IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 March 2023, accepted 10 April 2023, date of publication 26 April 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270395

RESEARCH ARTICLE

Urban Traffic Flow Estimation System Based on Gated Recurrent Unit Deep Learning Methodology for Internet of Vehicles

ABDUL HUSSAIN ALI HUSSAIN[1], MONTADAR ABAS TAHER[2], (Senior Member, IEEE), OMAR ABDULKAREEM MAHMOOD[2], YOUSIF I. HAMMADI[3], REEM ALKANHEL[4], (Member, IEEE), AMMAR MUTHANNA[5,6], (Senior Member, IEEE), AND ANDREY KOUCHERYAVY[6]

[1]Department of Architectural Engineering, College of Engineering, University of Diyala, Baqubah, Diyala 32001, Iraq
[2]Department of Communications Engineering, College of Engineering, University of Diyala, Baqubah, Diyala 32001, Iraq
[3]Department of Medical Instruments Engineering Techniques, Bilad Alrafidain University College, Baqubah, Diyala 32001, Iraq
[4]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[5]Department of Applied Probability and Informatics, Peoples' Friendship University of Russia (RUDN University), 117198 Moscow, Russia
[6]Department of Telecommunication Networks and Data Transmission, Bonch-Bruevich Saint Petersburg State University of Telecommunications, 193232 Saint Petersburg, Russia

Corresponding author: Reem Alkanhel (rialkanhal@pnu.edu.sa)

This work was supported in part by the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R323), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia; and the studies at St. Petersburg State University of Telecommunications. prof. M.A. Bonch-Bruevich were supported by the Ministry of Science and High Education of the Russian Federation by the grant 075-15-2022-1137.

ABSTRACT Congestion in the world's traffic systems is a major issue that has far-reaching repercussions, including wasted time and money due to longer commutes and more frequent stops for gas. The incorporation of contemporary technologies into transportation systems creates opportunities to significantly improve traffic prediction alongside modern academic challenges. Various techniques have been utilized for the purpose of traffic flow prediction, including statistical, machine learning, and deep neural networks. In this paper, a deep neural network architecture based on long short-term memory (LSTM), bi-directional version, and gated recurrent units (GRUs) layers has been structured to build the deep neural network in order to predict the performance of the traffic flow in four distinct junctions, which has a great impact on the Internet of vehicles' applications. The structure is composed of sixteen layers, five of which are GRU layers and one is a bi-directional LSTM layer. The dataset employed in this work involved four congested junctions. The dataset extended from November 1, 2016, to June 30, 2017. Cleaning and preprocessing operations were performed on the dataset before feeding it to the designed deep neural network in this paper. Results show that the suggested method produced comparable performance with respect to state-of-the-art approaches.

INDEX TERMS Flow prediction, BiLSTM, deep neural network, GRU, LSTM, urban transportation.

I. INTRODUCTION

In order for intelligent transportation systems (ITSs) to be used in urban traffic networks, accurate traffic forecasts are necessary. In other words, ITSs in urban traffic infrastructures rely heavily on accurate traffic flow predictions (TFPs). Due to its significance for the creation and improvement of ITS, TFP has been the subject of substantial research. While the efficacy of TFP models is often discussed in the literature, computing efficiency is often neglected.

As traffic data volumes grow, however, the pressing need for speedier models becomes more apparent. In other words, when planning urban cities, one should consider certain factors. For instance, most people now need cars in order to get around, what with the expanding economy and population. Congestion has worsened as the number of automobiles on

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

the road has increased. Vehicle restrictions and the implementation of sophisticated transportation systems are only two of the many regulations that have been implemented in an effort to reduce traffic congestion. Traffic flow prediction is affected by numerous complicated aspects owing to the chaotic intricacy of the traffic road infrastructure and the short-term movement of the population, making it very difficult to create an efficient TFP system.

As more and more people move to urban areas, more and more of them are adding to the already overwhelming traffic problems. Factors that contribute to this issue include increasing city populations, degrading infrastructure, inefficient and disjointed traffic signal scheduling, and a lack of real-time data. The repercussions are really serious.

According to INRIX, a company that collects and analyzes traffic data, commuters in the United States lost $305 billion in 2017 because of traffic congestion. This amount accounts for the money wasted on fuel, the time wasted waiting in traffic, and the expense of transporting products and services through crowded areas. Further, INRIX found that drivers in Bucharest and Bogota consume an average of 134 and 133 hours per year, respectively, due to gridlock [1].

Cities cannot simply construct more roads due to financial and logistical constraints; instead, they must adopt innovative approaches to traffic management when designing an urban city. More than half of the world's population now resides in urban areas, and that number is expected to rise to 68% by 2050, according to the United Nations [2].

As a result of rapid urbanization, many countries are struggling to keep up with the demands of an increasingly urban population and maintain sustainable growth. That is, for an urban smart city to function properly, mobility must be a top priority. The study and forecasting of traffic patterns are crucial components of mobility studies and innovations. Data from new sources, such as social media and GPS systems, has been rapidly added to the transportation data pool in recent years.

Road traffic prediction is only one example of how this information may be used to make cities run more efficiently. Single traffic flow forecast methods have been the norm in the research community; however, these have limited applicability. In recent years, research has emerged exploring how to combine existing approaches to create new hybrid approaches. Nevertheless, additional research is needed because this is still a young field of study. In addition, the majority of these hybrid approaches have been built on non-distributed, single-node systems, which severely restrict the scalability in terms of both data and problem size, as well as accuracy.

While previous research has examined the geographical correlation of traffic flows using Euclidean proximity or topological adjacency, it has discounted the major importance of higher-order connection patterns in a road network. And yet, little scholars have taken full advantage of the fact that traffic patterns exhibit unique multiple time-frequency features. Predicting how traffic is going to move is a crucial use for traffic monitoring technology in urban areas.

Traditional methods have only had limited success by employing artificial neural networks (ANNs) with a limited network architecture and inadequate training data for supervised learning. There have been a lot of studies done on TFP in the previous few decades because of its significance in ITS. Although spatial-temporal correlations are recognized to be adaptive in traffic networks, they are often extracted statically in existing TFP models.

Researchers have been thinking about TFP for decades, but it has remained difficult because of its stochastic, nonlinear character. In fact, making TFP reliable would benefit not only motorists but also pedestrians and cyclists, who are especially at risk, mostly on highways. Taking into account the needs of the public at large, prediction systems can be utilized to provide guidance to people on matters such as outside hazards, risky activities, and delays in schedule. Such factors are of interest when designing urban regions.

For traffic management and control to function properly, an accurate estimate of the current traffic status is required. Nevertheless, due to the varying nature of traffic on urban roads and at different times of day, it is challenging to develop prediction algorithms that can account for these factors. In particular, the traffic pattern identification is susceptible to bias and estimation error due to the limited selection of the traffic condition.

That is, this study proposes a deep learning technique based on the Gated Recurrent Unit (GRU) deep learning model to accurately predict the traffic flow at four intersections in an urban crowded city. The approach is that a combination of well-selected deep neural network (DNN) layers, such as the GRU and fully connected layers, is structured to form the foundation of the TFP system. When compared to state-of-the-art techniques, accurate results were obtained. The study can be formulated as follows:

- Addressing the past studies by investigating the most related and significant algorithms for traffic flow prediction in urban cities.
- Introducing a novel structure for DNN based on GRU that improves the traffic flow prediction values. The introduced structure focuses on the time-related relationship; therefore, the GRU units have been used. Furthermore, the suggested model can be widely deployed due to its simple structure and high-speed prediction output. This is because the building blocks of the recommended TFP system are simple and not computationally complicated.
- A real-world dataset has been employed in this study, which was gathered from kaggle.com. The dataset was first pre-processed and comprehensively studied, then it was used to train the suggested DNN model.

The rest of this work can be explained consequently as follows: the Related Work section (section II), which addressed the most significant work related to our study. The materials and dataset are then presented in Section III, where the DNN-GRU will be explained and the dataset study and analysis will be shown. The suggested DNN will be introduced in the fourth section, in which, the details of the DNN-GRU model will be discussed. The Results section (section V) shows the results and the discussions of the results. Last but not least, the concluding remarks are drawn in Section VI.

II. RELATED WORK

TFP studies are varied in methodologies, as will be stated in this section. Studies are either statistical or use deep learning techniques. However, DNN are not limited to the computer vision field or images; they were utilized in other studies such as security [3], [4]. That is, the study presented in this work will show the implementation of DNN in urban planning as well. It is essential to remember that ML comprises the entirety of DL, but not vice versa. In practice, DL constitutes a subset of ML that achieves great flexibility and strength by understanding to represent the world as a nested hierarchy of concepts, where each concept is described in regards to simpler ones and more abstract depictions are computed in terms of less abstract ones.

The study group in [5] provides an overview of the several strategies that can be used to anticipate and model traffic flow and then explores the limits of each strategy. This article is a discussion of the many different kinds of traffic and travel data sources. The article discusses a variety of notable big data analysis technologies, such as the Apache Spark platform. In the final part of this multi-part article, they outline a hybrid approach to road traffic prediction and offer a step-by-step guide to the hybrid traffic flow forecasting procedure.

The autoregressive integrated moving average (ARIMA) approach and the support vector machine (SVM) method both serve as the foundation for the hybrid method. The purpose of the study in [6] is to provide a comprehensive review and analysis of previous work that has been carried out by applying various artificial intelligence (AI) approaches, most notably a variety of machine learning (ML) models. The document compiles the models under their respective subfields of AI, and it reviews the strengths and limitations of each model.

Smart methodologies used in the analysis of movement data for the forecast of traffic flow in urban regions are categorized in a subsequent review of research [7]. In this case, the outcomes of employing these methods are displayed. The methods used are also detailed and assessed so that the potential and potential pitfalls of these intelligent approaches may be grasped. Thus, researchers present the data sets utilized in the literature and made available for use, and authors compare the quantitative results of the precision of the various techniques, highlighting advantages and limitations, allowing someone to identify the related challenges and, from there, suggest an overall taxonomy in which the knowledge gained in this traffic flow evaluation merges from a computational perspective.

A certain investigation is to give a complete overview of traffic prediction approaches in order to lay the groundwork for comprehending the open research issues in traffic prediction [8]. Because of its rapid progress and promise in traffic prediction, through a focus on multivariate traffic time series forecasting, they zoom in on the latest developments and new research prospects in AI-based traffic forecasting approaches. They begin by cataloging and explaining the several forms of data and sources found in the scholarly works. Then, they describe the prediction approaches and applications before moving on to classify the primary data preparation techniques used in the context of traffic prediction. Finally, researchers address some potential future research avenues and identify some key research difficulties in traffic prediction.

Short-term traffic flow predictions (TFP) were improved by proposing an adaptive hybrid model that takes into account both the periodicity and unpredictability of traffic flow and the limits of single prediction methods [9]. As a first step, researchers attempted to foretell traffic patterns using both the linear ARIMA approach and the non-linear Wavelet Neural Network (WNN) approach. Next, using fuzzy logic, they compared and averaged the results from the two algorithms and then used the weighted average as the final estimate of the hybrid model's expected traffic levels. Conclusions suggest the hybrid system outperforms the two individual algorithms in both stable and dynamic scenarios for predicting short-term traffic flow.

A unique hybrid architecture, the wavelet-based higher-order spatial-temporal approach (Wavelet-HST), is proposed in [10] for estimating traffic volumes throughout an entire network. At its core, Wavelet-HST is a discrete wavelet transform that is used to break down traffic data into many frequency-specific parts. Finally, they suggest a motif-based graph convolutional recurrent neural network (Motif-GCRNN) to acquire higher-order spatio-temporal correlations of road speed using low-frequency parts and subsequently use ARIMA approaches to mimic variability from the high-frequency constituents. However, the systems in [9] and [10] are built around signal processing (Wavelet), Neural Network (NN), ARIMA, and fuzzy logic, which means the system became very complicated.

In [11], a k-nearest neighbor (KNN) model is presented for predicting the short-term traffic flow with high accuracy. Three components—a historical database, a search method and algorithm parameters, and a forecast plan—make up a k-NN-based short-term urban expressway traffic prediction system. To boost prediction precision, it is necessary to first normalize the effective data after preprocessing the actual information and removing outliers. Finally, in Matlab, a short-term traffic forecast model using the k-NN nonparametric regression technique has been created. The measured traffic flow data from a segment of an urban expressway in Shanghai is used to evaluate the accuracy of the average and weighted k-NN nonparametric regression models. The results demonstrate that the suggested method has an accuracy of greater than 90% and is feasible for use in predicting traffic flows in the near future.

The research in [12] presents the development of a Deep Neural Network (DNN) infrastructure for analyzing and forecasting traffic conditions from real-time mobility big data within an ANN framework. The proposed DNN model uses logistics regression (LR) analysis to identify times of congestion and non-congestion. The proposed DNN architecture consisted of three layers [40, 50, and 40] of neurons in the layers, respectively. Activation functions for each layer are hypertangent functions, and the LR analyses are performed using the Traffic Performance Index (TPI). The structure in this work was very simple and could not predict very well the whole time series.

The purpose of the research presented in [13] was to create an easy-to-implement hybrid model for traffic volume forecasting by fusing the ARIMA and Radial Basis Function (RBF) ANN (RBF-ANN) methods. Combining systems allows for the capture of more nuanced characteristics of traffic flow dynamics. The linear portion of the traffic flow time series was modeled using the ARIMA model. After modeling the ARIMA modeling residuals, the RBF-ANN approach was used to extract the nonlinear element. The temporal combinations of five minutes were used to fine-tune the hybrid models. China's traffic data was used to verify the accuracy of the suggested hybrid technique. The results showed that, compared to using either the ARIMA or RBF-ANN model alone, the hybrid models had superior predictive capability. The two models' synergy in the combined approach made it a powerful tool for short-term road traffic prediction. Thus, again, there is a mix between two different analysis approaches, ARIMA and NNs. Accuracy and interpretability have both seen significant gains thanks to the introduction of attention-based models in recent decades, particularly in the area of natural language processing. As a result, we're motivated to present attention's use for doing things like predicting traffic.

A DNN-based TFP that pays both temporal and spatial attention is proposed in [14]. Focusing in on specific locations and times allows us to take advantage of the relationships between individual road segments and between individual time intervals. The experimental outcomes with a real-world traffic dataset prove the higher performance of the suggested approach. Furthermore, the results demonstrate that utilizing several data resolutions may aid in enhancing prediction accuracy. It is also shown that the suggested model has promise for enhancing our comprehension of spatial-temporal correlations in a TFP.

Existing models perform poorly in harsh environments because they don't take the impact of weather variations on traffic flow into account. Research into TFP driven by both traffic and weather data is required since the TFP model based on traffic data has important shortcomings due to the complex attributes of traffic data and the attribute of being subject to external weather variables.

To better reflect the temporal correlation and periodicity of traffic flow information and the perturbation of weather conditions, researchers in [15] offer a combination architecture of stacked autoencoders (SAE) and RBF to anticipate the flow of traffic. To begin, SAE is applied to the traffic data in several time slices to derive an initial estimate. To make further forecasts, RBF is then employed to identify the connection between weather instability and traffic flow frequency. Ultimately, the two forecasts are fused at the classification stage using a second RBF to produce a rebuilt forecast with improved accuracy.

An efficient and practical TFP model is provided in [16]. In this case, the standard autoregressive theory is reworked on the premise that only a subset of available historical traffic measurements is truly relevant for making accurate predictions about upcoming traffic quantities. By applying a hyperparameter-controlled qualification constraint, an effective cluster-based technique selects the relevant historical data. As a result of this modeling strategy, a sparse least squares problem arises, which is effectively handled by employing a unique, straightforward, repetitive methodology based on generalized approximation sparse pseudoinverse.

Rome is one of the Italian cities hardest hit by the chronic traffic congestion that has plagued Italy's road network for decades. Non-autonomous vehicle traffic prediction was performed using a heuristic model based on a Levenberg-Marquardt ANN in [17]. Inductive loop detectors and video cameras were used as sensing devices to collect data on traffic, with the help of a few carefully chosen characteristics (such as vehicle speed, time of day, traffic volume, and the number of cars on the road at any one moment). The model's R2 was 0.99892 in training, 0.99615 in testing, and 0.99714 in regression. In terms of traffic control and the supply of convenient travel routes for walkers and automobiles, the findings of this research contribute to the expanding body of knowledge on traffic flow modeling and aid urban planners and transportation administrators.

In order to solve the issue of multi-modal TFP with sparse data, the authors of [18] suggest a pattern-adaptive generative adversarial network (PA-GAN). Bayesian inference is implemented in PA-GAN to assign multi-modal probability distributions to the system characteristics. Therefore, the posterior sample will dynamically trigger the associated characteristics of each flow pattern based on the context features, allowing for a focused state estimate for every sequence. The PA-GAN then employs the traffic state producer and the discriminator to learn transportation patterns from the limited measurements, with the prediction being corrected via an error-feedback process using multi-level attributes in an encoder-decoder framework.

They use two publicly available datasets to undertake extensive examples regarding multi-modal traffic conditions and sparse selection, and then researchers use these results to draw conclusions about the efficacy of the suggested PA-GAN. The results of the experiments show that the PA-GAN can beat alternative estimate techniques and that the Bayesian Inference may enhance the learning network's flexibility to different traffic situations.

The seasonal autoregressive integrated moving average model (SARIMA) is frequently used to process seasonal time data due to its superior linear fitting features. On the other hand, the non-autoregressive dynamic neural network (NAR) is equipped with both nonlinear interpretation abilities and a significant memory function. In other words, you can use them to construct integrated forecasting models. Two merged models are built by mixing the SARIMA and NAR models with 15 hidden layer neurons and a fourth-order delay [19]: model 1, which uses the linear and nonlinear component combination approach, and model 2, which uses the MSE weight combination method.

For TFP in both data-rich and data-poor settings, [20] creates and assesses various ARIMA and Long Short-Term Memory (LSTM)-based approaches. However, the LSTM DNN was constructed based on two consecutive LSTM layers (128 and 64 units long, respectively). In contrast to the good findings acquired with LSTM networks, the findings obtained with ARIMA models reveal their low performance on large datasets. Still, they both saw patterns and the traffic's cyclicity.

To better anticipate holiday traffic, the research [21] applies the fluctuation coefficient technique, which is common in passenger flow management. Researchers separated vacation traffic into its regular and erratic components based on an examination of the features of traffic flow. The LSTM model is used to predict the steady flow, while the fluctuation coefficient is used to predict the erratic flow. This technique can make up for a lack of past records, and its efficacy has been confirmed experimentally.

The authors in [22] present a path-based DNN methodology for predicting network traffic speeds, fitting the machine to the spatiotemporal characteristics of traffic patterns throughout the whole network. Since the most traveled routes typically have a more consistent and dominant traffic flow, the most helpful data for speed prediction can be gleaned from studying these routes. After that, numerous layers are stacked along the temporal dimension, and a bidirectional LSTM NN (Bi-LSTM NN) is implemented to mimic each crucial path. This ensures that each model has high levels of interpretability while effectively capturing the spatial-temporal features of the path to improve speed prediction. When all the models' outputs have been combined, a final forecast can be made.

In [23], they offer a new TFP model called EnLSTM-WPEO that makes use of LSTM ensemble learning, no negative constraint theory (NNCT) weight integration, and the population extremal optimization (PEO) algorithm. Time lag is a major factor that can affect the accuracy of a predictor, so the first step is to build a colony of LSTMs to do forecasts with varied time lags. The PEO-based NNCT approach is then applied to calculate the ensemble model's weightings, which is the second step. The simulation results for six major Seattle highway datasets have shown that the recommended EnLSTM-WPEO is preferable to six others prominent TFP methods.

Using the variational mode decomposition (VMD) technique and the LSTM model, the authors of the work published in [24] suggest a hybrid TFP approach. To begin, we use the VMD technique to break down the raw data on traffic flows into a set of intrinsic mode function (IMF) components. Second, we create separate LSTM models to forecast individual components of the IMF. Predictions are made on a scale from one step to three steps, depending on the model used. At last, the TFP values are calculated by adding all the predictions made for each individual part. Data collected from inductive loop sensors installed along Shanghai's north-south expressway is used to probe the suggested hybrid model's predictive abilities.

The research in [25] creates an LSTM-based TFP model using time-series data from the Keras library within the context of deep learning to forecast traffic conditions in the future. Using the US-101 data set, a comparison experiment reveals that LSTM is more accurate than the GRU model at predicting the speed of traffic flows.

For the purpose of analyzing the multivariate road traffic time series, a TFP method built on the LSTM model has been developed [26]. It takes into account both past and present traffic on the route in question, in addition to past and present traffic on other nearby roads whose use is dependent on that of the one in question. They employ a stacked LSTM model to estimate traffic flows on weekdays and weekends with equal precision. Extensive simulations based on actual traffic information have been conducted, and the suggested method has been compared to established, state-of-the-art methods for TFP. The results demonstrate that the suggested method provides estimates of traffic flow that are close to being correct on weekdays and weekends and under normal and peak traffic situations.

Using the continuous and complete traffic flow data in the past period of time of the target prediction section and the incomplete traffic flow data in the past period of time of the target prediction section, a TFP technique utilizing a mix of multiple linear regression (MLR) and LSTM is proposed in [27] to jointly predict the traffic flow modifications of the target section in a short amount of time. When historical data on the intended road segment's traffic flows is incomplete, the aforementioned model can be employed to reliably anticipate how those flows will evolve in the future. The accuracy of the predictions is on par with that of existing mainstream predictions utilizing continuous and quasi-target link traffic data. Conversely, whenever the data time period is sufficiently short, a slight enhancement occurs.

In light of the spatial-temporal correlation of traffic flows, [28] presents a hybrid DNN approach, the AutoEncoder (AE), followed by two Gated Recurrent Units (GRUs) layers. The model is capable of efficiently utilizing the topological spatial association of the road infrastructure, minimizing the algorithmic sophistication, and circumventing the issue of destabilizing forecasting outcomes brought on by the advent of low traffic information, all while taking into account the effects of upstream and downstream interconnections on the traffic conditions of the current link. The outcomes demonstrate that this technique is a powerful TFP paradigm, with greater forecast accuracy than the conventional estimation method.

Current state-of-the-art works typify the temporal and spatial correlation of traffic patterns using Recurrent NN (RNN) and Convolutional NN (CNN) -Graph Convolutional Networks (GCN), with GCN RNN (GCRN) being the former. However, the large processing complexity of GCRN makes it challenging to apply it to the massive road networks. To solve this issue, they propose modeling the spatial-temporal interdependence of traffic flow with a Fast GCRNN (FastGCRNN), which abstracts the road network as a geometric graph [29]. Particularly, researchers integrate the GRU unit to grab the temporal correlation of traffic flow, incorporating the spatiotemporal features into Seq2Seq relying on the Encoder-Decoder approach, and use the FastGCN unit to proficiently grasp the topographic connection between the roadways and the surrounding streets in the graph while lowering the supercomputing complexity through significance sampling.

The study in [30] put up a plan for the TFP of Hangzhou's Wenyi Road. There are four intersections along Wenyi Road. There is a temporal and spatial correlation between the four crossings, with the same shifting trend in traffic at the same time, indicating that the streets have an effect on one another. Researchers offer the IMgru model, which is based on this aspect of traffic flow, to more accurately identify the temporal aspects of traffic flow. It is also suggested to collect the stochastic aspects of traffic concurrently using a model that combines the GCN part with the IMgru component. Last but not least, the Wenyi Road dataset is split into peak periods and off-peak periods for forecasting based on the morning and evening maximum features of Hangzhou. When compared against five reference models and a cutting-edge technique, the IMgruGcn model performs admirably. To put it another way, the IMgruGcn model was developed separately for peak and off-peak times.

Using long and high-frequency data, including 21,843 samples collected at 5-minute intervals, researchers sought an appropriate model to produce a one-step forecast [31]. High-frequency measurements lead to the appearance of daily seasonality, while extensive duration of data indicates weekly seasonality. According to the documentation from the data source, the data is extremely credible, with only 45 incomplete observations and a reliability score of 40. Particularly, researchers found that the double seasonal Holt-Winter (DSHW) model and the trigonometric seasonality model (Box-Cox transformation, ARIMA, trend, and seasonal elements) are the two statistical models best suited to double seasonality data (TBATS). In addition, the Five ML Models of Multiple Learning Perceptron (MLP), LSTM, GRU, CNN, and CNN-LSTM have been developed.

The standard was set using the seasonal ARIMA. Both univariate and multivariate analyses utilized the ML models. First, they do an additive decomposition of the time series into its trend, seasonal, and residual components so that they can use them as attributes in their multivariate modelsits trend, seasonal, and residual components so that they can use them as attributes in their multivariate models. The fact that the decomposed parts are both autonomous and correlated with the anticipated variable provides the impetus for this strategy. Researchers show that this method improves all ML models by analyzing their performance on three different roadway segments. This study's significance is amplified in situations where no other attributes are readily available for analogous data. As far as researchers are aware, additively decomposing parts have not been employed as attributes in multivariate ML models.

It is recommended to use a CNN-Bidirectional GRU unit with an attention technique (CNN-BiGRU-attention) and two Bidirectional GRU modules, each with their own attention function, to create a multifeature fusion model using deep learning techniques [32]. Daily and weekly periodic characteristics of the traffic flow are extracted using the CNN-BiGRU-attention component, while local trend attributes are extracted using the CNN-BiGRU-attention component, and long-term dependent characteristics are extracted using the two BiGRU-attention units. In addition, the attributes gathered by each unit are fused using an attribute fusion layer in the framework. Next, via simulated tests, choices are made based on the model's neuron count, loss function, and additional factors like the optimization process. At last, the field data is used to train and evaluate a multifeature fusion model. The outcomes show that the suggested methodology is more accurate at TFP and more stable overall. The experimental results demonstrate that the multifeature fusion model outperforms the baseline models in terms of predictive accuracy when using the same dataset.

They develop a bidirectional GRU network structure for TFP using a multi-range mask GCN, and they use three modules to learn spatial-temporal characteristics at different time scales: recent features, daily features, and weekly features [33]. Designers recommend a bidirectional mask GCN-GRU layer to learn node-specific forward and backward spatial information, in which a mask adaptive pairwise matrix approach is created to learn geometric distribution dynamically in the traffic information and a mask matrix is executed to filter the noise geometric distribution throughout the dynamic graph learning for more accurate and robust forecasting.

Extensive experimental validation on four datasets shows that the suggested technique outperforms state-of-the-art comparison techniques in terms of TFP accuracy. With the goal of filling in gaps in traffic data, the authors of [34] create a novel DNN architecture called the Dynamic GCN Recurrent

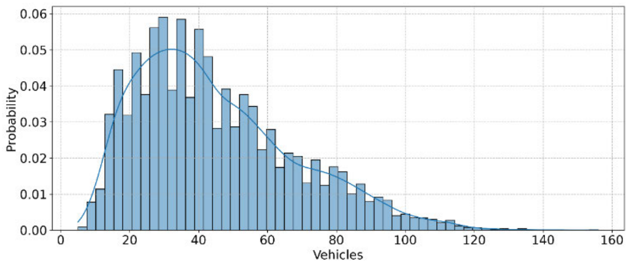

FIGURE 1. Probability distribution of the vehicles in the first junction.

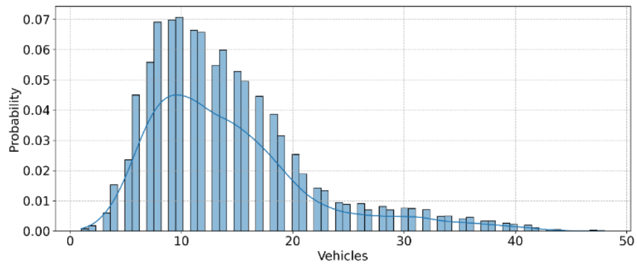

FIGURE 2. Probability distribution of the vehicles in the second junction.

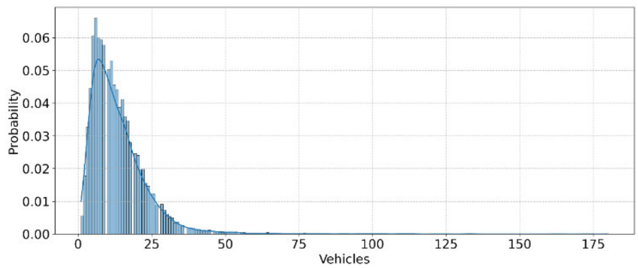

FIGURE 3. Probability distribution of the vehicles in the third junction.

Imputation Network (DGCRIN). To accurately simulate the dynamic spatiotemporal dependencies of the road network, the DGCRIN makes use of a graph generator and a DGCN gated recurrent unit (DGCGRU).

Moreover, a supplementary GRU is trained to learn the data's missing sequence, and a fusion layer with a decay process is implemented to fuse disparate data sets. Due to its flexible design, the DGCRIN can handle missing data in even the most complex of situations. That is, the model is constructed mainly for the management of missing data in the sequences.

III. MATERIALS AND DATA

This part of the article discusses the dataset that will be used in this work as well as the GRU and fully connected layers that will be utilized to construct the suggested DNN model.

A. DATASET

Earlier, it was mentioned that congestion is getting worse in cities all over the globe. Growth in urban areas, breakdowns in infrastructure, poor coordination of traffic lights, and a lack of real-time information all play a role. That is, distinct and various datasets have been put together for the purpose of studies and to provide sufficient planning for urban cities.

For instance, there is the accident dataset in the United Kingdom. There is also a dataset that counts the number of cars and trucks on the road. Multiple sites are used to gather data at 5-minute intervals. Also, there is the traffic flow dataset for Vietnam, and further datasets are all located at kaggle.com. The employed dataset in our study contains 48120 observations of the number of vehicles each hour at four different junctions. The sensors on each of these junctions were collecting data at different times, hence you will see traffic data from different time periods.

Some of the junctions have provided limited or sparse data, which required thoughtfulness when creating future projections. Therefore, this dataset is a tally of car counts at four intersections every hour.

There are four functions available in the.csv file: First, there's the ID; then there's the date and time; then there are the roads; and finally, there are the vehicles. Note that the dataset has no missing data entries.

However, Figure 1 shows the probability distribution of the vehicles at the first junction. From Figure 1, the probability density of the vehicles is a normal distribution, mostly between 10 and 60 vehicles, with a probability higher than 0.02. While the concentration of vehicles with a probability lower than 0.02 can be considered a rare event. For instance, the number of vehicles between 100 and 110 is less than 0.005, and when the number of vehicles increased, the probability decreased significantly.

In the second junction, as shown in Figure 2, the probability of vehicles is unlike that of the first junction. For example, above a 0.2 probability level, the number of vehicles is mostly between 5 and 22. That is, it can be said that junction 2 is less congested than the first junction. That is, the second junction has a smaller probability of vehicles. However, the distribution is again a normal distribution, as shown in Figure 2.

The third junction is also having normal distribution, as shown in Figure 3. Thus, at most, the number of vehicles is between 3 vehicles and 20 vehicles, when the probability is higher than 0.02 (inclusive). This means that the third junction has almost similar attitude of the second junction.

Moreover, the probability of number of vehicles more than 50 is very small, while in the first junction, this number of vehicles is not impossible and has a probability of 0.028, which is very high as compared to the third junction, as shown in Figure 3.

On the other hand, the probability of 100-vehicles in the first junction is 0.005, while in the third junction, the 100-vehicles probability is dramatically low. Thus, the third junction is also considered as less congestion compared to the first junction. For the fourth junction, the probability distribution is further normal, as shown in Figure 4.

Nonetheless, when the probability is 0.02 and above, the concentration of vehicles is 2 up to 13 vehicles, moreover, this junction also is less congested than previous junctions, where

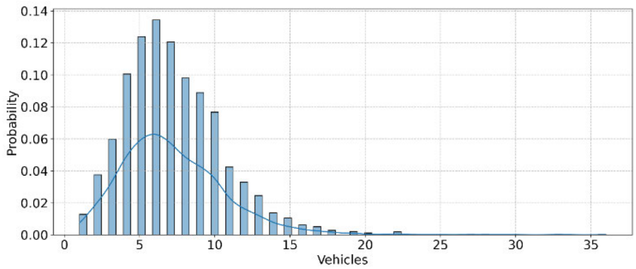

FIGURE 4. Probability distribution of the vehicles in the fourth junction.

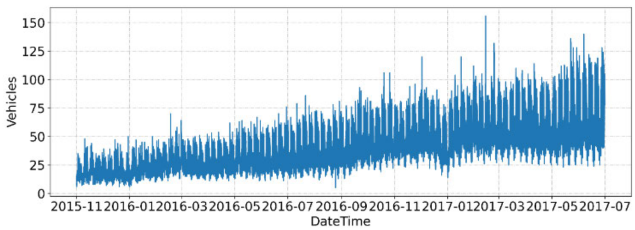

FIGURE 5. Vehicles concentration along time for the first junction.

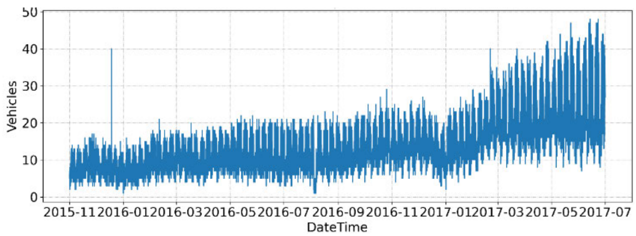

FIGURE 6. Vehicles concentration along time for the second junction.

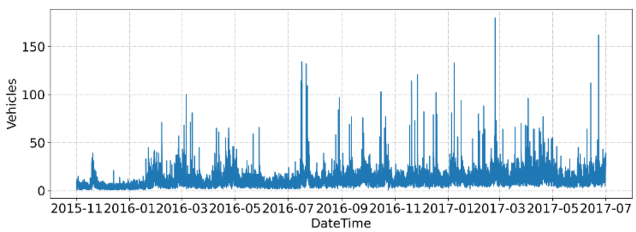

FIGURE 7. Vehicles concentration along time for the third junction.

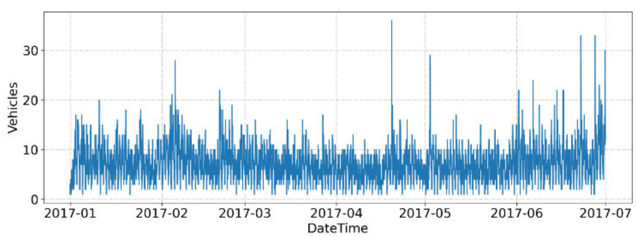

FIGURE 8. Vehicles concentration along time for the fourth junction.

maximum number of vehicles is 6 with higher probability, 0.135, and the higher number of vehicles, 35, is at probability less than 0.005, which confirm the little congestion at this junction, as compared to previous three junctions.

TABLE 1. Stationarity ADFT check for the four-time series junctions.

Junction	ADFT statistic	p-value	1%	5%	10%	Test Result
1	-15.3	0	-3.431	-2.862	-2.567	Stationary
2	-21.8	0	-3.431	-2.862	-2.567	Stationary
3	-28.0	0	-3.431	-2.862	-2.567	Stationary
4	-17.98	0	-3.432	-2.862	-2.567	Stationary

Figures 5-to-8 show the vehicle concentration during the time for the first, second, third, and fourth junction, respectively. That is, in Figure 5, it can be seen that the vehicles congestion from November 2015 to July 2017 being increased in the first junction from around 50 vehicles to more than 125, in 2015 and 2017, respectively.

The same behavior is captured in Figure 6, which is the vehicles distribution for the same period of the first junction in the second junction. But it is observed that the minimum number of vehicles in the second junction during 2017 is between 5 and ten vehicles, while in the first junction, the minimum number of vehicles is around 25, which reflects the extreme congestion of the first junction compared to the second junction.

In the third junction, the behavior is dissimilar to that in the two previous junctions, where the number of vehicles in August 2016 has a peak (more than 125 vehicles) and some few congested days in 2017, where the number of vehicles was more than 150. However, the number of vehicles in the third junction almost below 50, as shown in Figure 7. Last but not least, the last junction, fourth junction, the available data records are for January 2017 to July 2017. At the most times, the concentration of the vehicles is lower than the three previous junctions, as shown in Figure 8.

As a preprocessing operation to the dataset is the normalization operation. However, in this work, the z-score normalization will be conducted, where values are symmetrically distributed about the mean and have a standard deviation of one, when the attribute is transformed into zero and the resulting distribution has a standard deviation of one. The standardization can be computed mathematically by taking the feature value and removing it from the mean, then dividing that number by the standard deviation.

$$X_n = \frac{X_o - m}{\sigma} \tag{1}$$

where X_n is the normalized value, X_o is the original data before normalization, m stands for the mean value, and σ is the standard deviation. Next, based on the data from the junctions' observation logs, the seasonality can be removed by performing a weekly difference at the first junction, a daily difference at the second, and an hourly difference at the third and fourth. Last operation to be conducted is the stationarity check.

The Augmented Dickey Fuller Test (ADFT) is conducted on the time series for the four junctions, Table 1 lists the check results of the ADFT. It is shown that the ADFT statistic values

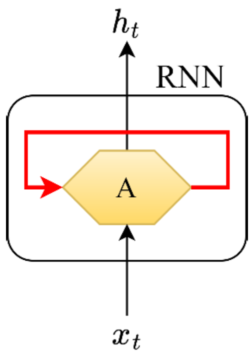

FIGURE 9. Rolled recurrent neural network (RNN) conceptual diagram.

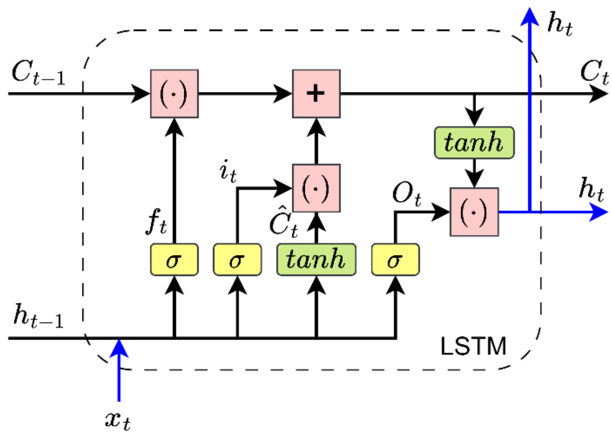

FIGURE 10. Long short-term memory (LSTM) operational diagram.

are, respectively for the first to the fourth junction, -15.3, -21.8, -28.0, and -17.98.

The p-value of the ADFT check is 0 for the four-junctions, the 1%, 5%, and 10% critical values are -3.431, -2.862, -2.567, respectively for the first three junctions, while the fourth junction the only difference is in the 1% critical value which was -3.432, which is slightly different than the others and will not change the check results. Thus, according to the check, all time-series junctions are stationary. Since the dataset is stationary, it is worth to split it to train and test subsets for the next step.

In other words, preparing the dataset to be used in the DNN model that will be discussed later.

B. DEEP LEARNING MODULES

The deep learning modules which will be gathered in this work are GRU, BiLSTM, and Dense (fully connected) layers. These layers will be discussed in this section briefly. In a NN, a dense layer is one in which each neuron is directly connected to every other neuron in the layer above it.

ANNs typically employ this layer as their primary connection layer. However, humans think continuously. This is a major drawback of conventional NNs. For instance, classify every movie event. How a typical NN could use its reasoning about cinematic events to educate later ones is unknown. RNNs fix this. Looped networks preserve information.

In Figure 9, we can see a section of the NN, A, which takes in the data represented by x_t and returns the result, h_t. The network can transfer data from one stage to another with the use of a loop. These iterative processes add a mystical air to RNNs. The more you contemplate, though, the less you realize they differ from a standard NN. It's helpful to visualize a RNN as several instances of an identical network, each of which relays a signal to the one that comes after it.

The RNNs' close relationship to sequences and lists is demonstrated by their chainlike structure. Because of their optimal structure for this kind of information, NNs are an obvious choice.

RNNs are typically composed of a series of NN modules that are repeated. Typically, this recurrent module in RNNs will consist of a single *tanh* layer, which is quite simple in comparison to more complex RNNs. The long-term reliance issue is specifically addressed in the architecture of LSTMs. They don't have to put in extra effort to remember things for long periods of time because it's basically their natural tendency.

There is a similar chain-like arrangement in LSTMs, but somehow the repeating module is organized differently. There are four layers of NNs here, all of which interact with one another in very specific ways. At the outset of LSTM, see Figure 10, we must choose which aspects of the cell state to ignore. The "forget gate layer," a sigmoid, is responsible for making this determination. A value between 0 and 1 is returned for each cell state C_{t-1} based on the inputs h_{t-1} and x_t. On this scale, 1 is "keep" and 0 is "ignore."

That is, the output of the first sigmoid, on the very left-side lower part in Figure 10, can be formulated as,

$$f_t = \sigma \left(W_f \cdot [h_{t-1}, x_t] + b_f \right) \tag{2}$$

where W_f is the weight and b_f are the bias. So following thing to do is figure out what brand-new data will be added to the cell state. You'll need to divide this into two sections. To begin, an input gate layer (a sigmoid) chooses which values will be modified. Then, the *tanh* layer generates a set of potential new state values ($\hat{C}_t$) in a vector form. After fusing these two elements, a new version of the state can be generated

$$i_t = \sigma \left(W_i \cdot [h_{t-1}, x_t] + b_i \right) \tag{3}$$

$$\hat{C}_t = \tanh \left(W_C \cdot [h_{t-1}, x_t] + b_C \right) \tag{4}$$

The transition from the previous cell state C_{t-1} to the current cell state C_t is now complete. The next stage is to put into action the course of action determined upon in the preceding steps. Multiplying the previous state by f_t. Following this, we incorporate $i_t \times \hat{C}$, in other words, (3) × (4). Here are the revised potential values, adjusted for the degree to which we decided to alter each existing state value

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \tag{5}$$

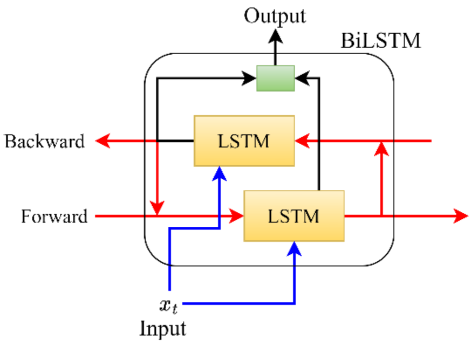

FIGURE 11. **Structure diagram of BiLSTM module.**

At this point, it's time to settle on some kind of output. Our cell status will serve as the basis for this output, albeit after some filtering. As a first step, we execute a sigmoid layer to determine which aspects of the cell state will be output. The outcome of the sigmoid gate is multiplied by the cell state that has been *tanh*-transformed (to force the quantities to be between 1 and -1) before being output

$$O_t = \sigma\left(W_O\left[h_{t-1}, x_t\right] + b_O\right) \tag{6}$$

then,

$$h_t = O_t \times \tanh\left(C_t\right) \tag{7}$$

That is, the current state, h_t, is obtained through the last expression, which is the multiplication of the O_t by the *tanh*(C_t). On the other hand, Time series or sequence data can be used by the BiLSTM layer to learn the long-term relationships in both directions between the individual time points. When training a network, such dependencies might be helpful in ensuring that each time step includes data from the whole time series, as shown in Figure 11.

A variant of the RNN family, the GRU can sometimes outperform the more common LSTM. When compared to LSTM, GRU is faster and utilizes less memory. Additionally, GRUs solve the issue of updating network weights with vanishing gradient values. The NN may become untrainable if the grade becomes too small over time due to back propagation.

If a NN's layers can't learn, RNNs can forget far longer sequences. The two gates in a GRU—the update gate and the reset gate—are what allow it to accomplish this. These gates are in charge of filtering information at the input and can be taught to remember details from earlier in the process. As a result, it can use the information gleaned from one event to inform its predictions about other events [35], [36].

Figure 12 shows the conceptual structure of the GRU. To determine the value of the update gate z_t for time step t, the network node multiplies x_t by its own weight, $W(z)$, whenever an x_t input is detected. Similarly, h_{t-1} stores data from the preceding t-1 units and is multiplied by its own weight, $U(z)$. After adding them up, we use a sigmoid activation function to make the total value shrink to something in the range of 0 to 1.

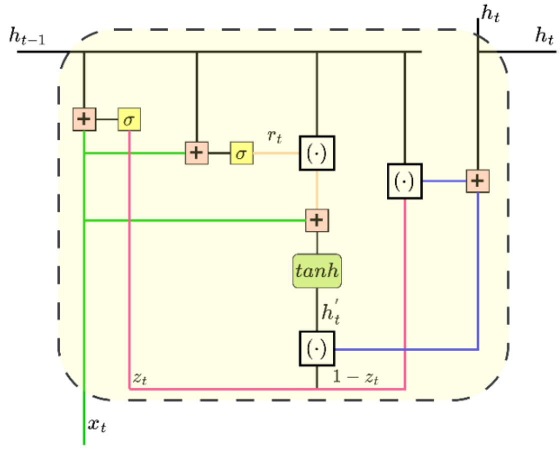

FIGURE 12. **Gated recurrent unit (GRU) neural network structure.**

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right) \tag{8}$$

where σ denotes the sigmoid function, *tanh* stands for the hyperbolic tangent function, and the Hadamard dot product is represented by $(\cdot)$ operation. That was the principle of operation of the update-gate. Regarding the reset-gate, this model gate is employed primarily to determine the extent to which previous knowledge is to be forgotten. The update gate's function is same to this one. The gate's role and the relative weight of the objects are key differentiators.

$$r_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right) \tag{9}$$

Just like before, multiplying h_{t-1} and x_t by their respective weights, add up the products, and then use the sigmoid function. To implement a new memory content that makes use of the reset gate to keep track of historical data. Following is the formula for determining it:

$$h_t' = \tanh\left(Wx_t + r_t \cdot Uh_{t-1}\right) \tag{10}$$

That is, in the last expression, (10), the weight W is multiplied by x_t (the input). Now, the dot product of the reset gate (r_t) with the Uh_{t-1} (remember that U is a weight) will be added up to the previous multiplication (Wx_t). Hence, this result will be passed to the hyperbolic tangential function. Finally, the network must compute h_t, the vector that contains the current unit's data and is passed down via the network.

$$h_t = z_t \cdot h_{t-1} + (1 - z_t) \cdot h_t' \tag{11}$$

The update gate is essential for accomplishing this. It decides what information to glean from the present memory contents (h_t') and what information to glean from the prior stages (h_{t-1}). It is now possible to observe how GRUs store and filter data utilizing their update and reset gates. Since the model does not discard the new input every time but instead stores the pertinent information and passes it on to subsequent time steps in the network, this solves the vanishing gradient problem. Given adequate training, they are capable of remarkable performance, even in highly complex situations.

Consequently, the abovementioned NN layers and modules will be combined in somehow to create the intended structure of the suggested DNN in this work. That is, GRNN, LSTM, BiLSTM, and dense layers will be used for that aim in the next section. Since the understandings of the layers are discussed in this section, no more discussion of these layers will be given in the consequent sections.

IV. PROPOSED DNN STRUCTURE

The suggested architecture of the proposed DNN composed of sixteen-layers (including input and output layers) as shown in Figure 12. In other words, the structure is composed of sixteen layers, five of which are GRU layers and a bidirectional LSTM layer. Basically, the suggested DNN built about the core layers, that are specified for time-series sequence, such as GRU, LSTM, BiLSTM, and time-distributed dense layers. Due to the time-series nature of the problem undergo, the GRU, LSTM, BiLSTM, and time-distributed dense layers are selected for this design.

The design is sequential and accepted the input sequence of size 32 through the input layer. However, there are five-GRU layers of different sizes and between them another layers. Specifically, after the input layer, the first GRU is followed the input layer with input of period of sequence equals to 32. The output shape of this GRU layer is (32,150), accordingly, the number of trainable parameters is 68850.

The next layer is the dense layer (fully connected layer), which here time-dependent, TimeDistributed-dense layer. When using a TimeDistributed layer, the same layer is used on many inputs at once. And it only needs one output per input to run quickly and accurately. This encapsulation makes it possible to add a layer to each discrete chunk of input time. That is, the input to this layer is the exact output of the previous GRU-layer. Thus, the output of the TimeDistributed-layer is (32,100) since this layer size is 100-points. Hence, the number of trainable parameters of the TimeDistributed-layer is accordingly will be 15100. After that, a dropout layer is employed with dropout value of 0.1.

The second GRU-layer is coming now after the dropout-layer. This second GRU-layer has the same dimension of the previous GRU-layer, but its input size is the main difference which is (32,100), accordingly, the number of trainable parameters will be 113400.

Thus, the output dimension of the second GRU-layer is set to (32,150), which will be the input to the next layer, TimeDistributed-layer of 100-points. This TimeDistributed-layer also has the same dimension of the previous TimeDistributed-layer; hence, the number of trainable parameters will be 15100.

Another dropout-layer is following with dropout value of 0.1. Note that the dropout-layer has no trainable parameters. Now, the third GRU-layer of output shape (32,50) resulting in 22800 trainable parameters. TimeDistributed-layer is used after the third GRU-layer of the same size of the previous TimeDistributed-layers, in other words, there are 5100 trainable parameters, i.e., a smaller number of weightings by 10000, this is due to that the input size is 50 (50-inputs $\times$100-neurons + 100-bias = 5100).

TABLE 2. Layers and related dimensions of the suggested DNN-model.

Sequence	Layer	Input shape	Output shape	Trainable parameters
1	GRU	(32,1)	(32, 150)	68850
2	Dense	(32, 150)	(32, 100)	15100
3	Dropout	(32, 100)	(32, 100)	0
4	GRU	(32, 100)	(32, 150)	113400
5	Dense	(32, 150)	(32, 100)	15100
6	Dropout	(32, 100)	(32, 100)	0
7	GRU	(32, 100)	(32, 50)	22800
8	Dense	(32, 50)	(32, 100)	5100
9	Dropout	(32, 100)	(32, 100)	0
10	BiLSTM	(32, 100)	(32, 200)	160800
12	GRU	(32, 200)	(32, 50)	37800
13	Dropout	(32, 50)	(32, 50)	0
14	GRU	(32, 50)	(1,50)	15300
15	Dropout	(1,50)	(1,50)	0

Next, a dropout-layer is followed with 0.1 dropping value before the employment of the bidirectional LSTM-layer.

Better representations can be learned in BiLSTM with the help of potential future context chunks. However, the dimension of this layer is (32,200) which supports 160800 trainable parameters. Consequently, the fourth GRU-layer is used after the BiLSTM-layer with 50-units, the trainable parameters are 37800 parameters, when its input is 200.

After that, a dropout-layer just like previous dropout-layers, to be followed by the fourth, last, GRU-layer. This last GRU-layer has 50 units and input of 50 as well, that is, the number of trainable parameters in this layer is 15300, then it is followed by the last dropout-layer and finally the output-layer. Summing up the abovementioned trainable parameters results in 454301 overall trainable parameters in the suggested DNN-model, as listed in Table 2. In Table 2, the first (input) layer is not included, but the last (output) layer was included.

V. RESULT AND DISCUSSIONS

First of all, it is essential to initialize some necessary parameters. For instance, Kaggle.com was the system employed to implement the suggested network. Kaggle.com has sufficient capabilities and facilities. In this work, the Graphical Processing Unit (GPU) was employed, which is freely accusable in Kaggle.com, GPU P100, whith 13GB RAM. Python was the programming language to implement the system, where Keras from TensorFlow package was utilized since it is dedicated for DNN and ML algorithms.

On the other hand, the activation of the GRU-layers was *tanh*. The optimizer was Stochastic Gradient Descent (SGD). Learning rate of the SGD-optimizer is 0.001 with momentum

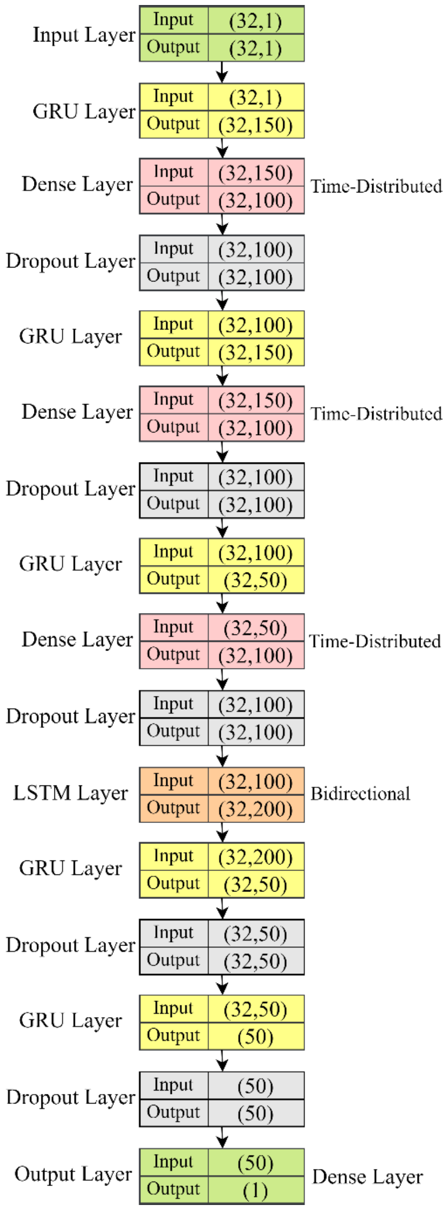

FIGURE 13. Sixteen-layers of the suggested architecture for the proposed DNN.

value of 0.9, while the loss function is MSE (Mean Squared Error) and the decay learning rate was set to 1×10^{-7}.

The dense-layer activation function is ReLU (Rectified Linear Unit). The batch size is 150 and number of whole epochs is 250. Note that the dataset has been split to train and test subsets, where the train subset is 80% and the test subset is 20% of the original dataset. Executing the simulation gives RMSE as 5.56, 5.01, 1.82, and 0.69 for the first, second, third, and fourth junction, respectively.

It is observed that the RMSE of the fourth junction is the smallest magnitude, this is due to that the original dataset was limited for this junction. The first and second junctions seems to be almost comparable to each other, while the third junction has RMSE of 1.82.

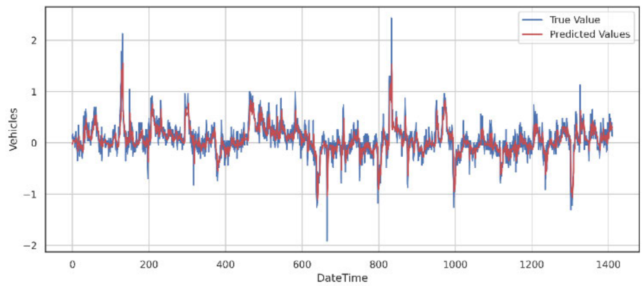

FIGURE 14. First junction transformed, original and predicted, vehicle concentrations.

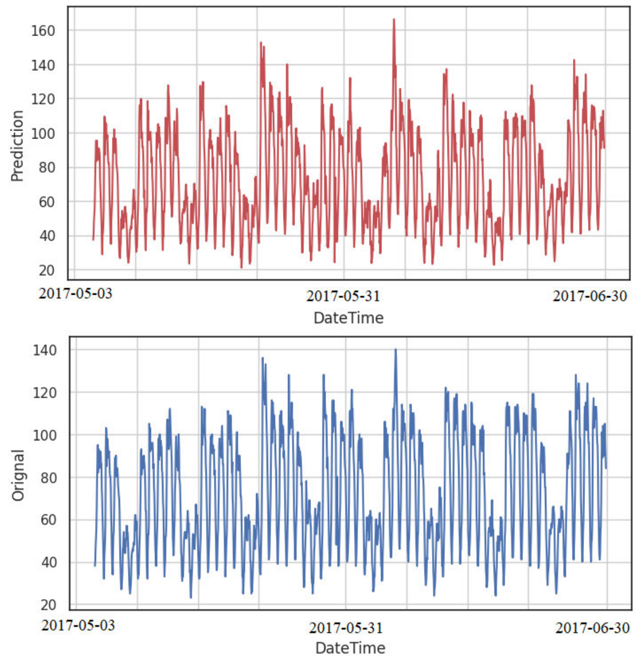

FIGURE 15. First junction true (actual) and predicted values using the suggested DNN-model.

However, these values can be reflected on the prediction of the vehicles compared with the original series, as shown in Figure 14, which shows the vehicles prediction (transformed) in the first junction. It is shown in the prediction of the first junction that there is high degree of match between the two curves, transformed true and transformed predicted values, except at the times where there is high, overshoot, number of vehicles.

For more convenience, Figure 15 explains the real-values of the vehicles with respect to the real-times. It can be seen that the predicted values (upper-part of Figure 15) have maximum number of vehicles more than 160, while the original numbers did not exceed 140 (see lower-part in Figure 15), which is also stated previously in Figure 5.

However, the seasonality behavior is clearly observed in Figure 15 for the predicted actual number of vehicles, which confirms the performance of the suggested DNN-model.

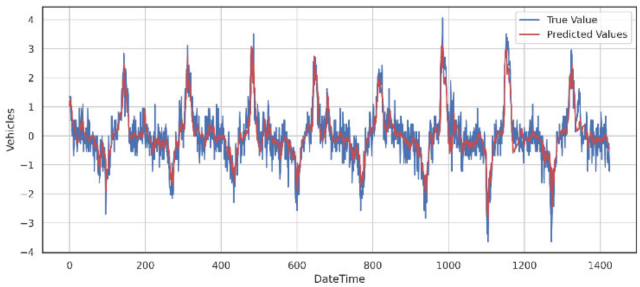

FIGURE 16. Second junction transformed, original and predicted, vehicle concentrations.

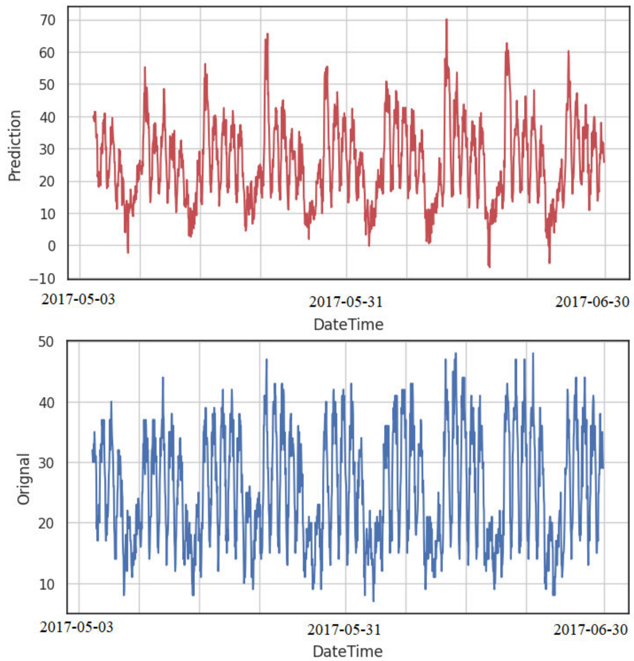

FIGURE 17. Second junction true (actual) and predicted values using the suggested DNN-model.

The same behavior is captured in the second junction, see Figure 16. That is, the transformed predicted values in Figure 16 show that the prediction is accurate where the RMSE is 5.01 which is slightly lower than that of the first junction.

Although the prediction in the second junction in the points of high concentrations, the predicted values in Figure 16, have not fluctuated as the true values. In other words, the blue-curve in Figure 16 did not follow up the exact points of the red-curve, the actual number of vehicles shows the seasonality structure of the second junction series in Figure 17 (upper-part). In Figure 6 as well as in Figure 17 (lower-part), the maximum number of vehicles is almost 50, but the predicted values reached the limit of 70 vehicles.

Furthermore, Figure 18 shows the transformed, original and predicted, vehicles concentrations along the time-series, for the third junction, while Figure 19 shows that information for the fourth junction. However, the transformed (true or original and predicted) values show slight differences in the third junction and specifically in the fourth junction. For instance, in the third junction, there are a small difference in some time-segments in terms of number of vehicles.

Moreover, the fourth junction has much less difference due to the RMSE is lower than those in the three previous

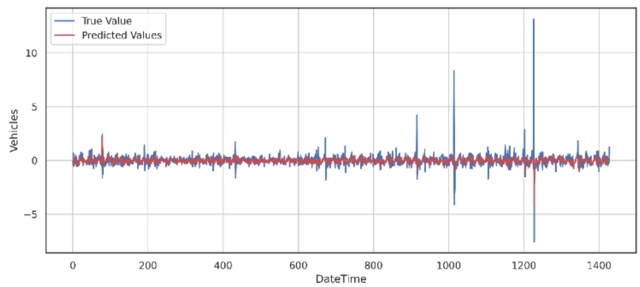

FIGURE 18. Third junction transformed, original and predicted, vehicle concentrations.

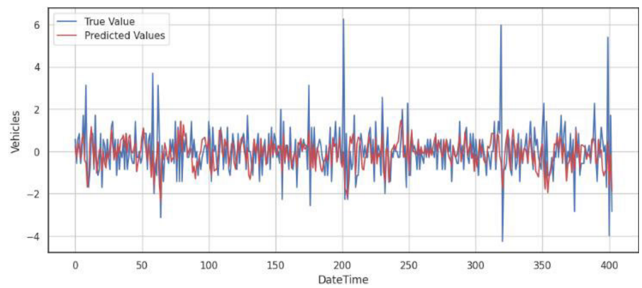

FIGURE 19. Fourth junction transformed, original and predicted, vehicle concentrations.

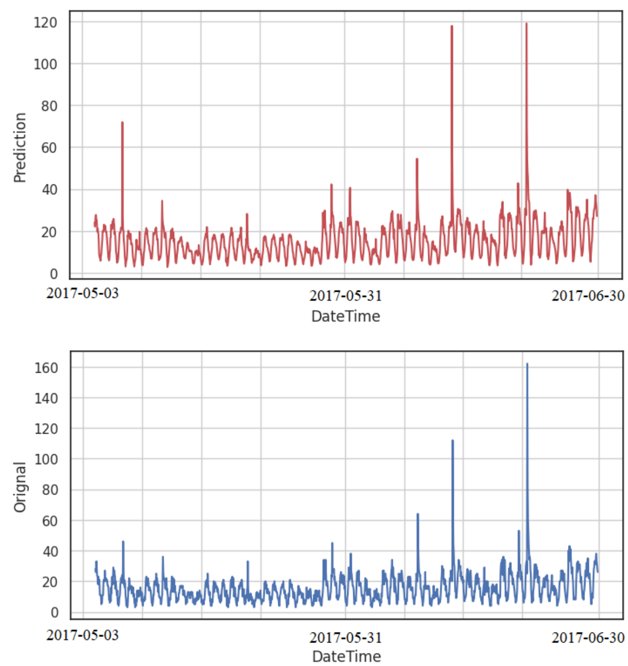

FIGURE 20. Third junction true (actual) and predicted values using the suggested DNN-model.

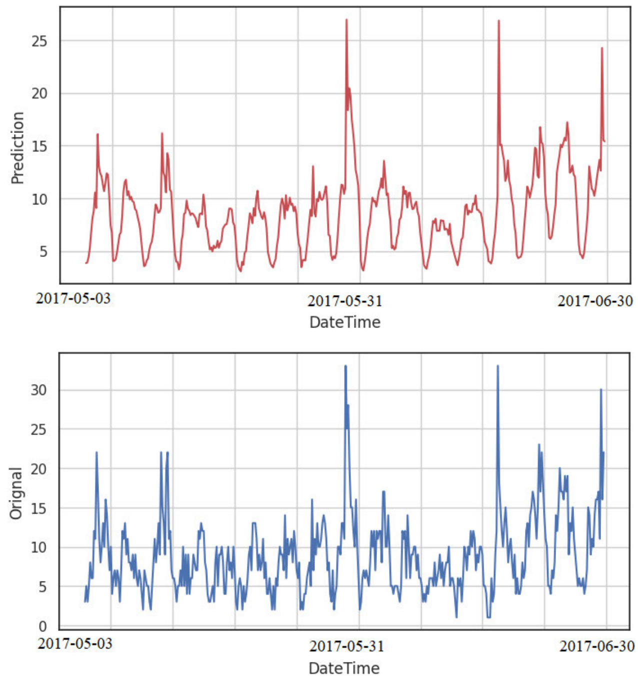

FIGURE 21. Fourth junction true (actual) and predicted values using the suggested DNN-model.

junctions, where the RMSE is 0.69. However, the RMSE in the fourth junction is affected by the limited number of records in the original dataset of the junctions. Nonetheless, Figures 20 and 21 show the actual and predicted number of vehicles in the third and fourth junctions, respectively. These two figures show that the following up of the predicted values is almost match the actual behavior but the number of vehicles is slightly different. For instance, in the last point of the time-series in Figure 21 shows that the actual number of vehicles is almost 30, while the predicted value is 24. However, more vehicles are using Junction 1 than each of the other two interchanges. Given the limited information available, one cannot draw any firm conclusions about junction 4.

In [30], GRU based DNN was implemented with RMSE result of 14.21, 39.99, 27.46, 8.84, and 5.46 in the situations: off-peak, peak, complete dataset, losloop, and Shenzhen Road, respectively. In [27], the RMSE for five sections were 31.847, 29.035, 19.352, 68.392, and 81.394, for the first to fifth sections, respectively in the United Kingdom traffic flow dataset, where the LSTM based DNN was implemented. Moreover, in [33] the RMSE obtained by the proposed GRU based DNN was 23.35. however, the suggested DNN in this article achieved RMSEs as 5.56, 5.01, 1.82, and 0.69 for the four-junctions. In our model, the RMSE is improved with respect to aforementioned models. Although that the suggested model is more complicated than previous models, but this was at the benefits of the improved results of the RMSE and the predicted traffic flows in the four junctions.

VI. CONCLUSION

In recent decades, urban cities designers have become increasingly interested in foreseeing the occurrence of traffic jams. Increases in infrastructure investment worldwide have resulted in a worldwide increase in traffic congestion. Authorities can plan ahead and take preventative measures if they have advance notice of the traffic. Various techniques have been applied to this area thanks to the advent of AI and the accessibility of large datasets for analysis.

The waste of time and money caused by lengthier trips and more frequent stops for gas is just one consequence of traffic congestion in the world's transportation networks. The incorporation of cutting-edge technologies into transportation networks presents both significant academic obstacles and promising opportunities to significantly improve traffic prediction.

With the use of gated recurrent unit layers and bidirectional long short-term memory layers, the authors of this article present a deep learning machine. Thus, deep neural network consisting of fifteen time-dependent layers are constructed to overcome the congestion problems in modern cities. The advantage here goes to deep learning in particular. Because of their ability to evaluate a big dataset, deep learning algorithms have gained in popularity. Yet many different machine learning algorithms have not been put into practice. So, there is still a huge window of opportunity for study in the area of traffic congestion forecasting. But the proposed model has shown encouraging results, and it can be widely implemented in the construction of contemporary urban centers.

Furthermore, more recent and modern datasets can be employed as a future extension to the current work. On the other hand, less complicated networks might be implemented in order to develop modern intersections in modern cities. Such employment of deep neural networks in urban cities will help architectural engineers be more powerful in accomplishing their job. Moreover, other engineering fields, such as communication engineering, will not be far behind in their contribution to putting more facilities in modern urban cities.

REFERENCES

[1] G. Cookson, "Inrix global traffic scorecard," INRIX, 2017. [Online]. Available: https://inrix.com/resources/inrix-2017-global-traffic-scorecard.
[2] *68% of the World Population Projected to Live in Urban Areas by 2050, Says UN*, United Nations Department of Economic and Social Affairs, New York City, NY, USA, 2018.
[3] M. H. Alwan, Y. I. Hammadi, O. A. Mahmood, A. Muthanna, and A. Koucheryavy, "High density sensor networks intrusion detection system for anomaly intruders using the slime mould algorithm," *Electronics*, vol. 11, no. 20, p. 3332, Oct. 2022.
[4] E. H. Salman, M. A. Taher, Y. I. Hammadi, O. A. Mahmood, A. Muthanna, and A. Koucheryavy, "An anomaly intrusion detection for high-density Internet of Things wireless communication network based deep learning algorithms," *Sensors*, vol. 23, no. 1, p. 206, Dec. 2023.
[5] B. Alsolami, R. Mehmood, and A. Albeshri, "Hybrid statistical and machine learning methods for road traffic prediction: A review and tutorial," in *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*, R. Mehmood, S. See, I. Katib, I. Chlamtac, Eds. Cham, Switzerland: Springer, 2020, pp. 115–133.
[6] M. Akhtar and S. Moridpour, "A review of traffic congestion prediction using artificial intelligence," *J. Adv. Transp.*, vol. 2021, Jan. 2021, Art. no. 8878011.

[7] B. Medina-Salgado, E. Sánchez-DelaCruz, P. Pozos-Parra, and J. E. Sierra, "Urban traffic flow prediction techniques: A review," *Sustain. Comput., Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100739.

[8] M. Shaygan, C. Meese, W. Li, X. Zhao, and M. Nejad, "Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities," *Transp. Res. C, Emerg. Technol.*, vol. 145, Dec. 2022, Art. no. 103921.

[9] Q. Hou, J. Leng, G. Ma, W. Liu, and Y. Cheng, "An adaptive hybrid model for short-term urban traffic flow prediction," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121065.

[10] N. Zhang, X. Guan, J. Cao, X. Wang, and H. Wu, "Wavelet-HST: A wavelet-based higher-order spatio-temporal framework for urban traffic speed prediction," *IEEE Access*, vol. 7, pp. 118446–118458, 2019.

[11] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved k-nearest neighbor model for short-term traffic flow prediction," *Proc.-Social Behav. Sci.*, vol. 96, pp. 653–662, Nov. 2013.

[12] H. Yi, H. Jung, and S. Bae, "Deep neural networks for traffic flow prediction," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 328–331.

[13] K.-L. Li, C.-J. Zhai, and J.-M. Xu, "Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 2804–2807.

[14] L. N. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, "An effective spatial–temporal attention based neural network for traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 108, pp. 12–28, Nov. 2019.

[15] Y. Hou, Z. Deng, and H. Cui, "Short-term traffic flow prediction with weather conditions: Based on deep learning algorithms and data fusion," *Complexity*, vol. 2021, Jan. 2021, Art. no. 6662959.

[16] A. I. Salamanis, A.-D. Lipitakis, G. A. Gravvanis, S. Kotsiantis, and D. Anagnostopoulos, "An adaptive cluster-based sparse autoregressive model for large-scale multi-step traffic forecasting," *Exp. Syst. Appl.*, vol. 180, Oct. 2021, Art. no. 115093.

[17] I. O. Olayode, A. Severino, T. Campisi, and L. K. Tartibu, "Prediction of vehicular traffic flow using Levenberg–Marquardt artificial neural network model: Italy road transportation system," *Commun. Sci. Lett. Univ. Zilina*, vol. 24, no. 2, pp. E74–E86, Apr. 2022.

[18] J. Tian, X. Song, P. Tao, and J. Liang, "Pattern-adaptive generative adversarial network with sparse data for traffic state estimation," *Phys. A, Stat. Mech. Appl.*, vol. 608, Dec. 2022, Art. no. 128254.

[19] Y. Wang, R. Jia, F. Dai, and Y. Ye, "Traffic flow prediction method based on seasonal characteristics and SARIMA-NAR model," *Appl. Sci.*, vol. 12, no. 4, p. 2190, Feb. 2022.

[20] B. Fernandes, F. Silva, H. Alaiz-Moretón, P. Novais, C. Analide, and J. Neves, "Traffic flow forecasting on data-scarce environments using ARIMA and LSTM networks," in *New Knowledge in Information Systems and Technologies*. Cham, Switzerland: Springer, 2019, pp. 273–282.

[21] G. Lu, J. Li, J. Chen, A. Chen, J. Gu, and R. Pang, "A long-term highway traffic flow prediction method for holiday," in *Advanced Multimedia and Ubiquitous Engineering* (Lecture Notes in Electrical Engineering). Singapore: Springer, 2019, pp. 153–159.

[22] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 100, pp. 372–385, Mar. 2019.

[23] F. Zhao, G.-Q. Zeng, and K.-D. Lu, "EnLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 101–113, Jan. 2020.

[24] Q. Bing, F. Shen, X. Chen, W. Zhang, Y. Hu, and D. Qu, "A hybrid short-term traffic flow multistep prediction method based on variational mode decomposition and long short-term memory model," *Discrete Dyn. Nature Soc.*, vol. 2021, Oct. 2021, Art. no. 4097149.

[25] J. Wang and L. Li, "Traffic flow velocity prediction based on real data LSTM model," presented at the Vehicle Electrific. Powertrain Diversification Technol. Forum, USA, Jan. 2021, Paper 2021-01-7014. [Online]. Available: https://doi.org/10.4271/2021-01-7014.

[26] M. A. Mondal and Z. Rehena, "Stacked LSTM for short-term traffic flow prediction using multivariate time series dataset," *Arabian J. Sci. Eng.*, vol. 47, no. 8, pp. 10515–10529, Aug. 2022.

[27] R. Shi and L. Du, "Multi-section traffic flow prediction based on MLR-LSTM neural network," *Sensors*, vol. 22, no. 19, p. 7517, Oct. 2022.

[28] D. Chen, H. Wang, and M. Zhong, "A short-term traffic flow prediction model based on AutoEncoder and GRU," in *Proc. 12th Int. Conf. Adv. Comput. Intell. (ICACI)*, Aug. 2020, pp. 550–557.

[29] Y. Zhang, M. Lu, and H. Li, "Urban traffic flow forecast based on FastGCRNN," *J. Adv. Transp.*, vol. 2020, Sep. 2020, Art. no. 8859538.

[30] Y. Zhao, X. Han, and X. Xu, "Traffic flow prediction model based on the combination of improved gated recurrent unit and graph convolutional network," *Frontiers Bioeng. Biotechnol.*, vol. 10, p. 5, Feb. 2022.

[31] M. Ali, K. M. Yusof, B. Wilson, and C. Ziegelmueller, "Traffic speed prediction of high-frequency time series using additively decomposed components as features," *IET Smart Cities*, vol. 4, no. 2, pp. 92–109, Jun. 2022.

[32] C. Chai, C. Ren, C. Yin, H. Xu, Q. Meng, J. Teng, and G. Gao, "A multifeature fusion short-term traffic flow prediction model based on deep learnings," *J. Adv. Transp.*, vol. 2022, May 2022, Art. no. 1702766.

[33] N. Hu, D. Zhang, K. Xie, W. Liang, C. Diao, and K.-C. Li, "Multi-range bidirectional mask graph convolution based GRU networks for traffic prediction," *J. Syst. Archit.*, vol. 133, Dec. 2022, Art. no. 102775.

[34] X. Kong, W. Zhou, G. Shen, W. Zhang, N. Liu, and Y. Yang, "Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data," *Knowl.-Based Syst.*, vol. 261, Feb. 2023, Art. no. 110188.

[35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[36] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1–15.

ABDUL HUSSAIN ALI HUSSAIN was born in Diyala, Iraq, in 1959. He received the master's degree in architecture from Ion Mincu University, Bucharest, Romania, in 1993, and the Ph.D. degree in architecture and city planning from Ion Mincu University, in 2015. He was with several engineering offices as an engineer and an architectural designer, from 1993 to 2014. Currently, he is a Senior Lecturer with the Department of Architecture, College of Engineering, University of Diyala. His research interests include architectural designs, city planning, and project management.

MONTADAR ABAS TAHER (Senior Member, IEEE) was born in Baghdad, Iraq, in 1976. He received the B.Sc. degree in electronics and communications engineering and the M.Sc. degree in satellite engineering from Al-Nahrain University, Iraq, in 2000 and 2003, respectively, and the Ph.D. degree from the National University of Malaysia, in 2015. He was with Motorola, as a Communications Engineer, from 2005 to 2009. During this period, he became the Team Leader of the BSS Department. He has been a Lecturer with the University of Diyala, Iraq, since 2010, where he has been a Senior Lecturer with the Department of Communications Engineering, since 2015. He is currently an Associate Professor with the Department of Communications Engineering and the Chief of the Space Navigation and Control Research Laboratory. His main research interests include multicarrier communication systems, DSP for telecommunications, 4G, and 5G. His current interests include artificial intelligent and deep learning systems. He is a reviewer of some international journals.

OMAR ABDULKAREEM MAHMOOD was born in Diyala, Iraq. He received the B.Sc. degree in communications engineering from the University of Diyala, Iraq, in 2006, and the M.Sc. degree in communications engineering and the Ph.D. degree in electronic science and wireless communication from The Bonch-Bruevich Saint Petersburg State University of Telecommunications, in 2013 and 2021, respectively. He has been a Lecturer with the University of Diyala, since 2007, where he is currently a Lecturer and the Chief of communication systems with the Department of Communications Engineering. His main research interests include wireless communication systems, data traffic in the Internet of Things, 4G, 5G, artificial intelligent, and deep learning systems. He is a reviewer of some international journals.

YOUSIF I. HAMMADI was born in Iraq, in 1988. He received the B.Sc. degree in communication engineering from the University of Diyala, Iraq, in 2011, the M.Sc. degree in electronic and communication engineering from Cankaya University, Turkey, in 2013, and the Ph.D. degree in laser/electronic and communication engineering from the University of Baghdad, Iraq, in 2019. He is a Lecturer with Bilad Alrafidain University College, Diyala, Iraq. His research interests include digital communications, optical communication, optical fiber, and laser generation.

REEM ALKANHEL (Member, IEEE) received the B.S. degree in computer sciences from King Saud University, Riyadh, Saudi Arabia, in 1996, the M.S. degree in information technology (computer networks and information security) from the Queensland University of Technology, Brisbane, Australia, in 2007, and the Ph.D. degree in information technology (networks and communication systems) from Plymouth University, Plymouth, U.K., in 2019. She has been with Princess Nourah Bint Abdulrahman University, Riyadh, since 1997, where she is currently an Assistant Professor with the College of Computer and Information Sciences. Her current research interests include communication systems, networking, the Internet of Things, software-defined networking, and information security.

AMMAR MUTHANNA (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the Saint-Petersburg State University of Telecommunications, in 2009, 2011, and 2016, respectively. From 2017 to 2019, he was a Postdoctoral Researcher with RUDN University. He took part in the Erasmus Student Program, Faculty of Electrical Engineering, University of Ljubljana, from 2012 to 2013. He was a Visiting Researcher with Tampere University, Finland, in 2014. He has been a Professor with the Bonch-Bruevich Saint Petersburg State University of Telecommunications (SUT), since 2019. His research interests include wireless communications, 5G/6G cellular systems, the IoT applications, edge computing, and software-defined networking. He has been an active member of the technical program committee on many international conferences and journals. He is an Expert at the Judges Panel and the Challenge Management Board at AI-5G-Challenge, ITU, and a Russian host organizer.

ANDREY KOUCHERYAVY received the degree from the Leningrad University of Telecommunications, in 1974. He joined the Telecommunication Research Institute LONIIS, where he worked until October 2003 (from 1986 to 2003 as the First Deputy Director). He has been a Professor with the Bonch-Bruevich Saint Petersburg State University of Telecommunications (SUT), since 1998, where he became a Chaired Professor with the Telecommunication Networks and Data Transmission Department, in 2011. He was an Advisor of the Central Science Research Telecommunication Institute (ZNIIS), from 2003 to 2010. He was a Co-Founder of the International Teletraffic Seminar, in 1993, 1995, 1998, and 2002; the Founder of the Model Network for Digital Networks with LONIIS, in 1997; and the Co-Founder of the Model Network for Packet Networks with ZNIIS, in 2004, the Internet of Things Laboratory, in 2012, and the Quality of Experience and IPTV Laboratory with SUT, in 2014. He was the Chair of the Scientific School on Teletraffic Theory in LONIIS, from 1990 to 2003; the Founder and the Scientific School Chair of the Internet of Things and Self-Organizing Networks with SUT, since 2010; a Steering Committee Member of IEEE technically co-sponsored series of conferences ICACT and NEW2AN. He was the SG11 ITU-T Vice-Chairperson, from 2005 to 2008 and from 2009 to 2012. He was the WP3/WP4 SG11 Chairperson, from 2006 to 2012; the WP4 SG11 Vice-Chairperson, from 2015 to 2016; and the Chairperson of SG11, from 2017 to March 2022. He was the Co-Founder of the International Testing Center for New Telecommunications Technologies with ZNIIS under ITU-D Competence. He was a Host and a Technical Program Committees Member of the ''Kaleidoscope 2014'' at SUT. He was an Honorary Member of Popov's Society, in 2002.