Kentsel Trafik Tahminine Yönelik Derin Öğrenme Tabanlı Verimli Bir Hibrit Model

Araştırma Makalesi/Research Article

Anıl UTKU

Computer Engineering Department, Munzur University, Tunceli, Turkey
anilutku@munzur.edu.tr

(Geliş/Received:26.08.2022; Kabul/Accepted:16.03.2023)
DOI: 10.17671/gazibtd.1167140

***Özet*—** Trafik yoğunluğu problemi, kentsel hayatın en önemli sorunlarından biri haline gelmiştir. Trafik yoğunluğu sebebiyle harcanan zaman ve yakıt, araç kullanıcıları ve ülkeler için önemli bir kayıptır. Trafikte geçen zamanı azaltmak amacı ile geliştirilen uygulamalar, uzun vadeli trafik yoğunluğu hakkında başarılı tahminlerde bulunamamaktadır. Kameralar, sensörler ve mobil cihazlar üzerinden elde edilen trafik verileri, trafik yönetimi sorununu çözebilmek amacıyla yapay zekâ teknolojilerinin kullanımını ön plana çıkarmaktadır. Bu çalışmada, trafik yoğunluk tahminine yönelik Convolutional Neural Network (CNN) ve Recurrent Neural Network (RNN) modelleri kullanılarak hibrit bir tahmin modeli geliştirilmiştir. Çalışmada, CNN ve RNN'in öne çıkan özelliklerinden faydalanmak amaçlanmıştır. CNN, özellik çıkarma aşamasında, RNN ise sıralı zaman serisi verileri üzerinde öğrenme ve tahmin için etkili bir modeldir. Bu yöntemler hibrit bir şekilde kullanılarak tahmin doğruluğunun arttırılması amaçlanmıştır. İstanbul Büyükşehir Belediyesi tarafından sunulan saatlik trafik yoğunluğu veri seti kullanılmıştır. Kullanılan veriseti 2321 farklı nokta için 2020 Ocak ile 2020 Aralık tarihleri arasındaki trafik yoğunluk bilgisini içermektedir. Geçen araç sayısı, Bağcılar Avrupa Otoyolu kavşağında daha yüksek olduğu için bu konum deneysel çalışmalarda kullanılmıştır. Seçilen konum için 9379 satır araç bilgisi bulunmaktadır. Geliştirilen hibrit model Linear Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Multilayer Perceptron (MLP), CNN, RNN ve Long-Short Term Memory (LSTM) ile İstanbul'un 2020 yılına ait trafik verileri kullanılarak test edilmiştir. Deneysel sonuçlar, önerilen hibrit modelin karşılaştırılan modellere göre daha başarılı sonuçlara sahip olduğunu göstermiştir. Önerilen model kavşaktan geçen araç sayısı tahmininde 0,929 R2 değerine, kavşaktan geçen araçların ortalama hızlarının tahmininde ise 0,934 R-Squared (R2) değerine sahip olmuştur.

***Anahtar Kelimeler* —** trafik tahmini, makine öğrenmesi, derin öğrenme, CNN, RNN

Deep Learning Based an Efficient Hybrid Model for Urban Traffic Prediction

***Abstract*—** The traffic density problem has become one of the most important problems of urban life. The time and fuel spent due to traffic density is a significant loss for vehicle users and countries. However, applications developed to reduce the time spent in traffic cannot make successful predictions about long-term traffic density. Traffic data obtained from cameras, sensors and mobile devices highlight the use of artificial intelligence technologies to solve the traffic management problem. In this study, a hybrid prediction model was developed using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models for traffic density prediction. It was aimed to benefit from the prominent features of CNN and RNN. CNN is an effective model in the feature extraction phase, while RNN is an effective model in the learning and prediction phase on sequential time series data. It was aimed to increase the prediction accuracy by using these methods in a hybrid way. The hourly traffic density dataset provided by the Istanbul Metropolitan Municipality was used. The dataset used includes traffic density information for 2321 different points between January 2020 and December 2020. Since the number of passing vehicles is higher at the Bağcılar European Motorway junction, this location was used in the experimental studies. There are 9379 lines of vehicle information for the selected location. The developed hybrid model was tested using Linear Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Multilayer Perceptron (MLP), CNN, RNN and Long-Short Term Memory (LSTM) with 2020 Istanbul's traffic data. Experimental results showed that the proposed hybrid model was more successful results than the compared models. The proposed model has 0.929 R-Squared (R2) in the predicting the number of vehicles passing through the junction, and 0.934 R2 in the predicting the average speed of the vehicles passing through the junction.

***Keywords*—** traffic prediction, machine learning, deep learning, CNN, RNN

1. INTRODUCTION

Today, with the increase in urbanization, the growth in the urban population causes an increase in traffic density [1]. Traffic density not only causes loss of time, but also causes loss of labour, income, health and social life, and results in environmental and noise pollution [2]. Therefore, meeting the need for mobility, especially in big cities, eliminating traffic-related problems, or keeping them under control is only possible with the development of intelligent systems.

The traffic density, which has become the biggest problem of Istanbul, is increasing daily. Especially due to the pandemic, people's use of their own vehicles instead of public transportation has increased traffic density [3]. As a result, there is a severe density at almost every point of Istanbul traffic at the beginning and end of working hours.

Traffic congestion in Istanbul increases considerably on weekdays, especially during commuting and commuting hours [4]. City governments use sensors and traffic cameras at critical junctions to collect mobility data. However, due to high deployment and maintenance costs, the applicability of these technologies across the city is limited. With the development of intelligent transportation systems, vast amounts of mobility data can be collected from mobile devices [5]. These obtained data make large-scale and reliable traffic predictions applicable. Increasing traffic data provides potential new perspectives for solving the traffic management problem. Traffic prediction has become an important research area in recent years. In the literature, studies use artificial intelligence methods for traffic density prediction.

Du et al. developed a hybrid model using CNN and LSTM for short-term traffic forecasting [6]. With the developed model, it was aimed to learn the temporal and spatial properties. By using LSTM, temporal dependencies and local trends were extracted. Experimental results showed that the developed model was more successful than classical deep learning models.

Mohammed and Kianfar presented a comparative analysis of Deep Neural Networks (DNN), RF, Gradient Boosting and Generalized Linear Model (GLM) for traffic density and traffic rate prediction [7]. In the study, traffic data from Interstate 64 in St. Louis Missouri is used. Experimental results have shown that RF gives better results than the compared models.

Zhang et al. developed a CNN-based short-term traffic prediction model [8]. Selected features are forwarded to CNN for prediction. The developed model is compared with SVM, Seasonal Autoregressive Integrated Moving Average (SARIMA), k-Nearest Neighbours (kNN), Artificial Neural Network (ANN) and CNN. Experimental results showed that the developed model was more successful than other models.

Gu et al. proposed the Fusion Deep Learning (FDL) model to predict lane-based traffic speed [9]. A dual-layer deep learning model was created using LSTM and Gated Recurrent Unit (GRU). The developed model was tested using Beijing traffic data with ARIMA, Lighthill-Whitham-Richards (LWR), Multi-Layer Perceptron (MLP), Kalman Filter (KF), Radial Basis Function Neural Network (RBFNN), CNN and LSTM. Experimental results showed that dual-layer deep learning model was more successful than the compared models.

Wang et al. proposed an LSTM-based model for traffic speed estimation [10]. In the study, it is aimed to determine the traffic flow mechanism by dividing the road network into critical points. Dividing roads are modelled using LSTM. Spatial-temporal features are used in traffic forecasting. Experimental studies using ANN, kNN and CNN showed that the proposed model is more successful.

Taş and Müngen developed an ANN based model for the prediction of regional traffic density [11]. In the study, it is aimed to increase the success of the forecast model by using environmental factors such as weather conditions. Experimental results showed that the developed model has an R-Squared (R^2) value of 0.88.

Takak et al. visualized traffic density and predicted traffic speed in the short, medium and long term [12]. Traffic speed data obtained from sensors at many points in Istanbul are visualized according to time and location. Experimental studies using Autoregressive Integrated Moving Average (ARIMA) and regression models showed that the regression model predicts short-term traffic flow with 23.99% Mean Absolute Percentage Error (MAPE) and 8.5 MAE.

Essien et al. propose a deep learning-based traffic forecasting model that combines information extracted from tweets with traffic and weather information [13]. The developed LSTM based model was tested using the Manchester traffic dataset. The developed model is compared with SVM, eXtreme Gradient Boosting (xGBoost) and RF. Experimental results showed that the developed model was more successful than the compared models.

Wang et al. developed a hybrid model based on CNN-LSTM for short-term traffic flow prediction [14]. CNN was used to extract temporal and spatial features from the data. It was aimed to learn the long-term dependencies by presenting the features extracted by CNN as input to LSTM. Experimental results showed that the developed model was more successful than the classical neural network models.

Zheng et al. developed a hybrid deep learning model using CNN and LSTM modules to extract temporal and spatial properties of traffic flow data [15]. An attentional mechanism was developed by assigning different weights to recipe flow data at different times. It was aimed to extract the daily and weekly characteristics of traffic flow data using the Bi-LSTM model. Experimental results showed that the developed model was more successful than the other models compared.

The studies in the literature generally use iterative neural network models for time-dependent data such as traffic flow data. Iterative neural network models are helpful because they can extract time dependencies from time series data. In the studies examined, it was seen that hybrid models were developed using CNN and LSTM models. In this study, a hybrid model was developed using CNN and RNN models, similar to the studies in the literature. The developed model consists of a 1D CNN layer, similar to the models used in the studies in the literature. It was aimed to extract temporal features from the data using 1D CNN. RNN was used to learn the features extracted by CNN and to increase the prediction accuracy.

The main contributions of this study to the literature can be summarized as follows:
- This is the first study using this dataset for traffic density prediction.
- Deep learning-based hybrid prediction model was proposed using CNN and RNN models.
- The proposed model was comparatively analysed with popular machine learning and deep learning models such as LR, RF, SVM, MLP, CNN, RNN and LSTM.

2. MATERIAL AND METHOD

In this study, a hybrid deep learning model was proposed for Istanbul by using CNN and RNN models to predict average traffic speed and vehicle density. The proposed model aims to predict the average speed and traffic density at the Bağcılar European Motorway junction, one of Istanbul's critical junction points. The proposed hybrid prediction model was compared with LR, RF, SVM, MLP, CNN, RNN and LSTM.

2.1. Dataset

This study used the hourly traffic density dataset between January 2020 and December 2020 presented by Istanbul Metropolitan Municipality [16]. The dataset consists of date time, longitude, latitude, geohash, minimum speed, maximum speed, average speed and number of vehicle attributes. Date time represents time in Year/Month/Day and time format. Longitude represents the longitude value and Latitude represents the latitude value. Geohash refers to the geolocation code obtained according to the latitude and longitude values. Minimum, maximum, and average speed attributes represent the minimum, maximum, and average speed values, respectively. Finally, the number of vehicles represents the hourly number of vehicles passing a specific location. The dataset used is shown in Table 1.

Table 1. Hourly traffic density dataset

Date time	Longitude	Latitude	Geohash	Minimum speed	Maximum speed	Average speed	Number of vehicle
2020-11-30 23:00:00	29.2950439453125	41.1026000976562	sxkcfd	41	137	80	130
2020-11-30 23:00:00	29.0863037109375	41.0092163085938	sxk9mc	6	122	49	77
2020-11-30 23:00:00	28.0975341796875	41.1904907226562	sx7fr6	51	129	86	65
2020-11-30 23:00:00	29.0753173828125	41.0256958007812	sxk9ms	9	89	56	29
2020-11-30 23:00:00	29.1412353515625	40.9927368164062	sxk9pn	14	124	75	36

The dataset contains traffic density information for 2321 different points. The locations with the highest number of vehicles were determined to select where the traffic density and speed prediction would be made. As seen in Figure 1, Bağcılar European Motorway junction, one of the points with the highest number of vehicle passes, was chosen for experimental studies.

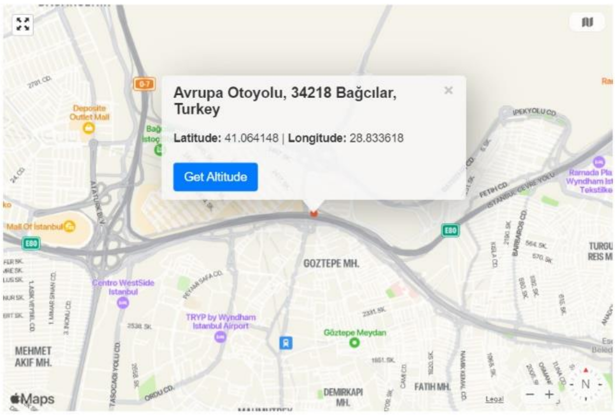

Figure 1. Bağcılar European Motorway junction

Figure 2 shows the number of vehicles passing through the Bağcılar European Motorway junction per hour.

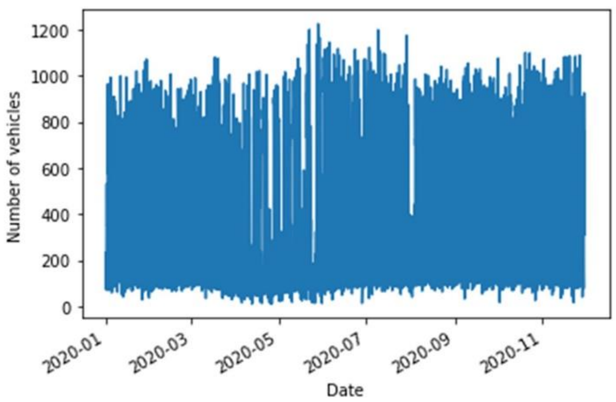

Figure 2. Number of vehicles per hour

Figure 3 shows the average speed of vehicles passing through the Bağcılar European Motorway junction per hour.

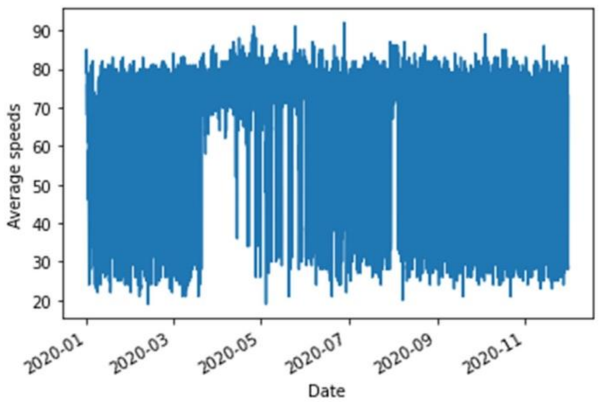

Figure 3. Average speeds of passing vehicles per hour

2.2. Baseline Models

This section explains LR, RF, SVM, MLP, CNN, RNN and LSTM models used in the study.

LR is a method used for linear and continuous variables [17]. When a linear relationship is observed between the variables, LR is used to predict the future and examine how the variables affect each other [18]. LR creates a straight line that minimizes discrepancies between predicted and actual values [19]. Then, the value of the dependent variable is predicted using the independent variable.

RF is used to make accurate predictions by producing more compatible models using multiple decision trees [20]. RF trains decision trees on different datasets and combines the prediction results. Then, it votes on the prediction results and uses the prediction with the highest votes [21].

SVM aims to find the hyperplane in the feature space that can optimally separate the two classes from each other. It is a supervised learning and classification method. The distance between the positive and negative samples closest to the hyperplane is called the margin [22]. SVM tries to find support vectors that make this distance furthest. Finding the maximum margin for linearly separable data is easier. However, while classifying the nonlinearly separated data, the data are transferred to a different space where they can be linearly separated and classified in this new space [23].

MLP has a structure in which many neurons with non-linear activation functions are hierarchically connected. MLP consists of one or more layers. The input layer receives the incoming data and sends it to the middle layer. Incoming information is transferred to the next layer. The number of intermediate layers varies according to the problem, at least one, and is adjusted according to the need. The output of each layer becomes the input of the next layer. Thus, the output is reached. Neurons are connected to all neurons in the next layer. The output layer processes data from previous layers and determines the network's output. The output number of the system is equal to the number of elements in the output layer [24].

CNN comprises multiple layers and connects the input data to the output data. It generally performs operations such as scanning objects, clustering objects, and finding similar ones over a picture or a video. In order to activate this network, various parameters and libraries that contribute to the formation of neural networks are used [25]. After the input data is created, filtering is carried out by various layers. Filters are expressed with matrix calculation, and a single matrix is obtained while generating the output data. Data are analysed by creating 5 different layers. A convolutional layer is created to detect the features on the image. Non-linearity is introduced to the system with the Non-Linearity Layer. Thanks to the Pooling Layer, the weight in the image is tried to be reduced, and the compatibility of the image with the parameters is tested in line with this reduced weight. Flattening layer is the layer where the data required to create the output data is prepared. Finally, Fully-Connected Layer is used for classification.

RNN is a class of neural networks in which node connections form a directed loop [26]. RNN has an input layer, hidden layers, and an output layer. All of these layers work independently [27]. Structures in each layer have weights and layer-specific thresholds. As a result of these recurrent steps, the previous input state is stored and combined with the newly obtained input value so that the relationship of the newly obtained input with the previous input is provided [28].

LSTM is a model that can learn long-term transactions. LSTM is an advanced variant of RNN used for modeling sequential data [29]. LSTM consists of forget it, input and output layers. The forget layer decides whether the incoming information will be forgotten or not. The input layer decides which information is stored in memory or not. Finally, the output layer decides whether or not information will be output [30].

2.3. Data Pre-processing

In this study, data pre-processing was performed on the dataset before the models were applied. Training, testing and validation data were selected. Using the GridSearchCV library, the parameters of the models were optimized on the validation data. Models were created by selecting the parameters with the lowest MSE values. Then the number of vehicles and the average speed of the vehicles were predicted. MSE, RMSE and R^2 values were calculated according to the estimation results obtained. The flowchart of the developed system is presented in Figure 3.

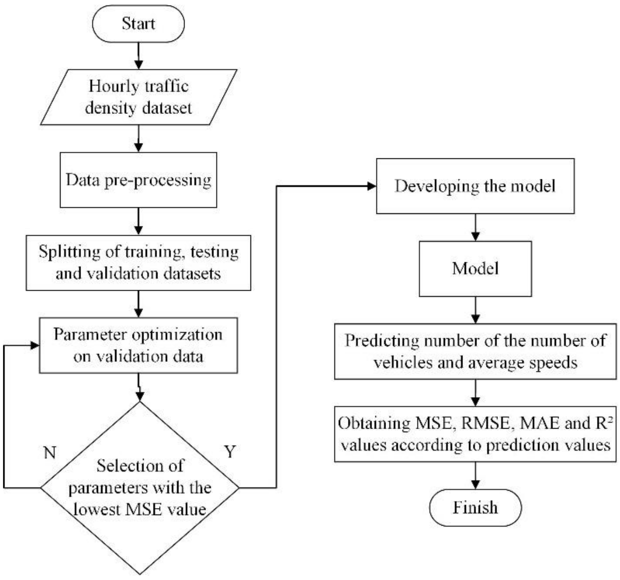

Figure 4. The flowchart of the developed system

The data used in this study is a time series dataset consisting of hourly indexed data points. In order to apply machine learning methods to time series data, these data should be structured as supervised learning problems using the sliding window method. In the sliding window method, historical observation data equal to the specified window size is placed as input into the sliding window. The value in the next time step to be predicted will be the output of the sliding window. In this study, the sliding window size was chosen as 3 according to the experimental studies. As seen in Figure 5, time series data is structured as a supervised learning problem, with the data in time steps t_1, t_2, and t_3 as the input of the sliding window, and the data in time step t_4 as the output of the sliding window.

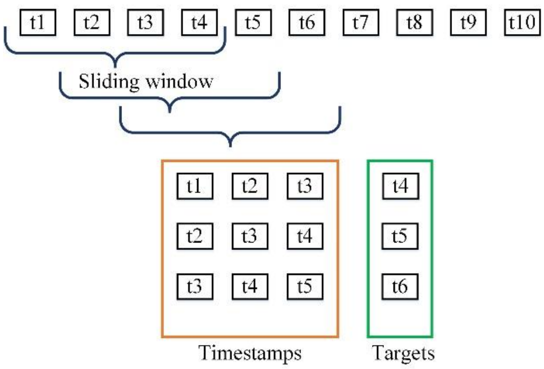

Figure 5. Sliding window method

After structuring the data as a supervised learning problem, the data were normalized in the 0-1 using MinMaxScaler. After the normalization, the data was split into training, test, and validation sets. These values were chosen for experimental studies, as the highest prediction accuracy was achieved in the combination of 80% training and 20% testing. In addition, 10% of the training data is split for validation. The validation data were used to optimize the parameters of the applied models. Model parameters were optimized using GridSearchCV from the Scikit Learn library so that the applied models could obtain the best prediction results.

2.4. Performance Evaluation Metrics

Mean Squared Error (MSE), Root Mean Squared Error (RMSE), MAE and R^2 metrics are mainly used to determine the error rate between the predictions and the actual values. MSE calculates the mean of the squares of the difference between actual and predicted values. MSE is calculated using Eq.1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2 \tag{1}$$

Here, y is the actual observation values, $\hat{y}$ is the predicted values, and n is the total data.

RMSE calculates the square root of the mean of the squares of the difference between the actual observation values and the predicted values. RMSE is calculated as seen in Eq.2.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2} \tag{2}$$

MAE refers to the mean of the absolute differences between the actual observation values and the predicted values. MAE is calculated using Eq.3.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}| \tag{3}$$

R^2 indicates how well the dataset fits the regression line. It describes the goodness of fit of the data point on the regression line. It is the square value of the correlation coefficient. R^2 is calculated using Eq.4.

$$R^2 = \frac{\sum(y-\hat{y})^2}{(y-\overline{y})^2} \tag{4}$$

Here, $\hat{y}$ is the predicted y values and $\overline{y}$ is the mean of the y values.

2.5. Proposed Hybrid Prediction Model

In this study, a hybrid deep learning model was developed using CNN and RNN models. The architecture of the proposed model is shown in Figure 6.

CNN is an efficient model for automatically extracting features and learning from one-dimensional series data such as univariate time series. In this study, a hybrid model was proposed using CNN to interpret sub-sequences that are input to RNN. CNN extracts features from the input data and transforms the univariate input data into multi-dimensional groups using convolution. The multi-dimensional datasets are then transmitted to the RNN for prediction.

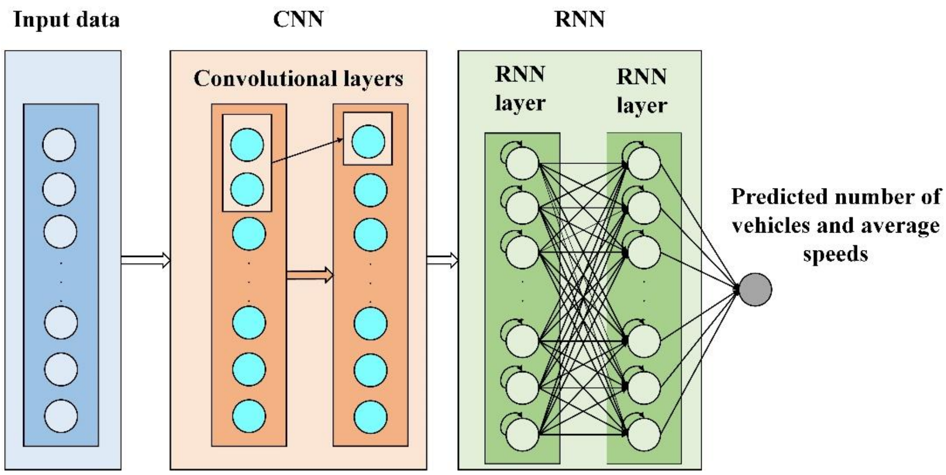

Figure 6. The architecture of the proposed model

CNN is an efficient model for automatically extracting features and learning from one-dimensional series data such as univariate time series. This study proposes a hybrid model using CNN to interpret sub-sequences that are input to RNN. CNN extracts features from the input data and transforms the univariate input data into multi-dimensional groups using convolution.

RNN contains feedback loops and encodes contextual information of a temporal sequence. For an input sequence given as $\{F_1, F_2, ..., F_T\}$, the hidden states h_t and outputs y_t can be calculated as follows:

$$h_t = H(W_{ih}F_t + W_{hh}h_{t-1}) + b_h \quad (1)$$

$$yt = W_{ho}h_t + b_o \quad (2)$$

Where, W_{ih}, W_{hh}, W_{ho} are weight matrices between input, hidden and output layers. Basically, for a x_t input received at time *t*, long-term memory C_{t-1} and transaction memory h_{t-1} are updated from the previous time step to time *t*. The developed model uses following equations for learning and prediction.

$$i_t = \sigma\left(w_{xi}\, x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i\right) \quad (3)$$

$$f_t = \sigma\left(w_{xf}\, x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + bf\right) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma\left(w_{xo}\, x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o\right) \quad (6)$$

$$h_t = o_t \odot h(c_t) \quad (7)$$

where, i_t represents input gate, o_t represents output gate, f_t represents forgetting gate, *c* represents the cell activation vector, *w* weight matrix and *b* represents the bias vector. Ѳ represents the scalar product of two vectors. x_t and h_t represent input-output sequences. g_t refers to the input string.

The CNN-RNN architecture was developed to support the prediction model. The developed model contains CNN layers combined with RNN to extract properties from input data. First, using convolution, CNN extracts essential information from the input data and converts the univariate input data into multi-dimensional groups. Then, after the input data is sent to RNN, it is passed to the prediction stage.

In the developed model, CNN is used for feature extraction, while RNN is used to analyze and predict features extracted by CNN. The first step is to divide the input sequences into sub-sequences that can be processed by the CNN so that the developed hybrid model can be used in the weather prediction problem. Univariate time series data can be divided into 3 inputs and one output as input/output samples.

CNN interprets these sub-array samples and sends them to RNN for processing as input. Here, CNN has a one-dimensional convolutional layer with kernel size 1 and filter number 64 for reading sub-arrays. The number of filters refers to the number of times the input string is read.

Following the convolutional layer, a max pooling layer that is used to interpret the input feature and the dense layer that interprets the properties extracted by the convolution layer of the model are defined. The convolution and pooling layers are 3D layers. For this reason, flatten layer is used to reduce the feature maps to a one-dimensional vector to be used as an input to RNN.
Parameter analysis studies aim to reach the highest prediction accuracy with parameters such as the number of layers, neurons, epochs, and batch size. Adam was used as the optimizer. The GridSearchCV library was used to optimize the parameters of the developed model. The parameters of CNN and RNN were determined using GridSearchCV, and thus the model was created. The parameters of RF are max depth: 9, max features: log2, max leaf nodes: 9, number of estimators: 25, respectively. SVM's parameters are C: 10, gamma: 1e-07, epsilon: 0.1 and kernel: linear, respectively.

3. EXPERIMENTAL RESULTS

In this study, the proposed hybrid model was extensively compared with LR, RF, SVM, MLP, CNN, RNN and LSTM. The results obtained according to MSE, RMSE, MAE and R^2 metrics were analysed comparatively. Table 2 and Figure 7 show comparative experimental results for predicting the number of vehicles passing through the junction.

Table 2. Experimental results for predicting the number of vehicles

Models	MSE	RMSE	MAE	R^2
LR	18768.258	136.997	85.756	0.772
RF	12222.608	110.556	76.289	0.851
SVM	17585.555	132.610	86.439	0.786
MLP	11067.972	105.204	72.410	0.865
CNN	11555.746	107.498	72.582	0.859
RNN	10544.777	102.687	68.449	0.872
LSTM	9830.563	99.149	68.167	0.880
Proposed model	5302.627	72.819	54.036	0.929

Experimental results for the predicting the number of vehicles passing through the junction showed that the proposed model was more successful than the compared models. After the proposed model, LSTM, RNN MLP, CNN, RF, SVM and LR were successful, respectively. Table 3 and Figure 8 show comparative experimental results for predicting the average speed of vehicles passing through the junction.

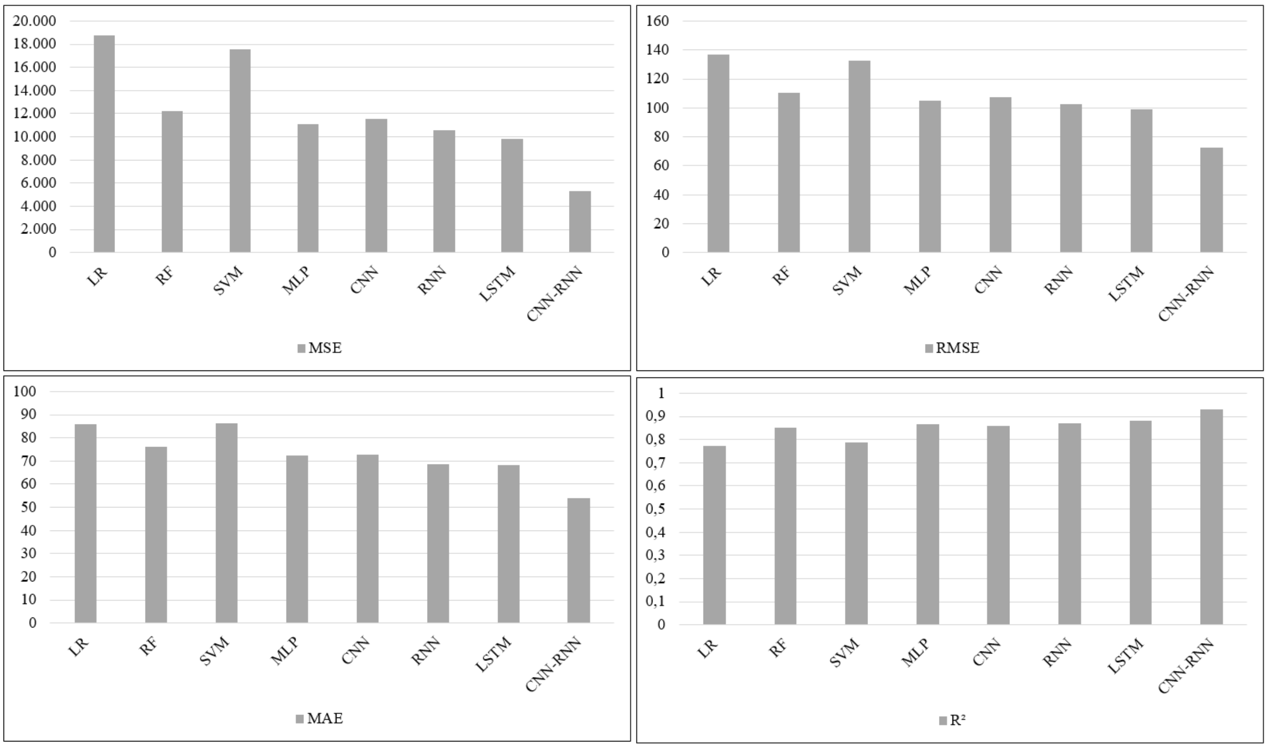

Figure 7. Experimental results for predicting the number of vehicles

Table 3. Experimental results for predicting average speeds of vehicles

Models	MSE	RMSE	MAE	R^2
LR	71.702	8.467	5.334	0.805
RF	68.623	8.283	5.461	0.814
SVM	69.239	8.321	5.389	0.812
MLP	64.974	8.061	5.241	0.824
CNN	65.773	8.110	5.592	0.821
RNN	64.197	8.012	5.306	0.826
LSTM	62.038	7.876	5.319	0.832
Proposed model	30.186	5.494	4.180	0.934

Experimental results for the predicting the number of vehicles passing through the junction showed that the proposed model was more successful than the compared models. After the proposed model, LSTM, RNN MLP, CNN, RF, SVM and LR were successful, respectively.

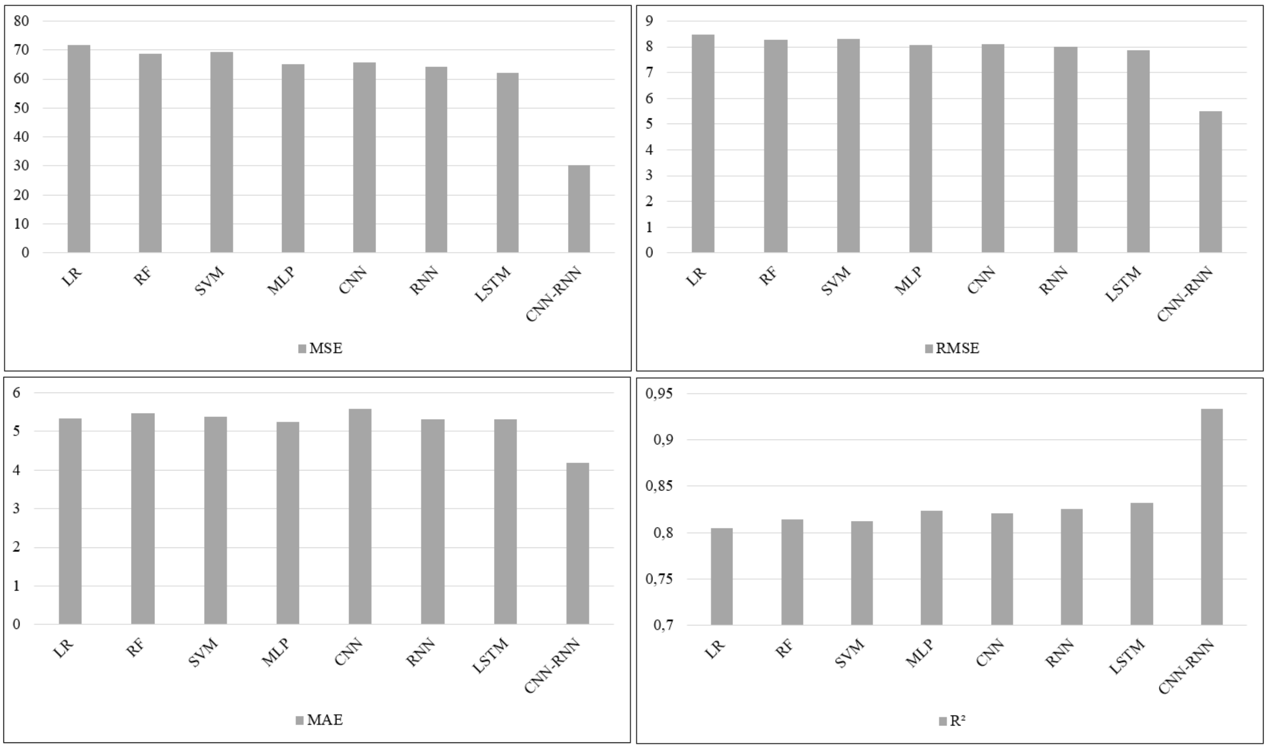

Figure 8. Experimental results for average speed prediction

Figure 9.a shows the graph of the predictions for the number of vehicles passing through the junction, and Figure 9.b shows the graph of the predictions of the average speeds of the vehicles passing the junction. As seen in Figure 9, the proposed model successfully predicted the number of vehicles and average speed value fluctuations.

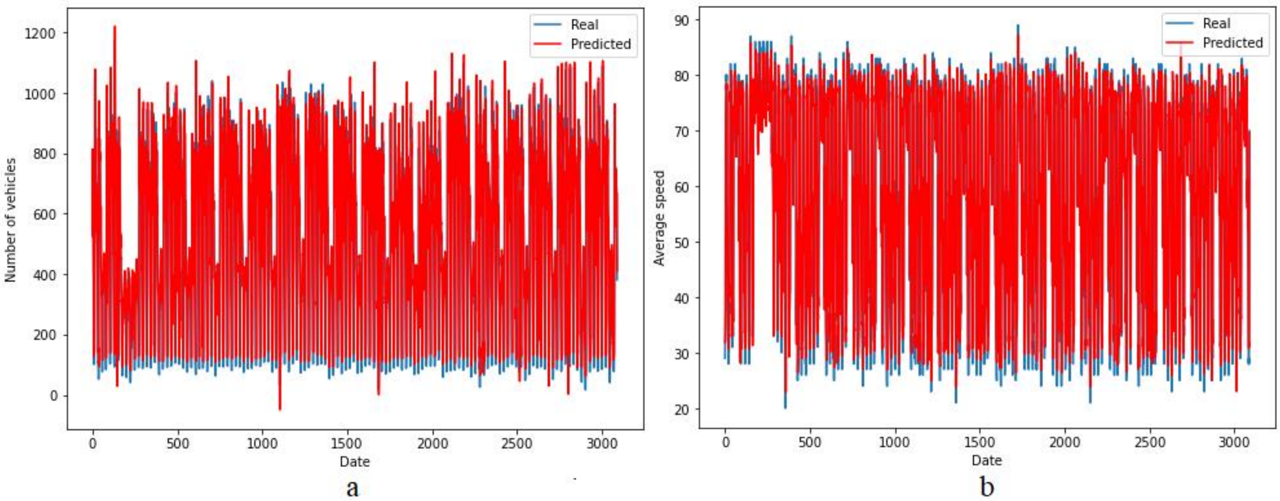

Figure 9. Prediction graphs of the proposed model

4. CONCLUSIONS

Traffic congestion is a part of the daily life of those living in big cities such as the megacity of Istanbul. Although the metro and metrobus lines reduce this problem a little, the fact that these vehicles are located only between specific routes causes many people to be exposed to traffic jams. Therefore, traffic congestion has economic effects and is a waste of time. At the same time, it causes a decrease in work efficiency due to the loss of time caused by traffic congestion, and the direct and indirect costs caused by traffic congestion cause an increase in the expenses of the enterprises. As a result, all these cause an increase in the price of the product or service offered.

In this study, a hybrid deep learning model was developed to predict the number and average speed of vehicles passing through the Bağcılar European Motorway. Bağcılar European Motorway is one of the most critical junction points of Istanbul. The developed model was compared with LR, RF, SVM, MLP, CNN, RNN, and LSTM. For each model, the results obtained according to MSE, RMSE, MAE, and R^2 metrics were analyzed comparatively. Experimental results showed that the proposed model has higher prediction accuracy than the compared models.

Experimental results showed that the developed hybrid model was more successful than the compared models. The developed model's success is that it can process very long input data that can be read in blocks by CNN and put together by RNN.

The success of LSTM compared to other models can be explained by the fact that LSTM has special units in addition to the standard units found in RNN. LSTM units contain a memory cell that can hold information in memory for long periods. In addition, a set of gates controls when information enters memory, when it leaves, and when it is forgotten. This structure allows the learning of longer-term dependencies.

The fact that RNN is more successful than CNN can be explained by the fact that CNN and RNN have different architectures. CNN is a feed-forward neural network using filters and pooling layers. RNN feeds the results back to the network. In CNN, the size of the input and the output are fixed.

The fact that MLP is more successful than CNN can be explained by the fact that MLP takes vectors as input and CNN takes tensors as input. The fact that RF is more successful than SVM can be explained by the fact that RF works with a mixture of numerical and categorical features. This allows RF to use the data as they are. SVM, on the other hand, maximizes the margin between different points and calculates the distance between points.

Experimental results for predicting the number of vehicles passing through the junction showed that the proposed hybrid model has 5302.627 MSE, 72.819 RMSE, 54.036 MAE, and 0.929 R^2. Experimental results for predicting the average speed of vehicles passing through the junction showed that the proposed hybrid model has 30.186 MSE, 5.494 RMSE, 4.180 MAE, and 0.934 R^2 values.

The results will reveal that the importance of controlling vehicle traffic is increasing, especially in metropolitan cities such as Istanbul, with a large population. Therefore, determining the current traffic congestion level should be taken under control together with the main reasons causing the traffic congestion, and possible measures to prevent this congestion should be implemented. For this reason, various traffic data obtained instantly through sensors, cameras, and mobile devices are analyzed using artificial intelligence methods. For this purpose, the results obtained in this study can be adapted to real-world applications, and effective planning can be done for traffic management.

REFERENCES

[1] W. Broere, "*Urban underground space: Solving the problems of today's cities*". Tunnelling and Underground Space Technology, 55, 245-248, 2016.

[2] G. Firdaus, & A. Ahmad, "*Noise pollution and human health: a case study of municipal corporation of Delhi*", Indoor and built environment, 19(6), 648-656, 2010.

[3] D. Muley, M. Shahin, C. Dias, & M. Abdullah, "*Role of transport during outbreak of infectious diseases: evidence from the past*", Sustainability, 12(18), 7367, 2020.

[4] Ş. İmre, & D. Çelebi, "*Measuring comfort in public transport: a case study for İstanbul*", Transportation Research Procedia, 25, 2441-2449, 2017.

[5] J. Zhang, F.Y. Wang, K. Wang, W.H. Lin, X. Xu, & C. Chen, "*Data-driven intelligent transportation systems: A survey*", IEEE Transactions on Intelligent Transportation Systems, 12(4), 1624-1639, 2011.

[6] S. Du, T. Li, X. Gong, Y. Yang, & S.J. Horng, "*Traffic flow forecasting based on hybrid deep learning framework*", In 2017 12th international conference on intelligent systems and knowledge engineering (ISKE), 1-6, 2017.

[7] O. Mohammed, & J. Kianfar, "**A machine learning approach to short-term traffic flow prediction: A case study of interstate 64 in Missouri,**" In 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, Missouri, USA, 1-7, 2018.

[8] W. Zhang, Y. Yu, Y. Qi, F. Shu, & Y. Wang, "*Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning*", Transportmetrica A: Transport Science, 15(2), 1688-1711, 2019.

[9] Y. Gu, W. Lu, L. Qin, M. Li, & Z. Shao, "*Short-term prediction of lane-level traffic speeds: A fusion deep learning model*", Transportation research part C: emerging technologies, 106, 1-16, 2019.

[10] J. Wang, R. Chen, & Z. He, "*Traffic speed prediction for urban transportation network: A path based deep learning approach*", Transportation Research Part C: Emerging Technologies, 100, 372-385, 2019.

[11] İ.C. Taş, & A.A. Müngen, "*Regional Traffic Density Estimation with Artificial Neural Networks and Support Vector Machines Methods*", Adıyaman University Journal of Engineering Sciences, 8(15), 378-390, 2021.

[12] İ. Takak, H. Görümez, H.İ. Türkmen, & M.A. Güvensan, "*Short, Medium and Long Term Traffic Flow Rate Estimation and Visualization Tool*", International Journal of Advances in Engineering and Pure Sciences, 33(4), 568-580, 2021.

[13] A. Essien, I. Petrounias, P. Sampaio, & S. Sampaio, S. "*A deep-learning model for urban traffic flow prediction with traffic events mined from twitter*", World Wide Web, 24(4), 1345-1368, 2021.

[14] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, & S. Dong, "**A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction**", Physica A: Statistical Mechanics and its Applications, 583, 126293, 2021.

[15] G. Zheng, W.K. Chai, J.L. Duanmu, & V. Katos, "**Hybrid deep learning models for traffic prediction in large-scale road networks**", Information Fusion, 92, 93-114, 2023.

[16] Internet: İstanbul Büyükşehir Belediyesi, Saatlik trafik yoğunluk veriseti, https://data.ibb.gov.tr/dataset/hourly-traffic-density-data-set

[17] G.Y. Oukawa, P. Krecl, & A.C. Targino, "*Fine-scale modeling of the urban heat island: A comparison of multiple linear regression and random forest approaches*", Science of the total environment, 815, 152836, 2022.

[18] F. Rustam, A.A. Reshi, A. Mehmood, S. Ullah, B.W. On, W.Aslam, & G.S. Choi, "*COVID-19 future forecasting using supervised machine learning models*", IEEE access, 8, 101489-101499, 2020.

[19] M. Kayakuş, & M. Terzioğlu, "*Yapay sinir ağları ve çoklu doğrusal regresyon kullanarak emeklilik fonu net varlık değerlerinin tahmin edilmesi*", Bilişim Teknolojileri Dergisi, 14(1), 95-103, 2021.

[20] A.L. Balogun, & A. Tella, "*Modelling and investigating the impacts of climatic variables on ozone concentration in Malaysia using correlation analysis with random forest, decision tree regression, linear regression, and support vector regression*", Chemosphere, 299, 134250, 2022.

[21] Y. Yang, & W. Chen, "*Taiga: performance optimization of the C4. 5 decision tree construction algorithm*", Tsinghua Science and Technology, 21(4), 415-425, 2016.

[22] M. Peker, O. Özkaraca, & B. Kesimal, "*Enerji tasarruflu bina tasarımı için ısıtma ve soğutma yüklerini regresyon tabanlı makine öğrenmesi algoritmaları ile modelleme*", Bilişim Teknolojileri Dergisi, 10(4), 443-449, 2017.

[23] V.A. Kumari, & R. Chitra, R. "**Classification of diabetes disease using support vector machine**", International Journal of Engineering Research and Applications, 3(2), 1797-1801, 2013.

[24] A.A. Heidari, H. Faris, S. Mirjalili, I. Aljarah, & M. Mafarja, "*Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks*", Nature-inspired optimizers, 23-46, 2020.

[25] I. Namatēvs, "*Deep convolutional neural networks: Structure, feature extraction and training*", Information Technology and Management Science, 20(1), 40-47, 2017.

[26] Z. Qiao, N. Sun, X. Li, E. Xia, S. Zhao, & Y. Qin, "*Using machine learning approaches for emergency room visit prediction based on electronic health record data*", In Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth, IOS Press, 111-115, 2018.

[27] A. Shewalkar, "*Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU*", Journal of Artificial Intelligence and Soft Computing Research, 9(4), 235-245, 2019.

[28] Y. Tian, R. Lai, X. Li, L. Xiang, & J. Tian, "*A combined method for state-of-charge estimation for lithium-ion batteries using a long short-term memory network and an adaptive cubature Kalman filter"*, Applied Energy, 265, 114789, 2020.

[29] A. Farzad, H. Mashayekhi, & H. Hassanpour, "*A comparative performance analysis of different activation functions in LSTM networks for classification*", Neural Computing and Applications, 31(7), 2507-2521, 2019.

[30] X.H. Le, H.V. Ho, G. Lee, S. Jung, "*Application of long short-term memory (LSTM) neural network for flood forecasting*", Water, 11(7), 1387, 2019.

.