IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2021, accepted July 10, 2021, date of publication July 14, 2021, date of current version July 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097141

Intelligent Traffic Flow Prediction Using Optimized GRU Model

BASHARAT HUSSAIN[1], MUHAMMAD KHALIL AFZAL[1], (Senior Member, IEEE), SHAFIQ AHMAD[2], AND ALMETWALLY M. MOSTAFA[3]

[1]Department of Computer Science, COMSATS University Islamabad at Wah, Wah Cantt 47040, Pakistan
[2]Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11421, Saudi Arabia

Corresponding author: Muhammad Khalil Afzal (khalilafzal@ciitwah.edu.pk)

This work was supported by the Deanship of Scientific Research at King Saud University through the research group under Grant RG-1441-425.

ABSTRACT Facilitating citizens with accurate traffic flow prediction increases the quality of life. Roadside sensors and devices are used to capture live streams of huge data and the Internet of Things (IoT) is becoming popular for the deployment of effective Intelligent Transportation Systems (ITS). Traffic flow prediction from the live datastreams require building a data-driven model. This is a challenging task and has attracted researchers for better interpretation of the traffic characteristics. The core problem in traffic prediction is modeling a diversity of traffic trends and unpredictable flow variations with temporal dependencies. Initially, statistical and shallow neural network models were applied to some extent. Recently, deep learning has come up with proven and promising outcomes. Gated Recurrent Unit (GRU) is a variation of recurrent neural networks used effectively for traffic flow prediction. Like other deep networks, GRU uses hyperparameters and a sliding window time-steps mechanism to prepare and tune the model. Better tuning for hyperparameters and search for optimal window size is a tedious process. In this research work, we present an algorithm for hyperparameters tuning along with sliding window steps optimization. Results obtained on a real-time public traffic dataset show a higher capability of the proposed method to reduce the error and an average gain of the optimized model over the untuned network is 4.5%. Furthermore, we apply the optimal hyperparameters obtained in the experiment to other deep learning models and present that our approach improves prediction accuracy and stability.

INDEX TERMS Traffic flow prediction, deep learning, gated recurrent unit, hyperparameters optimization.

I. INTRODUCTION

Intelligent Transportation System (ITS) is an essential element of smart cities. ITS not only provides real time traffic characteristics but also predicts short-term traffic flow. Traffic prediction is useful for route diversions and congestion control. However, due to the stochastic nature of transportation data, time-series prediction based on huge and real-time data provided by road sensors is a challenging task. These challenges are noticed in data representation, model validation, optimal prediction framework building, estimation time-lag ahead, and the effect of external factors on traffic models. Various techniques are proposed by scientists to validate traffic predictions [1]-[3]. Classic mathematical and statistical approaches mainly apply linear models and shallow machine learning techniques to estimate traffic characteristics. Those are not accurate because of unpredictable traffic flow variations which are essentially nonlinear and random in nature.

It is trivial to anticipate the usual traffic congestion ahead of time. For example, it is easy to anticipate that the traffic flow is generally higher during day timings as compared to midnight intervals. However, it requires significant effort to predict the traffic flow to a given degree of accuracy in a given time interval. Such statistical models use the structured expressions which are predetermined and make assumptions about the variables. These assumptions sometimes turn out to be correct resulting in correct prediction, and sometimes perform badly. The core issue lies with the stronger assumptions which are incompatible with unpredictable and highly variant

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

traffic data. The data-driven deep neural models are popular to predict traffic flows. Recurrent Neural Networks (RNN) belong to a type of deep learning model that can memorize historical input data and use it to improve prediction accuracy. RNN can remember historical features in temporal sequential data. However, it leads to a famous vanishing gradient issue and loses the remembrance of longer sequences over time [4]. To overcome this issue, researchers suggested variations of RNN. Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two widely used variations of the recurrent network. A GRU model was originally proposed by Cho *et al.* [5] that aims to solve the vanishing gradient problem which appears in a standard recurrent neural network.

We aim to fine-tune the existing GRU based deep neural networks to increase their accuracy of traffic flow predictions. These tuned sets of configuration values for the neural network are called hyperparameters. In this paper, we search these hyperparameters for GRU deep neural network and find the values of these hyperparameters to improve its prediction accuracy. Our proposed method improves hyperparameters and sliding window steps and further evaluated the prediction accuracy and stability. We also generalize the search results of this approach, applying the same hyperparameters to various neural network architectures by replacing the model and perform analysis, for example, as discussed in [6]. The main contributions of this paper are described below:

1) We propose a search mechanism for recurrent neural network that results in obtaining network hyperparameters, optimized number of sliding window steps ahead and improve the prediction accuracy by reducing the error in traffic flow prediction.
2) Once refined time-lags and hyper parameter set is obtained by applying the proposed mechanism, this study further evaluates and compares the accuracy of GRU with various deep learning techniques, for example, bidirectional, stacked and convolutional variants of LSTM for traffic prediction.
3) Experimental Result applied on a public traffic dataset shows an improved accuracy with the proposed method by applying a search mechanism to fine-tune the neural network models.

The rest of the paper is organized as follows. Section II states the related background of the models for traffic flow prediction; Section III presents the proposed model for GRU network tuning and describes deep learning models used for traffic prediction. The experimental design on a real-time traffic database and experimental results are given in Section IV and finally, the conclusion and future work are described in Section V.

II. RELATED BACKGROUND

Traffic flow prediction models are generally divided into two broad classifications i.e., parametric models and non-parametric models [7].

Parametric models refer to the models where the structured expression is predetermined making the assumptions about the variable [8]. A parametric algorithm may perform better if the assumptions turn out to be correct, otherwise, it performs badly. Among parametric models the linear regression, Auto-regressive Integrated Moving Average (ARIMA) model [9], Kalman Filtering (KF) model, and maximum likelihood estimate [10] are discussed. Although the parametric models were applied in the past for traffic characteristics description, they are inflexible. Moreover, due to the non-linearity and variations in traffic data, they do not predict the traffic flow effectively [11]. This is by virtue of Artificial Intelligence (AI) approaches that they outperform the classical statistical models and the relevant modeling constraints due to the superior capability of mining information from messy and multi-dimensional traffic data.

To mitigate these issues, scientists have researched prediction models. These are the non-parametric models that use a flexible number of parameters. These parameters often grow in numbers as the model learns from more data. Therefore, the non-parametric algorithms are computationally slower. On the other hand, the non-parametric models make fewer assumptions about the data. The protocols falling under this category are; Support Vector Machine (SVM) [13], K-nearest Neighbor (KNN) [14], and Neural Networks [15], and many more.

More recently, deep learning is introduced to traffic prediction and is widely accepted as a better approach for describing traffic systems. In contrast to traditional shallow structures, deep learning models use multi-layer nonlinear structures to describe distributed and hierarchical features for more complex traffic flow data. Some researchers have proposed clustering approaches (such as Deep Belief Networks (DBN) for traffic flow prediction [16]. Li *et al.* [17] proposed an intelligent swarm-based model to optimize parameters in DBN and enhance its multiple steps ahead prediction capability. Reference [18] proposed an attention based model like Stacked Auto-Encoder (SAE) which outperformed Random Walk (RW), Feed-Forward Neural Networks (FFNN), and Radial Basis Function (RBF). These models are dense where every neuron of the previously hidden layer is connected to each neuron of the next layer. There are no assumptions about the features in the fully-connected neural network models. Thus, fully-connected networks extract features/ characteristics from the dataset automatically [19].

It is observed that RNN and its variants are good in capturing the stochastic and nonlinear variations over time in traffic data. However, RNN leads to a vanishing gradient problem and doesn't predict longer sequences due to short memory remembrance. Further, RNN does not deal with special features where the Convolutional Neural Network (CNN) outperforms. Also, with an increase in time intervals, RNN tends to lose the ability to remember far feature information. Moreover, as training the RNN model requires a predetermined size of sliding window data, it is difficult to obtain

TABLE 1. Systematic literature review table: a comparison of various models used for data-driven traffic-flow prediction. Each row is specified with the objective. domain, method and measurement metrics used. Abbreviations: Pa, parametric model; NPa, non-parametric model; DL, deep learning model; ARIMA, autoregressive integrated moving average; VLSTM, vanilla long short-term memory; BLSTM, bidirectional LSTM; CLSTM, a convolutional LSTM network.

	Domain			Learning Method						
Objective	*Pa*	*NPa*	*DL*	*KNN*	*ARIMA*	*LSTM*	*GRU*	*DBN*	*BLSTM*	*CLSTM*
Predicts traffic flow using statistical approximation for short-term free-way traffic prediction [9]	✓				✓					
Using clustering approaches for traffic flow prediction [15]		✓		✓						
LSTM and GRU applied to short-term traffic flow predic-tion and results compared with SAE, FFNN, SVM [21]			✓			✓	✓			
Discusses the role of external factors intraffic flow predictions [22]			✓			✓		✓		
The impact of temporal features on traf-fic prediction has been discussed [23]			✓			✓				
Argues that the GRU mixes cellular sta-tus and hidden status, and model is sim-pler than the LSTM, still with stronger implications [28]			✓			✓	✓			
Optimization and Prediction of traffic flow characteristics by progressivelytuning GRU neural network model (This work)			✓		✓	✓	✓		✓	✓

the optimal parameter size of the window automatically in practice.

A GRU architecture was proposed as a variation of RNN by Cho *et al.* [5]. GRU aims to solve the vanishing gradient problem. GRU were successfully applied to the transportation problem, specially traffic-flow estimation a few steps ahead and results showed that the models are superior to SAE, FFNN and SVM [21]. GRU model is very similar to LSTM and both can produce equally excellent results. They can confront the vanishing gradient and over-fitting regarded as well-known problems with the traditional RNNs. GRU's internal structure is simpler and more rapid to train than LSTM. It has two control gates (reset and update gate) to overcome the vanishing gradient problem in RNN. Consequently, GRU requires lesser computations steps to update its latent state. The update gate helps to determine the past information from the time series that need to pass to the future. Keeping such information helps eliminate the risk of vanishing explosion. Reset gate, however, determines to decide how much of the past information to forget.

Researchers proposed that external factors can play a significant role in traffic flow predictions. Some examples of external factors are; weather conditions, changing populations, and social-economic events. Though traffic flows possess identical temporal patterns during day and night intervals, external factors may severely influence these patterns.

Mou *et al.* [23] argued that existing LSTM studies lack the impact of temporal features to be exploited for traffic prediction. They proposed a temporal information enhancing variant of LSTM (T-LSTM) to forecast the traffic speed of a single road section. However, there is a special and temporal correlation that exits in time series traffic flow data, when captured through the Roadside Units (RSU), which transmit real-time measurements to the ITS system. T-LSTM, extending only temporal dependency lacks spatial dependency. We always need a time-efficient prediction model especially during the rush hours when the traffic conditions change rapidly.

Some authors applied SVM to find the spatio-temporal correlation for traffic flow prediction. Reference [32] manages Adaptive Multi-Kernel SVM with Spatial–Temporal Correlation (AMSVM-STC) for short-term traffic prediction. It attempts to find the nonlinearity and randomness of the traffic flow at a Point of Interest (POI) mainly using probabilistic models. The author claims its model outperforms existing models but did not include the comparisons with deep learning models. Reference [33] proposed a STANN model, inspired by the idea of attention-based models, the spatial

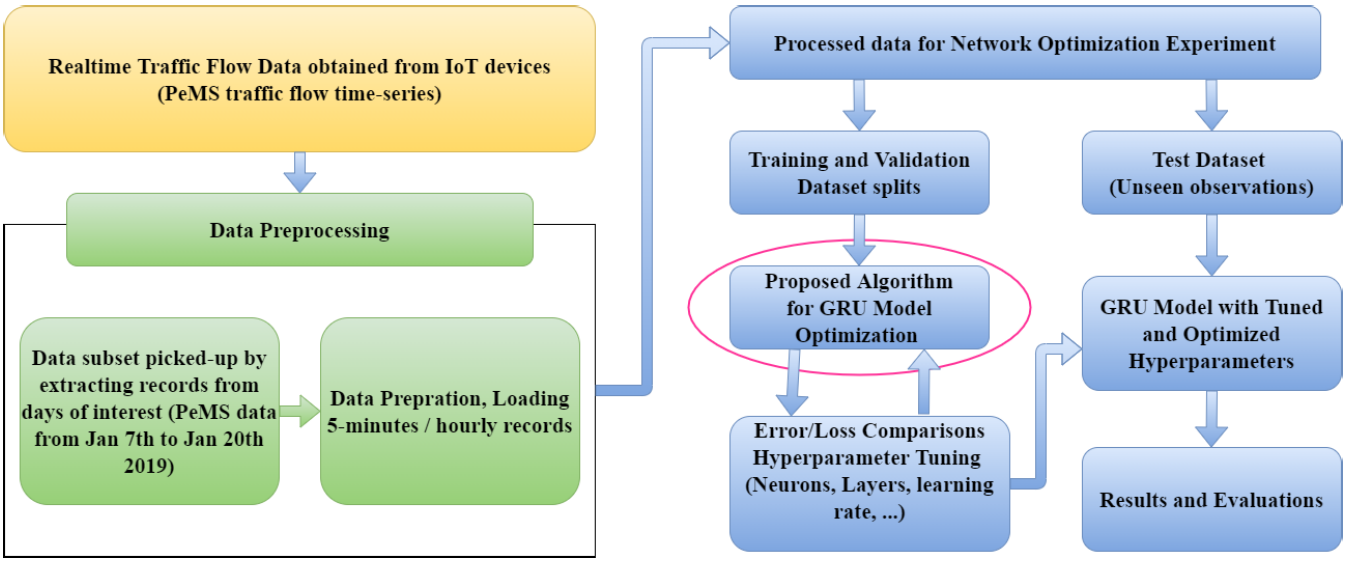

FIGURE 1. Traffic flow training & validation optimization framework for traffic flow detection with GRU neural network.

and temporal attentions are proposed to be used for traffic flow prediction. The model is providing promising results for traffic flow prediction.

As already mentioned, CNN is another popular traffic flow prediction model that uses strong spatial features modeling ability [24]. However, it lags in mapping temporal features. Since RNN is better at exploiting temporal features, some researchers have proposed mixed CNN with RNN to decrease prediction error. The CNN-RNN model uses historical data for temporal characteristics and imaging data to map special characteristics. It then exploits spatio-temporal features of the input sequence [19]. We included the CNN-LSTM model in our research work to compare the performance and accuracy with GRU after parameters tuning.

Kong et. al described the emerging aspect of real-time traffic flow data as big data [25]. The intelligent transportation system (ITS) is gradually becoming big data-driven system with increased heterogeneity, volume, and complexity of IoT data samples [20], [21]. ITS prime goal is to build an advanced, well-informed, and smarter use of transportation network systems for citizens and authorities. A directive of the European Union defined ITS as systems in which information and communication technologies are applied in the field of road transport, including infrastructure, vehicles, and users, and in traffic management and mobility management, as well as for interfaces with other modes of transport [34]. ITS technologies are being adopted widely due to two reasons: to increase the capacity of busy roads and reduce journey times [35].

Our work utilizes GRU for determination of optimal time-lags and hyperparameters values for traffic flow prediction. GRU has stronger applicability, like LSTM, with a simpler network configuration model [22]. Our approach also solves the already mentioned problems with RNN. From the literature survey, we concluded that:

1) GRU is applied for the first time for traffic prediction in 2016 [31] and observed that GRU performs a little better than LSTM by converging faster.
2) The motivation observed for exploiting GRU neural network: "If trained carefully with the right parameters/ hyperparameters, the prediction will be significantly accurate in complex scenarios".

We presented a systematic survey in Table. 1 to compare the efforts that evolve with various relevant models for traffic prediction. We are presenting an effort to mention the core objective of articles with metrics, domain and approach used. Initially, statistical models are mentioned for traffic flow prediction. Later deep learning models are explored which are promising and still explored deeper describing the stochastic traffic characteristics in a better way.

III. METHODOLOGY

In this section, we will introduce the GRU model and present the mechanism to tune its hyperparameters by describing the proposed algorithm. In this section, we first describe the time-series prediction problem formulation followed by an overview of the GRU network. Then, we will detail the model to search for optimum values in a defined universe of discourse. Furthermore, we apply our search mechanism to optimize the GRU network hyperparameters and results are obtained on PeMS database for better prediction of results after executing the training mechanism.

A. TIME-SERIES PREDICTION PROBLEM

Given an input sequence $X = \{x_1, x_2, \ldots, x_k\}$ as original data, where $x_i \epsilon R^d$. We choose a sliding window of length L to define characteristics sequences from original sequence X such that $X = (X_1, X_2, \ldots, X_k)$ where $X_i = (x_i, x_{i+1}, \ldots, x_{i+L})$, where each $X_p \epsilon X$, R^L. The historical or

ground values are given by $y = (y_1, y_2, \ldots, y_{k-1})$ where $y_p \epsilon R^1$. Alternatively, we can denote it by $y = \{y_i : i \epsilon T\}$ where T is called the index set. Our goal here is to predict the next value denoted by $\hat{y^T}$. We learn a nonlinear function f by mapping the temporal sequence feature X and its corresponding ground-truth value y to obtain the estimated value $\hat{y^T}$ with the following formulation:

$$\hat{y^T} = f(X, y) \tag{1}$$

where f is the nonlinear mapping function and learning f our goal using GRU Neural Network model in this work.

B. THE FRAMEWORK

The proposed neural network optimization framework is defined below in Fig. 1. Firstly, we draw traffic flow data from PeMS database (described in data set section in more details). PeMS contains Internet Of Things (IoT) based time sequences of traffic data recorded by the road sensors installed across the highways. Dataset is processed to obtain univariate traffic flow sequences. Original data is real-time observations of average traffic flow obtained every 5-minutes interval. Data is further cleaned up by obtaining hour level details for simplification and the idea is to reduce the processing costs. Training is then preformed using proposed algorithm for GRU optimization, Network tuning is performed in order to obtain better set in each iteration and after getting better hyperparameters we performed prediction using defined set of performance metrics to compare the accuracy.

C. THE ALGORITHM

We start by data sequence denoted by $X = F_{d07-20}$ which represents traffic flow time-series data obtained from PeMS website ranges from $07^{th}Jan - 20^{th}Jan$, 2019 for freeway 9, district 10, California, USA. L denotes the number of steps of sliding windows, η is network learning rate, X_t and X_v denotes the training and validation sets over X. Traffic flow for Training and Validation sets based Optimization algorithm requires lower error rate. An algorithm is proposed to fulfill dual objective, that is, tune the hyperparameters (learning rate) and search for the sliding window time-step for better validation and prediction. The input to the algorithm is an original traffic flow time-series sequences; a set of sliding window of length L, Learning Rate η from a finite set. We initialize various seed values for # of neurons of input layer, activation function, batch window size, # of epochs, optimizer function and loss functions for GRU network. Our goal is to obtain an optimized set of neural network parameters and hyperparameters.

D. THE PARAMETER SETTINGS

There are various neural network parameters and hyperparameters. We have defined finite set of values for each parameter for basic GRU model. Important network parameters are: the number of time-lag window steps L and the size of input neurons and the batch-size b in training process. The size of hidden neurons for each layers in GRU are kept fix to 64 and 32 respectively in this search experiment. Preview of the range of values we choose in this experiment for sliding window steps (L), number of neurons (N), and learning rate (η), batch size (b), epochs (E) and other parameters are specified in Table. 2.

Algorithm 1 Traffic Flow for Training and Validation Sets and Optimization Algorithm for Tuning *GRU* Network Parameters

Input: Original traffic flow time series F_{d07-20}; a set of sliding window of length L, Learning Rate η from finite set.
Output: Optimized *GRU* model M_t according to validation set X_v, Training error ϵ_t, Validation error ϵ_v.

Initialization: Initialize random seed values for # of neurons, activation function, batch window size, # of epochs, optimizer and loss functions for GRU_{net}.

Given: A range for sliding window $L \leftarrow range(2, L_{max})$, and $\eta \epsilon \{\eta_1, \eta_2, \ldots, \eta_n\}$, error threshold $\epsilon_{min} = CONSTANT$

split X into training set X_t and Validation set X_v
set $X_t \leftarrow$ reshuffle (X_t)
set $X_v \leftarrow$ reshuffle (X_v)
set $flag \leftarrow false$
for each window time-step $L \epsilon \{2, 3, \ldots, L_{max}\}$; **do**
  for each value of learning rate $\eta \epsilon \{\eta_1, \eta_2, \ldots, \eta_n\}$ denoted by $iteration_\eta$; **do**

    assert: $L \geq 2$ and $L_{max} \ll length(F_{d01-31})$
    set $M_t \leftarrow GRU_{net}(X_t^L, X_v^L, seeds, \eta)$
    calculate $\epsilon_t^L \leftarrow$ trainingloss (M_t)
    calculate $\epsilon_v^L \leftarrow$ validationloss (M_t)
    if $\epsilon_t^L \leq \epsilon_{min}$ and $\epsilon_v^L \leq \epsilon_{min}$ **then**
      $flag \leftarrow true$
      break loops
    else
      repeat until convergence and set $flag \leftarrow true$
    end if
  end for
end for
if $flag \leftarrow true$ **then**
  Proceed with model M_t for GRU_{net} prediction
else
  repeat experiment with diverse seeds or end the process.
end if

IV. EXPERIMENTAL RESULTS

After data is picked up from real-time traffic flow time series dataset, we used python environment with Anaconda installed on the Windows 10, 64 bit machine. Experiment required scikit-learn, Pandas, NumPy and Matplotlib extension libraries installed.

TABLE 2. List of hyperparameters & values for GRU optimization.

GRU hyperparameters applied in this text		
Symbol	**Parameter**	**Values**
N	# of Neurons	[64,128, 256, 512, 1024]
η	Learning Rate	[.01, .05, .1, .2, .5, .8, 1.0]
L	Sliding Window Steps	[3, 6, 12, 18, 24]
	Optimizer	['Adam']
	Loss	['MSE']
BS	Batch Size (b)	[16, 32, 64, 128, 256]
E	Epochs	[500, 1000]
	Activation Function	['RELU']
	Dropouts	N/A
	Weight Regularizer	N/A
	Kernel Initializer	['Normal']
	Last Activation	['Sigmoid']

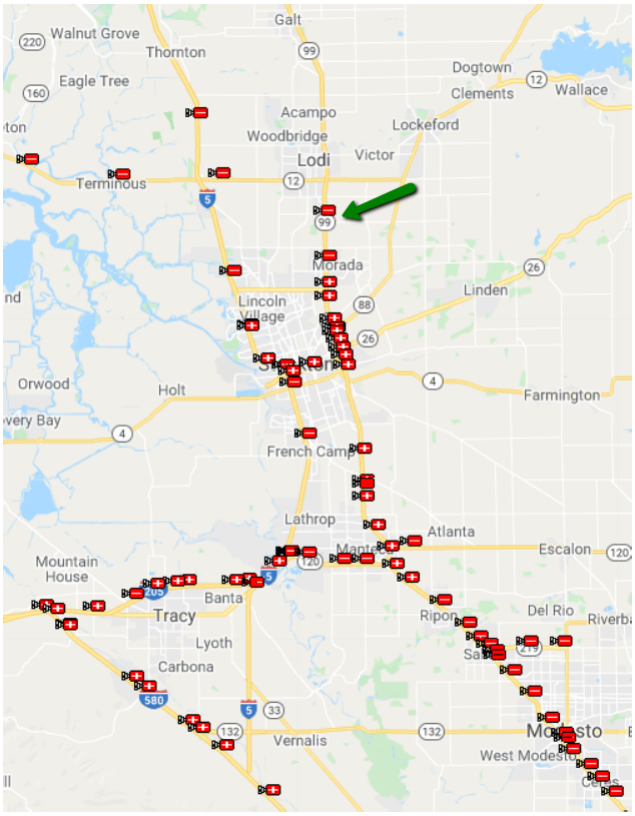

FIGURE 2. Road segment sensors at Freeway #99, District 10, California, USA. Source [31].

A. DATASET DESCRIPTION

Experimental data is obtained from Caltrans Performance Measurements Systems (PeMS) dataset, which offers historical database of traffic flow in California. Data can be downloaded from the California Department of Transportation PeMS website [29], The data describes the traffic flow of different car lanes of San Francisco bay area freeways. There are more than 39000 individual sensor detectors spanning the freeway system across all major metropolitan areas of the State of California and its dataset is widely used for researcher to develop and evaluate traffic models.

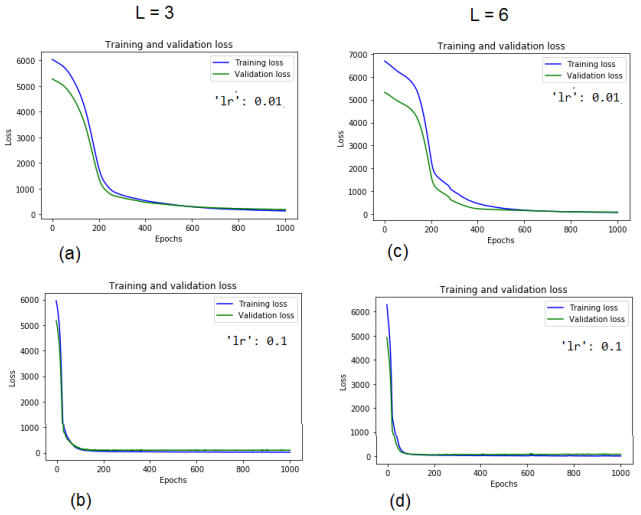

FIGURE 3. Search experiment for GRU network parameter optimization and iteration loss graph: settings with activation='relu', batch size=64, dropout=0, epochs=1000, first neuron=256, hidden layers=1, kernel initializer='normal', last activation='Sigmoid', loss='mse', 'optimizer': 'Adam', weight regularizer=None, window steps and learning rate pairs are specified as: (a) $L = 3$, $\eta = 0.01$, (b) $L = 3$, $\eta = 0.1$, (c) $L = 6$, $\eta = 0.01$, (d) $L = 6$, $\eta = 0.1$.

The collected traffic data is a 5-min interval for traffic flow measures of District 10 in the first whole month of 2019. The collected traffic flow data is divided into two parts, i.e., training data and test data. The test data is used to predict traffic flow and evaluate the model's accuracy. List of hyperparameters and their values for GRU optimization experiments are defined in Table. 2.

B. PROCESSING OF TRAFFIC SERIES

In this study, we trust in the data directly obtained in real-time from PeMS dataset and we draw the data set for evaluation. We observed no missing points for given time period for the whole month of January 2019. Traffic flow data as time-series captured for 13 consecutive days, the task is to predict the traffic flow representing number of vehicles after every 5 minutes slot using GRU and other neural networks after obtaining tuned hyperparameters. The traffic flow data considered for the District 10 and Freeway 99 and as mentioned ranges precisely from January 7, 2019 to January 20, 2019 for experiment purpose. We split the dataset into training, validation and testing subsets with the ratio of 80/20/20 respectively in the work.

C. ALGORITHM EXECUTION

We can load the dataset using Python Library. Once data is ready for further processing, we defined various settings to optimize the search for parameters and hyperparameters of the network. Important network parameters are: the number of sliding window steps L and the size of input neurons, hidden units for each layers in GRU are set to 64 and 32 respectively. To demonstrate the better performance of the model, we conducted a grid like search mechanism defined in Algorithm 1 over Number

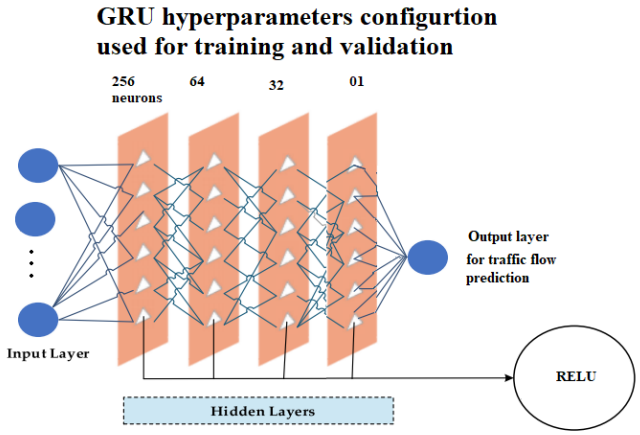

FIGURE 4. A proposed GRU network configuration used to train the model.

TABLE 3. Experiment performed to optimize the learning rates: GRU parameter optimization. Experiment with epochs = 1000, batch size= 64, first layer neurons = 256, optimization = Adam, and window-length = 6 and performed the search for learning rate η = {0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 1.0}. Learning rate 0.2 performed better.

#	η	TE	VE	$RMSE$	$MAPE$	MAE
1	0.01	47.81	157.36	9.96	11.96	6.61
2	0.05	5.01	100.96	7.69	8.21	4.43
3	0.1	15.24	107.25	7.86	9.45	4.84
4	**0.2**	**1.17**	**122.14**	**7.96**	**6.74**	**4.04**
6	0.5	17.04	111.64	8.56	10.23	5.76
7	0.8	0.07	131.43	8.71	7.08	4.12
8	1	0.89	133.61	8.77	8.01	4.56

of neurons in first layer $N\epsilon\{64, 128, 256, 512, 1024\}$, sliding-window steps $L\epsilon\{3, 6, 12, 18, 24\}$, and learning rate $\eta\epsilon\{0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 1.0\}$.

D. MODEL CONFIGURATIONS

We used Python computation library for numerical analysis, to build GRU neutral network models and evaluate the single-step time-series traffic flow prediction performance. We further used Talos python library to preform initial hyperparameters tuning for GRU model optimization along with our proposed algorithm for tuning window time-steps. The GRU network is built by four dense layers. Second, third and fourth layers are kept fixed with 64, 32 and 1 neurons respectively. A proposed GRU network configuration used to train the model is presented in Fig. 4. The diagram is built on an idea from [36]. Preview of the range of values we choose in this experiment for the network parameters are specified in Table. 2. Proposed algorithm is applied to obtain a superior set of learning rate, sliding window length, number of neurons and optimization technique.

We conducted five experiments such that we start with GRU training & validation optimization framework for PeMS traffic data ranges from 07-Jan-2019 to 20-Jan-2019 at Freeway 99, District10, California. In the first step, we performed an open search keeping parameters bounded to subset values to save processing time. GRU configurations

TABLE 4. Experiment performed to evaluate windows steps: GRU parameter optimization. Experiment with epochs = 1000, batch size= 64, first layer neurons= 256, optimization = Adam, and learning rate = 0.2 and performed the search for window length= {3, 6, 12, 18, 24}. Window-length 12 performed better.

#	L	TE	VE	$RMSE$	$MAPE$	MAE
1	3	23.12	100.42	9.13	10.10	5.68
2	6	0.12	133.40	8.11	6.75	4.10
3	**12**	**0.71**	**75.51**	**6.99**	**6.52**	**3.56**
4	18	9.44	82.05	6.29	8.53	3.99
5	24	0.07	114.59	7.28	6.67	3.35

TABLE 5. Experiment #3 performed to evaluate optimization functions: GRU parameter optimization. Experiment with epochs = 1000, batch size= 64, first layer neurons= 256, and learning rate= 0.2 and window length= 6 and optimization = {Adam, Nadam}, 'Adam' performed better than 'Nadam'.

#	$Optimizer$	TE	VE	$RMSE$	$MAPE$	MAE
1	**Adam**	**2.13**	**114.95**	**7.69**	**7.21**	**4.13**
2	Nadam	29.25	89.86	8.45	9.67	5.26

used in these experiments are: activation: RELU, loss: MSE, optimizer: Adam, Last Layer Activation: Sigmoid and Batch Size: 64 and 256 respectively. We observed following set with better performance $\{\eta = 0.1, BatchSize = 64, Slidingwindowsteps = 6, Epochs = 100, Neurons = 256\}$. Next experiment (Table. 3) is conducting with already searched best settings found in previous step. we set the number of learning rate in each row as the η. We deliberately kept now one parameter variable, the others are fixed - in this case the learning rate $\eta\epsilon\{0.01, 0.05, 0.1, 0.2, 0.5, 0.8, 1.0\}$. The experiment results in obtaining $\eta = 2$ performed better than others. Next experiment (Table. 4) is performed with already searched best settings found till previous step. We set L as the number of sliding window steps ahead used in each row. We kept one parameter as variable, the others are kept fixed - in this case the window steps $L\epsilon\{3, 6, 12, 18, 24\}$. The experiment results in obtaining $L = 12$ performed better than others on proposed dataset. Eventually, last experiment (Table. 5) is performed with already searched best settings by evaluating the two optimizer functions 'Adam' and 'Nadam'. We found that the optimizer 'Adam' performed better than 'Nadam'. An experiment is illustrated in Fig. 3 to perform a search tuning learning rate and time-windows steps for the GRU network. An illustration of time-series is presented in Fig. 8 presenting the traffic-flow (number of Vehicles) ranges from January 7, 2019 to January 20, 2019 at Freeway 99, District 10, California. Red line represents actual data and green line is GRU prediction using sliding window mechanism proposed in this work.

E. COMPUTATIONAL ANALYSIS

Both LSTM and GRU are widely known variants of RNN. They are used for traffic flow prediction using temporal dependency. This process uses time-lags or sliding window that essentially feed the data to neural network step by step. Having used the various deep learning variants of LSTM,

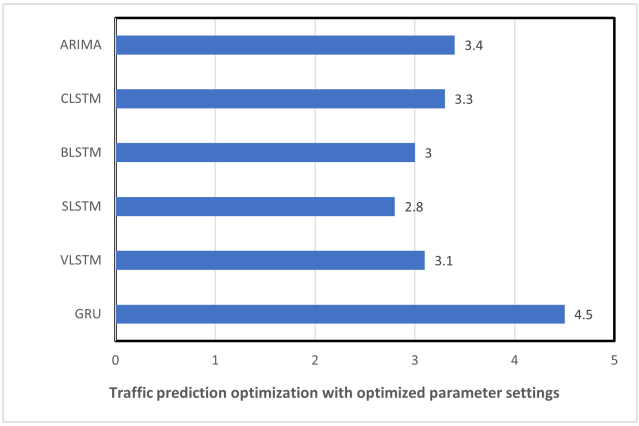

FIGURE 5. Average performance gain of models for traffic prediction with optimized parameter settings. Abbreviations: GRU, gated recurrent unit; ARIMA, autoregressive integrated moving average; VLSTM, vanilla long short-term memory; SLSTM, stacked LSTM; BLSTM, bidirectional LSTM; CLSTM, a Convolutional LSTM Network.

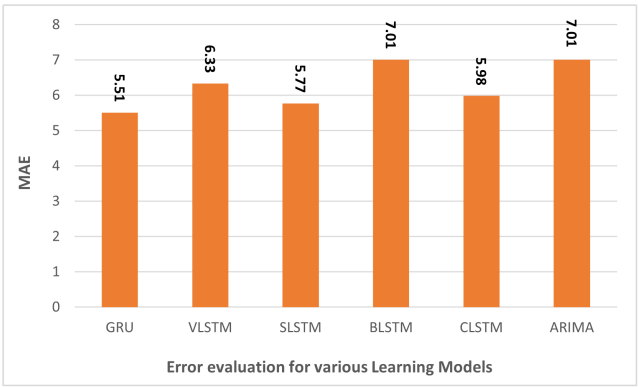

FIGURE 6. Mean absolute error of models for traffic prediction with optimized parameter settings.

we presented to compare accuracy and computation time during training by using the fine-tuned hyperparameters for vanilla LSTM, Stacked LSTM, bidirectional LSTM and GRU models. These models are used to process the time series traffic data and predict the traffic flow. Training time for different models is presented in following graph with Fig. 7. We can observe that GRU possess training time lesser that other models. This is because GRU by virtue of design is a simpler model with lesser number of gates. Bidirectional LSTM model performed badly with traffic time-series based data due to the reason of processing data in two directions.

F. EVALUATION METRICS

To evaluate the effectiveness of the proposed GRU method, we used three performance metrics: the mean absolute error (MAE), the root mean square error (RMSE) and the mean absolute percentage error (MAPE). MAPE is the sum of the individual absolute errors divided by the demand (each period separately). It is the average of the percentage errors. The Mean Absolute Error (MAE) is a popular KPI to measure forecast accuracy and is the mean of the absolute error. The Root Mean Squared Error (RMSE) is another KPI, very

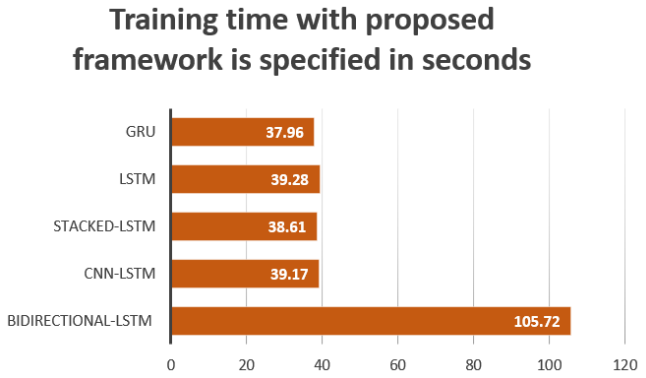

FIGURE 7. Training time (in seconds) of the models for traffic prediction with optimized parameter settings.

helpful in time-series predictions and is defined as the square root of the average squared error. Below is the mathematical expression of these measures. $e_T = \hat{y^T} - y^T$ in below expressions is the error defined by the difference between predicted value and ground truth. The mathematical expression is the three metrics RMSE, MAE and MAPE are represented by following equations: [1], [21]

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_T^2} = \sqrt{\frac{1}{n} * \sum_{t=1}^{n} (\hat{y^T} - y^T)^2} \tag{2}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |e_T| = \frac{1}{n} * \sum_{t=1}^{n} |\hat{y^T} - y^T| \tag{3}$$

$$MAPE\% = \frac{100}{n}\sum_{t=1}^{n} \left|\frac{e_T}{y^T}\right| = \frac{100}{n} * \sum_{t=1}^{n} \left|\frac{\hat{y^T} - y^T}{y^T}\right| \tag{4}$$

Evaluation results are presented in Table. 6, we compared Gated recurrent unit, Autoregressive integrated moving average, Vanilla LSTM, Bidirectional LSTM, Stacked LSTM, and CNN-LSTM. AVG^1 represents the average value of RMSE, MAPE and MAE measures for the experiment conducted to determine traffic prediction with common or untuned parameter settings. Similarly, AVG^2 represents the average value of RMSE, MAPE and MAE for the experiment traffic prediction with optimized parameter settings. We presented the Gain in average error by following expression:

$$Gain = |AVG^2 - AVG^1| \tag{5}$$

The error e_T can be a positive or negative depending upon the forecast overshoots or undershoots the ground truth, however, in these measures we are more interested in the error magnitude. To evaluate the performance of Proposed model, we tend to use four effectiveness metrics. Various traffic prediction research articles use unanimous set of evaluation metrics because it is an intricate task [37]. The most common metrics are RMSE, MAPE and MAE. A problem with them is that they fail to provide comparable measures when data is multi-dimensional, and complexities of the network models taken in experiment are divergent [38]. Another aspect is about spatio-temporal nature of the traffic data,

TABLE 6. Evaluation performed for prediction accuracy of various models. Test performed on hourly Traffic flow ranges from January 7 to 20 at Freeway #99, District 10, California. Traffic data with random setting is: No. of Neurons in first layer: 64, window steps: 2, Epochs: 1000, Batch size: 128 and Learning rate: 0.01. Traffic data with optimized hyperparameters setting is: No of Neurons in first layer: 256, window steps: 6, Epochs: 1000, Batch size: 64 and Learning rate: 0.2. After tuning network parameters - the prediction accuracy is the network models increased significantly.

		Traffic prediction with common or untuned parameter settings				Traffic prediction with optimized parameter settings				
#	*Model*	*RMSE*	*MAPE*	*MAE*	AVG^1	*RMSE*	*MAPE*	*MAE*	AVG^2	*Gain*
1	GRU	12.20	11.18	6.65	10.01	**7.13**	**5.93**	**3.47**	5.51	**4.50**
2	VLSTM	11.01	10.92	6.50	9.47	**7.16**	**7.62**	**4.22**	6.33	3.14
3	SLSTM	9.86	10.13	5.60	8.53	**6.97**	**6.58**	**3.77**	5.77	2.75
4	BLSTM	11.46	11.77	6.81	10.01	**7.68**	**8.75**	**4.60**	7.01	3.00
5	CLSTM	11.08	10.78	5.98	9.28	**7.35**	**6.62**	**3.99**	5.98	**3.29**
6	ARIMA	11.14	12.89	7.28	10.43	**7.73**	**8.51**	**4.79**	7.01	**3.42**

when trained with the neural network models, errors can possibly propagate through time and space dimensions during training. Therefore, proposing a measure like average error gain is a motivation to help measuring model's performance by describing a diversified correlation. Results are presented with the gain measure obtained for all of the six deep learning models described in the next sub-section.

G. PREDICTION RESULTS AND COMPARISON OF MODELS

We mentioned in the previous section - to obtain better parameters from the sample space, we performed five successive experiments on the GRU model. We performed the search using hourly samples to reduce the processing cost instead of original 5-minutes observations found at PeMS traffic data. Assuming that the parameters will perform well for other DL models which use similar sliding window technique for prediction, we tested our approach with six models: GRU, Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, CNN-LSTM and ARIMA. The first experiment is conducted as follows: The performance comparison with different prediction models before tuning of hyperparameters is performed by choosing a random setting from the sample space such that: No. of Neurons in first layer 64, window steps 2, Epochs 1000, Batch size 128 and Learning rate 0.01. For efficient measurement we evaluated three metrics: RMSE, MAPE, and MAE. The second experiment is performed using the optimized hyperparameters obtained after the execution of the intended algorithm. After using optimized hyperparameters values to verify the prediction of six network models, we found the accuracy is increased. this is presented in Fig. 5 and Fig. 6 as graph. We used following fine-tuned parameter values to perform the second prediction test: No of Neurons in first layer 256, window steps 6, Epochs 1000, Batch size 64 and Learning rate 0.2. We can observe the prediction accuracy of various models increases by using tuned hyperparameters and tuned sliding window length obtained for GRU network. Results are presented with the gain measure obtained for all of the six models. Results show that average gain value for the optimized GRU is higher than the normal untuned model with a relative improvement of 4.5% and is compared with rest of the neural network models presented in Table. 6.

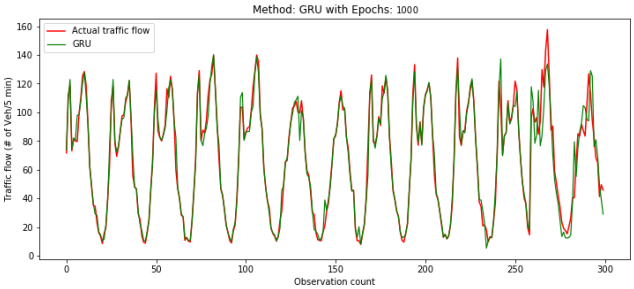

FIGURE 8. Detail of Average Traffic flow (number of Vehicles) ranges from January 7, 2019 to January 20, 2019 at Freeway #99, District 10, California. Red line represents actual data and green line is GRU prediction using sliding window mechanism to obtain performance metrics.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented the fundamentals of the GRU network and proposed a hyperparameter optimization analysis coupled with window steps tuning for time series prediction. We addressed the advantage of starting with a simple search and progressively narrow down subsequent searches keeping one parameter variable and others fixed. To search for better parameter sets from sample space, we conducted five experiments in series. Results show a higher capability of the proposed method to reduce the error and an average gain of the optimized network over the normal model is 4.5%. After obtaining the refined parameter sets, we performed tests on various models in order to achieve accuracy and stability. The proposed GRU parameter optimization algorithm helps to search the suitable parameter combinations that minimize the error metrics; RMSE, MAPE and MAE measure 7.13, 5.93 and 3.47 respectively. This paper further verifies the applicability and validity of the proposed model. The best value of performance gain is provided by GRU model then comes ARIMA and CNN-LSTM models with the Gain values 4.50, 3.42 and 3.29 respectively.

A potential candidate for future work might be to feed the network the historical temporal information with external

factors like non-recurrent events coupled with traffic flow to reduce error and further, we can tune the hyperparameters of the network and benchmark the prediction accuracy in defined scenarios. Secondly, we may explore spatial and temporal features using graph-based convolution. Graph convolution will be our next area of exploration for prediction improvements.

REFERENCES

[1] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 3, pp. 247–254, Jul. 2016.

[2] X. Zheng, W. Chen, P. Wang, D. Shen, S. Chen, X. Wang, Q. Zhang, and L. Yang, "Big data for social transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 620–630, Mar. 2016.

[3] X. Wang, X. Zheng, Q. Zhang, T. Wang, and D. Shen, "Crowdsourcing in ITS: The state of the work and the networking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1596–1605, Jun. 2016.

[4] S. Wambura, H. Li, and A. Nigussie, "Fast memory-efficient extreme events prediction in complex time series," in *Proc. 3rd Int. Conf. Robot Syst. Appl.*, Chengdu, China, Jun. 2020, pp. 60–69.

[5] 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," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[6] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with LSTM neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3127–3141, Aug. 2020.

[7] A. J. Khattak and B. Wali, "Analysis of volatility in driving regimes extracted from basic safety messages transmitted between connected vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 48–73, Nov. 2017.

[8] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1946–1960, Nov. 2017.

[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[10] G. Bao, Z. Zeng, and Y. Shen, "Region stability analysis and tracking control of memristive recurrent neural network," *Neural Netw.*, vol. 5, no. 1, pp. 74–89, 2017.

[11] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, China, Dec. 2015, pp. 153–158.

[12] A. Rosenblad, "J. J. Faraway: Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models," *Comput. Statist.*, vol. 24, no. 2, pp. 369–370, May 2009.

[13] Y. Zhang and Y. Liu, "Traffic forecasting using least squares support vector machines," *Transportmetrica*, vol. 5, no. 3, pp. 193–213, Sep. 2009.

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

[15] H. Chang, Y. Lee, B. Yoon, and S. Baek, "Dynamic near-term traffic flow prediction: Systemoriented approach based on past experiences," *IET Intell. Transp. Syst.*, vol. 6, no. 3, pp. 292–305, Sep. 2012.

[16] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[17] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowl.-Based Syst.*, vol. 172, pp. 1–14, May 2019.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[19] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 166–180, May 2018.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, Nov. 2018.

[22] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transp. Syst.*, vol. 11, no. 9, pp. 531–536, Nov. 2017.

[23] L. Mou, P. Zhao, H. Xie, and Y. Chen, "T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction," *IEEE Access*, vol. 7, pp. 98053–98060, Jul. 2019.

[24] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

[25] F. Kong, J. Li, B. Jiang, T. Zhang, and H. Song, "Big data-driven machine learning-enabled traffic flow prediction," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 9, p. e3482, Sep. 2019, doi: 10.1002/ett.3482.

[26] H.-P. Lu, Z.-Y. Sun, and W.-C. Qu, "Big data-driven based real-time traffic flow state identification and prediction," *Discrete Dyn. Nature Soc.*, vol. 2015, no. 9, pp. 1–11, 2015.

[27] F. Kong and J. Li, "The promotion strategy of supply chain flexibility based on deep belief network," *Appl. Intell.*, vol. 48, no. 5, pp. 1394–1405, May 2018.

[28] H. Khadilkar, "Data-enabled stochastic modeling for evaluating schedule robustness of railway networks," *Transp. Sci.*, vol. 51, no. 4, pp. 1161–1176, Nov. 2017.

[29] (2014). *Caltrans, Performance Measurement System (PeMS)*. [Online]. Available: http://pems.dot.ca.gov

[30] *Caltrans CCTV Map*. Accessed: Jan. 27, 2020. [Online]. Available: http://cwwp2.dot.ca.gov/vm/iframemap.htm?long=-120.49&lat=37.7&zoom=24

[31] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. Chin. Assoc. Automat. IEEE*, 2017, pp. 324–328.

[32] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001–2013, Jun. 2019.

[33] 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.

[34] (Jul. 2010). *Directive 2010/40/EU of the European Parliament and of the Council of 7*. [Online]. Available: https://eur-lex.europa.eu/

[35] PerthNow. (Jul. 7, 2020). *Smart Tech to End FWY Congestion*. Accessed: Oct. 7, 2020. [Online]. Available: https://www.perthnow.com.au/news/transport/way-of-the-future-perths-47m-smart-freeway-to-be-switched- on-in-august-ng-b881601834z

[36] J. Khan, M. Fayaz, A. Hussain, S. Khalid, W. K. Mashwani, and J. Gwak, "An improved alpha beta filter using a deep extreme learning machine," *IEEE Access*, vol. 9, pp. 61548–61564, 2021.

[37] B. Van Arem, H. R. Kirby, M. J. M. Van Der Vlist, and J. C. Whittaker, "Recent advances and applications in the field of short-term traffic forecasting," *Int. J. Forecasting*, vol. 13, no. 1, pp. 1–12, Mar. 1997.

[38] H. van Lint and C. van Hinsbergen, "Short-term traffic and travel time prediction models," *Artif. Intell. Appl. Critical Transp.*, vol. 22, pp. 22–41, Nov. 2012.

BASHARAT HUSSAIN received the M.S. degree in computer science from IIUI, Islamabad, Pakistan, in 2014. He is currently pursuing the Ph.D. degree in computer science with COMSATS University Islamabad at Wah. He has more than 22 years of professional experience building software applications and has worked with multinational companies from Europe and USA. He is also working as a Senior Lecturer with the Faculty of Computing, Riphah International University, Islamabad. He has professional skills, including software design and architecture with C++, C#, ASP.NET Core, Matlab, and python languages.

MUHAMMAD KHALIL AFZAL (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from the COMSATS Institute of Information Technology at Wah, Pakistan, in 2004 and 2007, respectively, and the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, South Korea, in December 2014. From January 2008 to November 2009, he has served as a Lecturer for Bahauddin Zakariya University, Multan, Pakistan, and with King Khalid University, Abha, Saudi Arabia, from December 2009 to June 2011. He is currently working as an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Wah Cantt. His research interests include wireless sensor networks, *ad-hoc* networks, data-driven intelligence in wireless networks, smart cities, 5G, and the IoT. He was a recipient of the Best Paper Awards of ACM 7th International Conference on Computing Communication and Networking Technologies and in International Conference on Green and Human Information Technology (ICGHIT-2018), and fully funded scholarship for Masters and Ph.D. He is serving as a Guest Editor for *IEEE Communication Magazine*, *Transactions for Emerging Telecommunications Technologies* (ETT), *Future Generation Computer Systems* (Elsevier), IEEE Access, *Journal of Ambient Intelligence and Humanized Computing* (Springer), and a Reviewer for IEEE Access, *Computers and Electrical Engineering* (Elsevier), *Journal of Network and Computer Applications* (Elsevier), *FGCS*, and IEEE Transactions on Vehicular Technology.

SHAFIQ AHMAD received the Ph.D. degree from RMIT University, Melbourne, Australia. He has more than two decades of working experience both in industry and academia in Australia, Europe, and Asia. He is currently working as an Associate Professor with the Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh, Saudi Arabia. He is also a Certified Practitioner in Six Sigma business improvement model. He has published a research book, a book chapter, and several research articles in international journals and refereed conferences. His research interests include smart manufacturing, the Industrial Internet of Things (IIoT) and data analytics, multivariate statistical quality control, process monitoring and performance analysis, operations research models, and bibliometric network analysis.

ALMETWALLY M. MOSTAFA received the B.Sc. degree in computers and systems engineering and the M.Sc. degree from Al-Azhar University, Egypt, and the Ph.D. degree in computers and systems from a channel program between the Université catholique de Louvain, Belgium, and Al-Azhar University. He worked for two years at the Centre d'Excellence en Technologies de l'Information et de la Communication (CETIC), Belgium. He is currently an Associate Professor with the Department of Information Systems, College of Computers and Systems (IS/CCS), King Saud university (KSU). His research interests include distributed systems protocols, cloud–mobile-pervasive computing, fault-tolerance, and security of information systems.