Reset Gate (Short-term Memory)

Reset Gate is crucial for short-term memory, with the hidden state (H_t) utilized by the model to determine how much historical data to forget. Reset gate symbolized by r_t. Parameters are input denoted by X_t and prior state defined by H_{t-1} data multiplied by corresponding weights. Due to the sigmoid function, the value of r_t will vary between 0 and 1. A sigmoid activation is employed to calculate r_t in Equation 2.

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \quad (2)$$

where

r_t – reset gate,
σ – Sigmoid activation function,
W^r –Weights,
U^r – Weights,
X_t - input,
h_{t-1} – previous state information.

3.3.2. Update Gate (Long-Term Memory)

The most important job of the update gate is to provide the model with information regarding "how much of the information from the past has to be kept," which is another way of saying "must be passed along to the future." The update gate is denoted by U_t, shown in Equation 3, the input is denoted by X_t, and information about the state before the most recent one is multiplied by relevant weights to generate the parameters.

$$u_t = \sigma(W^u x_t + u^u h_{t-1}) \quad (3)$$

where

u_t – update gate,
σ – Sigmoid activation function,
W^u –Weights,
u^u – Weights,
X_t - input,
h_{t-1} – previous state information.

3.3.3. Working of GRU

As a result, GRU gates have two distinct phases: the hidden state and the candidate hidden state. The candidate's hidden state is initially derived.

3.3.4. Candidate Hidden State

The end result of the reset gate r_t amplifies the input state and hidden state at time t-1. The candidate hidden state can be calculated by feeding all of this data into the tanh function in Equation 4. When r_t equals 1, all of the data from the

previous hidden state H_{t-1} is taken into account. Similarly, if r_t is equivalent to 0, the entire data against the prior hidden state is discarded.

$$\acute{h_t} = tanh(WX_t + r_t \cdot Uh_{t-1}) \quad (4)$$

where
h_t – hidden state,
tanh – activation function,
W – weights,
X_t – input,
r_t – reset gate,
U – Update gate,
h_{t-1} – previous state information.

3.3.5. Hidden State

The current hidden state H_t is built using the hidden state. The update gate is opened for the first time at this point. GRU makes use of a single update gate to manage both the historical information, which is denoted by the value H_{t-1}, and the current information, which is derived from the candidate state in Equation 5. If the value of ut is close to zero, then the first term in the equation will vanish, which indicates that the newly revealed hidden state will not have a great deal of information from the previously revealed hidden state. On the other hand, the second portion becomes almost one, which effectively means that the information from the candidate state will be the only part of the hidden state at the present timestamp [23].

$$h_t = U_t \cdot h_{t-1} + (1 - U_t) \cdot \acute{h_t} \quad (5)$$

where
h_t – hidden state,
U_t – Update gate,
h_{t-1} – previous state information.

4. Implementation

The parameter configuration of AE and GRU are depicted below in Table 1 and Table 2 subsequently.

Table 1. AE parameters

Summary	Values
Input Parameters	1(2D array)
Output Parameters	1(2D array)
encoder layers	2
decoder layers	2
Batch size (Records loaded to train)	256
Number of epochs (Forward+Backward)	500
Loss	Binary-cross-entropy

Input: UK Traffic dataset **Output:** Traffic flow
1. Load the UK traffic Dataset 2. Apply Data Pre-process activities 2.1 Remove outliers if exists. 2.2 Transform direction_of_travel column values as E-1, W-2, N-3 and S-4. Count_date column as yyyy-mm-dd. Input Data Preparation: 3. Add StandardScalarization transform technique with a range between -1 and 1. 4. Add reverse StandardScalarization. 5. Autoencoder: 5.1 Add 2 Encoding Layers with 3 Neurons 5.2 Add 2 Decoding Layers with 3 Neurons. NOTE: 3 Neurons because considered only date, direction, all_motor_vehicles. 6. Sort the dataset based on count_date, hour, and direction_of_travel columns. Model Construction: 7. Create AEGRU model: 7.1 Add 50 Hidden layers with neurons as 36. 7.2 Add 3 Dense layers with neurons as 70, 50, and 25, respectively. 8. Add a single flatten layer. 9. Fit or Train the Model with input data for 275 epochs and capture the loss. 11. Predict traffic flow.

Table 2. GRU parameters

Summary	Values
Input Parameters	1 [3D Array]
Output Parameters	1 (2D array)
Hidden layers	50
Neurons in the Hidden Layer	36
Batch size	256
Number of epoch	275
Activation function	Softmax
Loss Function	Mean Absolute Error (MAE)
Dense layers	3
Neurons in Dense layers	50, 25 successively
flatten layers	1

5. Results and Discussion

5.1. Data Description

In this research, the traffic dataset was gathered from Zone 1 to Zone 5 at the Department of Transport in Great Britain, United Kingdom (UK) (Road Traffic Statistics, Great Britain, UK) [24]. The sensors were employed to keep track of how many cars were on major and minor roads in the UK.

Using 8,000 physical observations of the roads, the electronic vehicle monitors record the important information—the experimental evaluation compared to other algorithms used in this dataset. The dataset obtained 21 years (2000-2021) includes 35 variables with spatiotemporal data, such as rain, snowfall, and windy seasons, all of which have a significant impact on traffic flow during the weekdays and were used to assess the efficacy of proposed AEGRU technique.

In this study, the dataset was split into two sections, with the training dataset accounting for 70% and the testing dataset for 30%. Also, there's a column labelled "all motor vehicles" that contains information on all vehicles. This feature adds cars, pedal cycles, taxis, buses, two-wheeled motor vehicles, light goods vans, four-rigid-axle heavy goods vehicles, articulated-axle heavy goods vehicles, and three-rigid-axle heavy goods vans vehicles on weekdays and weekdays off during busy and quiet periods.

5.2. Data Pre-Processing

Furthermore, at the beginning stages of development, the proposed AEGRU model requires data pre-processing to standardize the input data. This is accomplished by the application of the standardschalarization method from the Scikit package. Standardization is a method of scaling that involves centering values on the meanwhile maintaining a unit standard deviation; this indicates that the mean of the attribute remains zero and the distribution produced has a unit standard deviation.

5.3. Validation

Three indices were employed to quantify the effectiveness of the proposed AEGRU model: RMSE exhibited in Equation 6, MAPE exhibited in Equation 7, and MAE exhibited in Equation 8 was evaluated [25]. The formulas are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{m}}\sum_{\mathrm{j}=1}^{\mathrm{n}}\left(\hat{d}_{\mathrm{k}} - \mathrm{d}_{\mathrm{k}}\right)^2} \tag{6}$$

$$\mathrm{MAPE} = \frac{1}{m}\sum_{j=1}^{n}\frac{|\hat{d}_k - d_k|}{d_k} \tag{7}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{j=1}^{n}\left|\hat{d}_k - d_k\right| \tag{8}$$

Where
$\hat{d}_{\mathrm{k}}$ signifies the predicted value,
d_{k} signifies the observed value and
n signifies the number of data elements.

5.4. Results Analysis

To test the efficacy of the proposed AEGRU technique, contrast the five approaches: ARIMA, naive bayes, linear regression, recurrent neural network (RNN), and LSTM using the UK traffic dataset. The error rates achieved by training and testing models are summarized below in Table 3.

Since the UK traffic dataset was divided into training and testing data sets with a ratio of 70%:30%, the proposed AEGRU model was trained on the training dataset and tested on the testing dataset. As the number of epochs (training loops) increased, the loss rate in the testing data set went from 60% to 5%, and in the training data set, it went from 52% to 5%. Four convergences between training loss and testing loss occurred at epochs 97, 146, 220, and 270, with loss rates of 47%, 33%, 29%, and 8%, respectively. Eventually, the loss during testing reached 6%, whereas the loss during training reached 4%. This occurred before significant divergence commenced. A loss of around 6% is predicted by the proposed AEGRU model, shown in below Figure 3.

Furthermore, the comparison of predicted traffic to actual traffic for the following twenty-four hours of the day is depicted below in Figure 4. In this context, "predicted traffic" refers to the prediction made by the proposed model. There is a significant increase in traffic volume during the morning peak hour, which begins at 6:00 a.m. and ends at 12:00 p.m. as well as during the evening peak hour, which runs from 15:00 p.m. to 23:05 p.m. During off-peak hours, the flow of traffic is relatively normal. Therefore, the prediction made by the proposed AEGRU model is quite close to the actual flow of traffic, except for a few hours; the expected traffic is somewhat higher and lower than real traffic, with the difference being essentially insignifican

Table 3. Error metric values

	Training			**Testing**		
Algorithm	RMSE	MAPE	MAE	RMSE	MAPE	MAE
ARIMA	37.1	32.5	29.4	38.9	36.71	39.8
Naive Bayes	29.7	28.5	27.1	35.7	33.51	32.54
Linear Regression	13.4	11.7	10.3	16.35	13.47	14.01
RNN	9.3	8.6	7.67	12.65	13.9	11.74
LSTM	6.2	8.5	7.9	6.14	5.86	4.6
AE-GRU✓	1.6	2.3	1.9	1.87	2.15	3.21

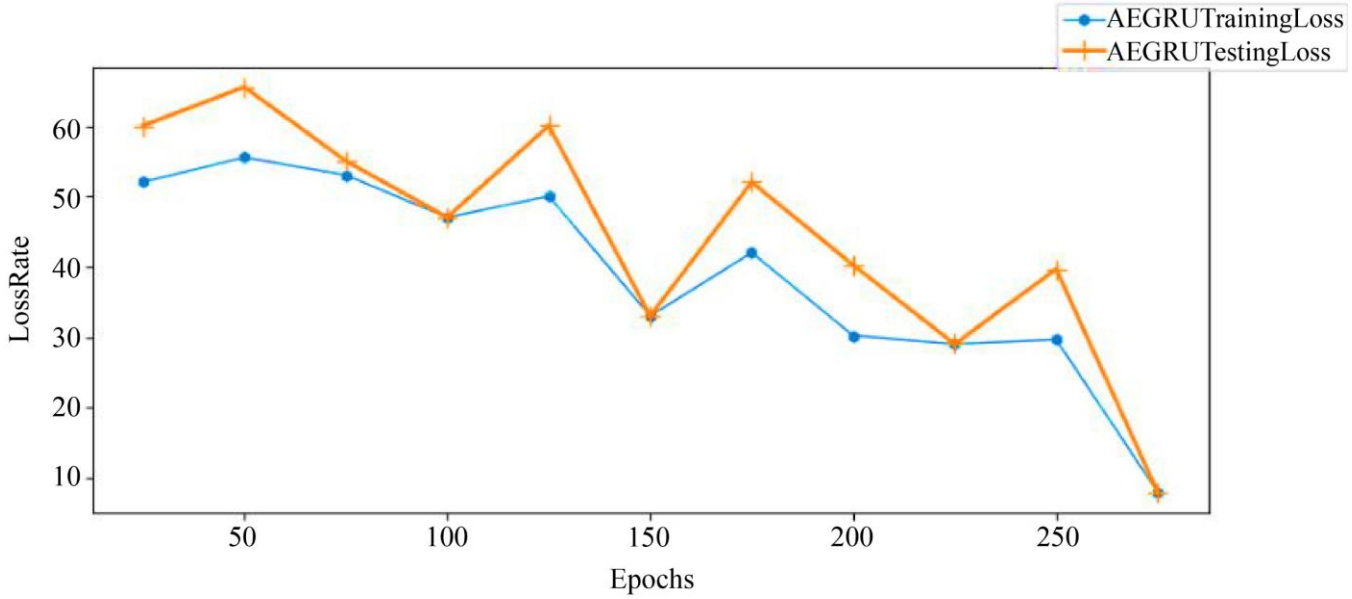

Fig. 3 Loss of proposed algorithm on UK Traffic data per epoch

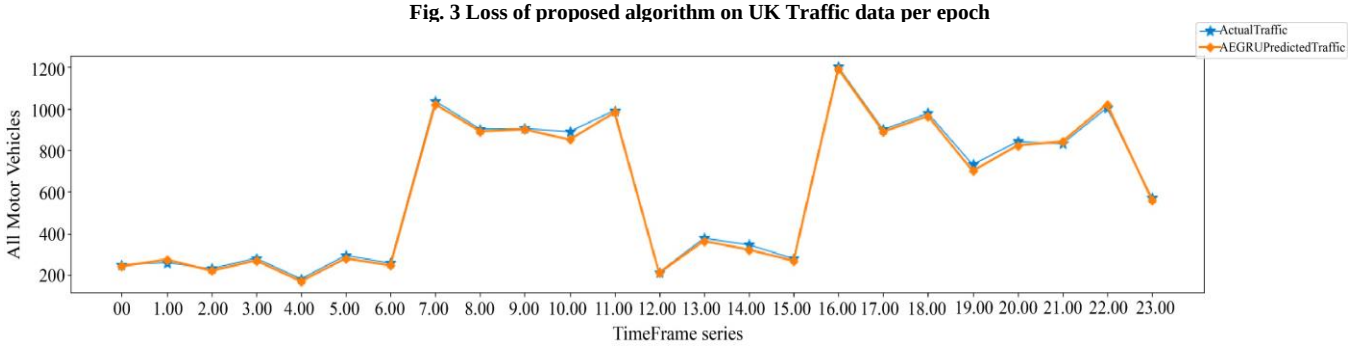

Fig. 4 Proposed algorithm traffic flow prediction

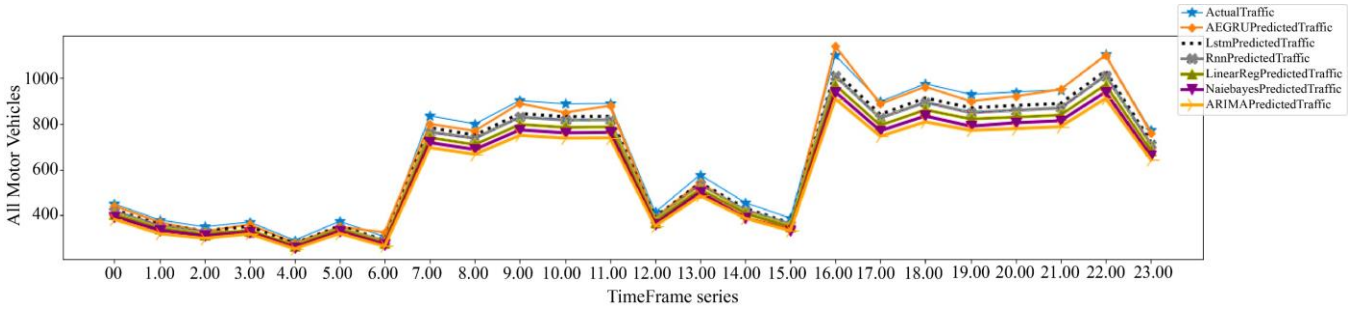

Fig. 5 All algorithms traffic flow prediction evaluation

In order to predict the traffic flow that is expected to occur in the future with the results of using various models, including the ARIMA model, the naive bayes model, the linear regression model, the RNN model, and LSTM model, which are depicted in above Figure 5, compared to other algorithms, where error rates are far higher than the proposed model's, it is evident that the proposed AEGRU forecast is consistent and, in most cases, aligns with the actual traffic. Therefore, the proposed AEGRU model is more successful at predicting traffic patterns than alternative methods.

Moreover, during training of the proposed hybrid model, AEGRU attained error values of 1.6%, 2.3%, and 1.9% for RMSE, MAPE, and MAE. In comparison, the LSTM error value is 4.2%, and the error values of other techniques are above 5%. Similarly, the proposed method has a 1.9% RMSE error value, whereas all peer-comparison methods have greater than 6% error values. As a result, this study concluded that the proposed AEGRU model had a lower error rate in training, as demonstrated in below Figure 6.

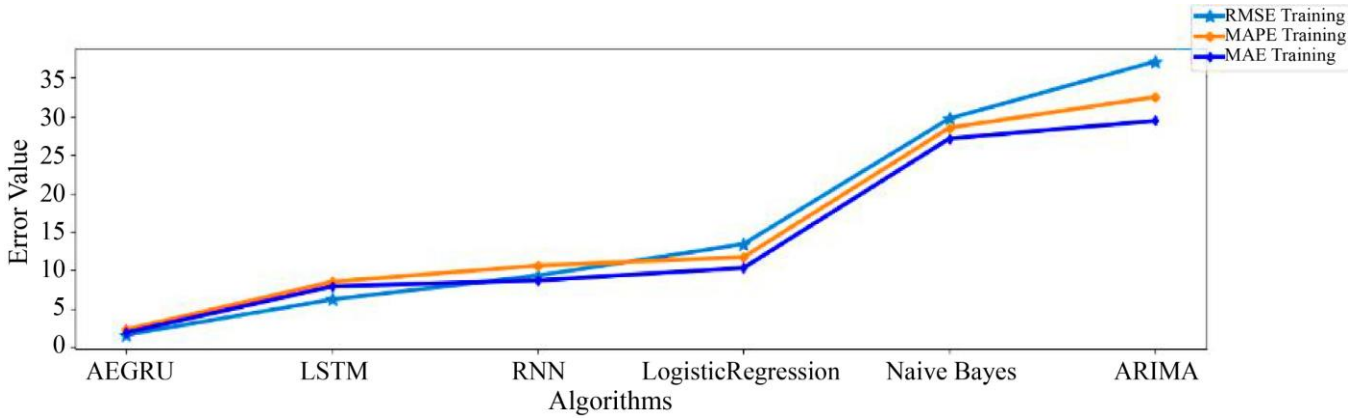

Fig. 6 Training RMSE, MAPE, MAE on UK traffic data

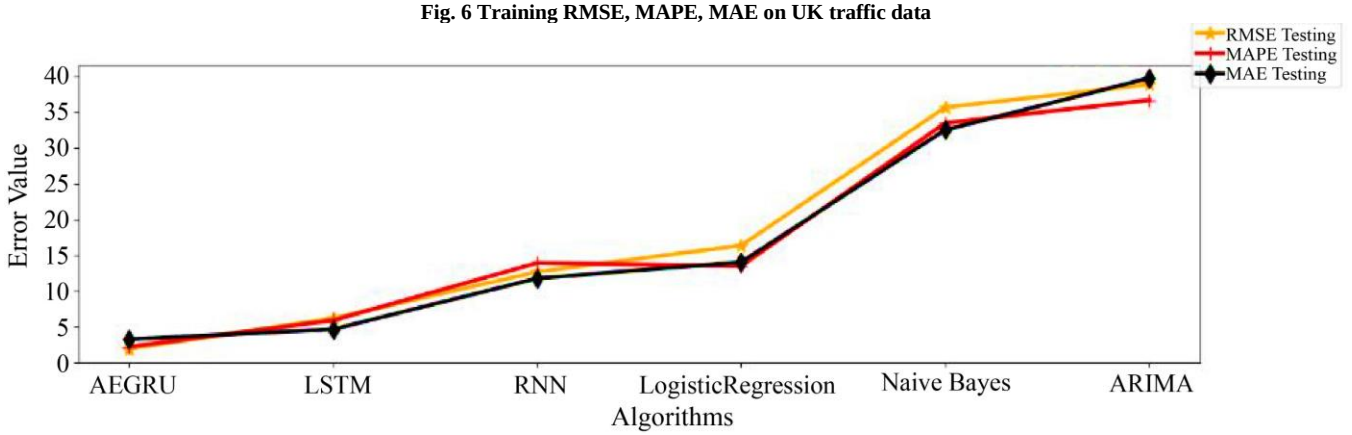

Fig. 7 Testing RMSE, MAPE, MAE on UK traffic data

Furthermore, the testing of the proposed hybrid model AERU is insignificant for RMSE, MAPE, and MAE error values of 1.87%, 2.15%, and 3.21%, respectively. In contrast, it is greater than 4% for other methods. Also, in testing, it was determined that the proposed AEGRU method performed more effectively, with a lower overall error rate, as shown in above Figure 7.

The benefits of the proposed AEGRU algorithm are listed below.

- The proposed model best fits linear data because the proposed LTTF model has RMSE, MAPE, and MAE ratios of less than 4% in training and testing data sets. It also outperforms other algorithms in terms of training speed, fewer parameters (less memory consumption), weight sharing, and error rates.
- LTTF prediction is more accurate and closer to the actual prediction.

6. Conclusion

This research presents the AEGRU model, a hybrid of AE and GRU, for LTTF prediction. AE will take the raw data and pick out the most important features before doing dimensionality reduction. AE will then prepare well-defined and intuitive input data for GRU so that it can work faster. GRU uses the information given by the AE to make predictions about how much traffic volume there will be in the future. The research methodology focuses mostly on the LTTF prediction for the upcoming twenty-four hours of a day, making it easier for commuters to plan their itineraries. Finally, the effectiveness of the AEGRU forecast was evaluated by employing actual data sets from the United Kingdom's road traffic statistics. The outcomes of the following metrics reveal that the proposed methodology performed better than other methods being used, including ARIMA, Naive Bayes, Linear Regression, RNN, and LSTM models. When compared to alternative methods, long-term traffic flow prediction endure minor error rates and comes quite close to the actual value. Additionally, the execution time is shortened.

- Lesser error rates for RMSE, MAPE, and MAE in testing data, with values as 1.87%, 2.15%, and 3.21%, respectively.
- Lesser error rates for RMSE, MAPE, and MAE in training data, with values as 1.6%, 2.3%, and 1.9%, respectively.

In future work, to make the long-term traffic flow prediction more stable, the proposed AEGRU algorithm ensemble with automatic feature selection technique enhances the performance.

References

[1] Yuhan Jia, Jianping Wu, and Ming Xu, "Traffic Flow Prediction with Rainfall Impact Using a Deep Learning Method," *Journal of Advanced Transportation*, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Junping Zhang et al., "Data-Driven Intelligent Transportation Systems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624-1639, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[3] Fei Su et al., "Long-Term Forecasting Oriented to Urban Expressway Traffic Situation," *Advances in Mechanical Engineering*, vol. 8, no. 1, pp. 1-16, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiaobo Chen et al., "Ensemble Learning Multiple LSSVR with Improved Harmony Search Algorithm for Short-Term Traffic Flow Forecasting," *IEEE Access*, vol. 6, pp. 9347-9357, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Eleni I. Vlahogianni et al., "Short-Term Traffic Forecasting: Where We are and Where We're Going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3-19, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yiqun Li et al., "A Hybrid Deep Learning Framework for Long-Term Traffic Flow Prediction," *IEEE Access*, vol. 9, pp. 11264-11271, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Zhumei Wang, Xing Su, and Zhiming Ding, "Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561-6571, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Senzhang Wang, Jiannong Cao, and Philip S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jürgen Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural networks*, vol. 61, pp. 85-117, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mascha Van Der Voort, Mark Dougherty, and Susan Watson, "Combining Kohonen Maps with ARIMA Time Series Models to Forecast Traffic Flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307-318, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[11] Kit Yan Chan et al., "Neural-Network-Based Models for Short-Term Traffic Flow Forecasting Using a Hybrid Exponential Smoothing and Levenberg–Marquardt Algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644-654, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[12] Selvaraj Vasantha Kumar, "Traffic Flow Prediction Using Kalman Filtering Technique," *Procedia Engineering*, vol. 187, pp. 582-587, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zheng Zhao et al., "LSTM Network: A Deep Learning Approach for Short-Term Traffic Forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68-75, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Yi Hou, Praveen Edara, and Yohan Chang, "Road Network State Estimation Using Random Forest Ensemble Learning," *2017 IEEE 20th International Conference on Intelligent Transportation Systems*, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Zhang Mingheng et al., "Accurate Multisteps Traffic Flow Prediction Based on SVM," *Mathematical Problems in Engineering*, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[16] Shiliang Sun, Changshui Zhang, and Guoqiang Yu, "A Bayesian Network Approach to Traffic Flow Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp.124-132, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[17] Bin Sun et al., "Short-Term Traffic Forecasting Using Self-Adjusting K-Nearest Neighbours," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 41-48, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[18] Deva Priya Isravel, Salaja Silas, and Elijah Blessing Rajsingh, "Long-Term Traffic Flow Prediction Using Multivariate SSA Forecasting in SDN Based Networks," *Pervasive and Mobile Computing*, vol. 83, p. 101590, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hao Peng et al., "Dynamic Graph Convolutional Network for Long-Term Traffic Flow Prediction with Reinforcement Learning," *Information Sciences*, vol. 578, pp. 401-416, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Asma Belhadi et al., "A Recurrent Neural Network for Urban Long-Term Traffic Flow Forecasting," *Applied Intelligence*, vol. 50, no. 10, pp. 3252-3265, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Qu Licheng et al., "Daily Long-Term Traffic Flow Forecasting Based on a Deep Neural Network," *Expert Systems with Applications*, vol. 121, pp. 304-312, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Autoencoders in Deep Learning. [Online]. Availability: https://www.v7labs.com/blog/autoencoders-guide/

[23] Introduction to Gated Recurrent Unit (GRU). [Online]. Availability: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

[24] Road Traffic Statistics, Great Britain, UK. [Online]. Available: https://roadtraffic.dft.gov.uk/downloads

[25] Alexei Botchkarev, "Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology," *ArXiv preprint arXiv: 1809.03006, 2018*. [CrossRef] [Google Scholar] [Publisher Link]