An End-to-End Recommendation System for Urban Traffic Controls and Management Under a Parallel Learning Framework

Junchen Jin, Haifeng Guo, Jia Xu, Xiao Wang, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*— A paradigm shift towards agile and adaptive traffic signal control empowered with the massive growth of Big Data and Internet of Things (IoT) technologies is emerging rapidly for Intelligent Transportation Systems. Generally, an adaptive signal control system fine-tunes signal timing parameters based on pre-defined control hyperparameters using instantaneous traffic detection information. Once traffic pattern changes, those hyperparameters (e.g., maximum and minimum green times) need to be adjusted according to the evolution of traffic dynamics over a very short-term period. Such adjustment processes are usually conducted by professional and experienced traffic engineers. Here we present a human-in-the-loop parallel learning framework and its utilization in an end-to-end recommendation system that mimics and enhances professional signal control engineers' behaviors. The system has been deployed into a real-world application for an extended period in Hangzhou, China, where signal control hyperparameters are recommended based on large-scale multidimensional traffic datasets. Experimental evaluations demonstrate significant improvements in traffic efficiency through the use of our signal recommendation system.**

***Index Terms*— Intelligent traffic control, traffic signal control, parallel learning, recommendation systems, deep neural networks.**

I. Introduction

A. Background

URBAN traffic controls have been widely implemented and are one of the most important technical means for regulating traffic flow, improving safety, and reducing emissions. With decades of experience in road traffic controls, intelligent traffic control systems become effective and vital for alleviating traffic-related problems in urban areas, especially with the recent and rapid development of emerging technologies in Big Data and the Internet of Things (IoT).

In the past few decades, several self-adaptive signal control systems have been proposed. For instance, SCOOT [1] and SCATS [2] are deployed in many cities around the world whose signal timing parameters are adjusted according to infrastructure-based detectors as instantaneous input for their internal control logic. Similarly, the recently developed novel adaptive signal control systems determine signal timing parameters using connected-vehicle data with the advances in wireless communication technology, i.e., [3]. Such adjustments are based on pre-determined control hyperparameters (e.g., nominal timing parameters, maximum and minimum green lengths, etc.).

Traditionally, control hyperparameters are regularly changed according to a pre-defined schedule that relies on historical traffic patterns. However, due to the highly stochastic property of the traffic system, traffic patterns with a specific urban area are holistic time-varying. The available settings of control hyperparameters cannot handle situations that traffic patterns constantly change in a short-term period (e.g., an hour), especially during rush hours.

Nowadays, information service providers that hold large-scale heterogeneous data sources contribute to urban traffic controls by providing real-time traffic analytics. Thanks to a large amount of data available, a complete picture of real-time traffic patterns can be analyzed using streaming spatiotemporal data. Accordingly, control hyperparameters can be frequently adjusted using the time-varying urban traffic features to facilitate traffic efficiencies. To complete the decision-making process of control hyperparameters, multidimensional information and dynamic traffic patterns need to be taken into consideration. Such a complicated process is usually performed by humans in engineering practice.

For example, a real-time traffic analytics platform called the City Brain Platform has been deployed to monitor city-level urban traffic in Hangzhou, China [4]. Meanwhile, a professional team of signal control engineers manually change signal control hyperparameters using the obtained multidimensional traffic information. Fig. 1 demonstrates the operational process for adjusting control hyperparameters by humans.

Manuscript received April 14, 2019; revised November 4, 2019; accepted January 24, 2020. This work was supported in part by the China Post-Doctoral Science Foundation under Grant 2019M660136, in part by the Natural Science Foundation of Zhejiang Province under Grant LY20E080023, and in part by the National Natural Science Foundation of China under Grant U1811463. The Associate Editor for this article was L. Li. *(Corresponding author: Haifeng Guo.)*

Junchen Jin is with Enjoyor Co., Ltd., Hangzhou 310030, China, and also with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: junchen@kth.se).

Haifeng Guo is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310013, China, and also with Enjoyor Co., Ltd., Hangzhou 310030, China (e-mail: guohf@zjut.edu.cn).

Jia Xu is with Enjoyor Co., Ltd., Hangzhou 310030, China, and also with the College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China (e-mail: xu.jia@enjoyor.net).

Xiao Wang and Fei-Yue Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: x.wang@ia.ac.cn; feiyue@ieee.org).

Digital Object Identifier 10.1109/TITS.2020.2973736

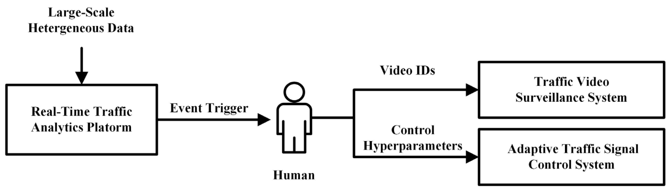

Fig. 1. The operational process for adjusting traffic control hyperparameters by humans.

The real-time traffic analytics platform continuously reports events that cause urban traffic problems. An event indicates the problem type (e.g., accident, over-saturation, etc.), problem location (e.g., intersection or lane), and problem duration. The professional signal control engineers pay attention to the traffic conditions of the relevant intersections via a traffic video surveillance system. Thereafter, an engineer adjusts control hyperparameters for traffic signal control system according to his/her signal control knowledge and experiences.

Human adjustments on control hyperparameters are successful to some degree.[1] Adjustments made by professional engineers are time-consuming (five to 10 minutes per-adjustment per person). Thus, professional engineers devote themselves to handling the most congested intersections in the city during rush hours and leave the other problems to dissipate independently along with the decrease in traffic demand. Manual regulation is practically difficult to handle large-scale urban traffic controls, especially during peak periods. There is an urgent need for developing a software-designed real-time system integrating models capable of imitating the behaviors of signal control professionals and improving traffic efficiencies. Moreover, suitable signal control hyperparameters are not necessary to be continuously adjusted if traffic patterns are not substantially changed.

B. Parallel Learning for Urban Traffic Controls

Indeed, a small portion of signalized intersections is treated well relative to the other intersections in the city, however, which hinders a data-hungry problem while implementing state-of-art algorithms. A methodological framework called the parallel learning (PL) framework was designated to address this issue for complex systems [6]. In this framework, parallel systems, including real-world complex systems and artificial systems, are synchronized [7].

Three learning processes are formulated within the framework: descriptive learning, predictive learning, and prescriptive learning. Descriptive learning is used to form a self-consistent system that coincides with observations from the real-world system. A self-boosting process is performed by predictive learning to rectify the mapping relationship between the observed data and the system state in the artificial system. Then, similar actions can be generated in both real and artificial systems. In turn, more and more records reveal the consequences of the control policy. When prescriptive learning is employed, an optimal control policy is determined in the artificial system that can be simultaneously adopted in the real system.

According to the descriptive learning concept under the PL framework, the foremost task is to develop a system that understands and imitates professional signal engineers' behaviors and can automatically adjust signal control hyperparameters. The system is subject to the concept of recommendation technologies, where recommendations are provided to satisfy the demands of users by mining the internal relationship between users and recommended objects with a large amount of data. A recommendation system generally facilitates the decision-making process in a wider range of applications due to its ability to solve many complex tasks and provide start-of-the-art results [8].

Consequently, the presented paper demonstrates the initial study by applying a parallel learning framework into the recommendation system for urban traffic controls and management. In particular, a human-like recommendation system is proposed for urban traffic controls under the concept of descriptive learning, which captures the true underlying relationship between professional signal control engineers' knowledge, real-time traffic patterns, and control hyperparameter decisions. This paper is organized as follows. In section II, the developments of self-adaptive control systems are pointed out and the related recommendation technologies are reviewed. Section III then presents the detailed design of the proposed end-to-end urban traffic control recommendation system. In section IV, a case study is carried out and the corresponding setups, data description, model result analysis, and discussions are described. The last section summarizes the main findings and discusses possible future research.

II. Related Studies

After fixed-time and actuated signal control systems, self-adaptive signal control systems are the third generation of urban traffic control systems [9]. The development of self-adaptive traffic control systems is divided into five generations [10], starting with completely isolated self-adaptive control utilizing multi-time timing control. Then, during the second and third generations, centralized and distributed signal control systems were proposed, respectively. These systems dynamically fine-tune signal timing parameters according to the collected data, including fixed-location [11] and mobile data [12]. In the fourth generation, integrated management and control of network traffic were realized, which maximized the technical and performance advantages of multiple subsystems. The need for frequently changing signal control hyperparameters has been spotlighted by the research community. Thus, signal control systems in the fifth generation have been developed based on the abilities of self-learning of control hyperparameters to improve system performances and compensate for the effects of the continuous changes in traffic patterns.

Despite a substantial amount of researches make effort on the fifth-generation signal control systems, most studies

[1] According to the "Traffic Analysis Report," which was produced in the third quarter of 2018 by a third-party (AutoNavi, one of the largest mobile navigation service providers in the world), Hangzhou has the highest signal control ability (according to traffic flow and signal control parameters) in morning and evening peak periods out of the top five major cities in China [5].

have focused on developing self-learning traffic signal control systems using emerging machine learning (ML) methods (e.g., [13], [14]). Nevertheless, most of the developed signal control systems have been implemented in a simulated environment without actual deployment in practice. Although many studies have attempted to adjust signal control hyperparameters for the deployed system by different optimization techniques (e.g., [11]), to the best of the author' s knowledge, no previous research has demonstrated the ability of signal hyperparameter adjustments at run-time, particularly on a city-wide scale. Besides, Zhao et al. began pioneering the application of recommendation techniques in traffic signal control systems. A well-known recommendation technology, collaborative filtering, was applied to find the best signal timings [15]. Nevertheless, none of the studies have incorporated human's professional knowledge into a recommendation for urban traffic signal control systems.

Traditionally, pre-defined features are hand-engineered for a recommendation system that limit the potential performance as some of them can be a poor approximation of reality. Although, some studies have made efforts on traffic control methods based on various techniques, such as temporal dependency analysis [16], optimization method [17], and reinforcement learning [13], [18], the recently developed deep neural network (DNN) models have been revolutionizing the recommendation architectures due to their end-to-end property [19]. End-to-end learning means that the pipeline is with a single learning algorithm so that it directly uses inputs to model the desired output. DNN models are capable of modeling nonlinearity in data as well as are efficacious in learning the underlying explanatory factors and useful representations from input data. In this way, DNN models provide opportunities to improve recommendation accuracy [20].

Different DNNs have various technical characteristics [21]. For instance, recurrent neural networks (RNNs) are specially designed for sequence modeling, e.g., time series analysis [22]. Long short-term memory (LSTM) units [23] and gated recurrent (GRU) units [24] have overcome the limitations of RNNs in vanishing gradients. Moreover, some studies have utilized attention-based RNNs to select parts of hidden states across all time steps [25]. Additionally, DNN model performance highly relies on a neural structure that is usually determined by a search technique [26]. Therefore, the choice of a DNN model type together with its model structure should be carefully determined in engineering practice.

III. End-to-End Recommendation System for Urban Traffic Controls

A. System Architecture

This study proposes an architecture of a DNN-based end-to-end recommendation system for urban traffic controls (see Fig. 2). In general, the recommendation system interacts with a real-time traffic analytics platform, a traffic video surveillance system, and an adaptive traffic signal control system. The operational process of recommending control hyperparameters is shown in Fig. 2. The difference between human operations in Fig. 1 is that an event automatically triggers the executions of end-to-end urban traffic recommendation system. The suggested control hyperparameters and video IDs are later generated.

The recommendation system consists of five major components: historical multidimensional databases, a data preprocessing module, a deep recommendation algorithm pool, an algorithm selector, and a neural architecture generator. The training process of a DNN model in the deep recommendation algorithm pool is as follows. First, raw historical control hyperparameters and detection information are stored in the databases. The data is further manipulated by a data processing module using existing outlier detection and missing data imputation techniques [27]. Then, the processed historical data are used to train DNN models that effectively capture the spatiotemporal relationships between the observed multidimensional data and control hyperparameters. Noted that the historical multidimensional database does include engineers' past experience and knowledge on the practice of deciding hyperparameters such that the recommendation can learn from professional behaviors. Different DNN models, including GRU, LSTM, bidirectional RNN (BiLSTM and BiGRU), attention-based RNN (AttRNN), dual attention RNN (DARNN), and combined convolutional neural networks (CNN) and RNN (CNN-BiLSTM and CNN-BiGRU) are trained in the study. Model details of these DNNs are illustrated in the next subsection.

The neural architecture generator determines a neural architecture (e.g., number of hidden units for each layer and layer size) and neural operators (e.g., activation function) using a controller. The controller implements an efficient reinforcement learning-based RNN to maximize the expected recommendation accuracy of the generated architectures on a validation set. The applied neural architecture search algorithm is proposed in [28]. After the best neural architectures have been determined, nine trained DNN models and validation results are stored for each intersection. When an intersection is called, the algorithm selector compares the validation results and automatically chooses the most appropriate DNN model to generate control hyperparameters.

B. Deep Recommendation Algorithm Pool

1) Notation and Problem Statement: A recommendation model aims to provide future control hyperparameters using the historical control hyperparameters and traffic detection information. Specifically, if control hyperparameters at T are predicted, n traffic information series is given, i.e.,

$$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T) \tag{1}$$

$$= \begin{bmatrix} x_{1,1} & \cdots & x_{1,t} & \cdots & x_{1,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{k,1} & \cdots & x_{k,t} & \cdots & x_{k,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,t} & \cdots & x_{n,T} \end{bmatrix} \in \mathcal{R}^{n\times T} \tag{2}$$

where T is time-window size. In particular, $x_{k,t}$ denotes the k^{th} traffic information series at time t. The previous values of

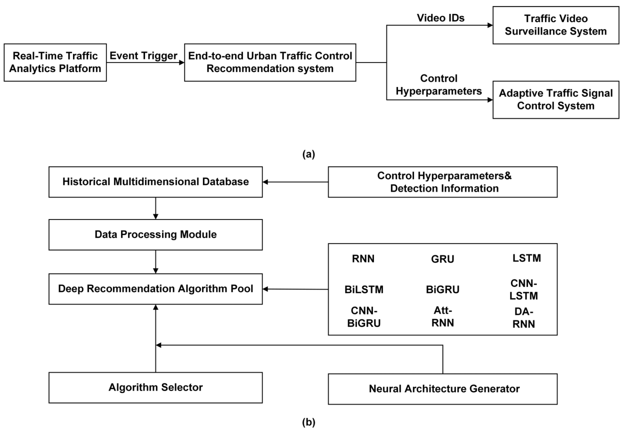

Fig. 2. The operational process (a) and architecture (b) of the end-to-end urban traffic control recommendation system.

control hyperparameters are denoted as:

$$(\mathbf{y}_1, \cdots, \mathbf{y}_t, \cdots, \mathbf{y}_{T-1}) \tag{3}$$

with $\mathbf{y}_t \in \mathcal{R}^m$, where m refers to the output dimension. The current control hyperparameters are predicted by

$$\hat{\mathbf{y}}_T = F(\mathbf{y}_1, \cdots, \mathbf{y}_{t-1}, \cdots, \mathbf{y}_{T-1}, \mathbf{x}_1, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T) \tag{4}$$

where $F(\cdot)$ is one of the applied DNN models.

2) RNN Variants: An RNN model uses loops in its internal memory to deal with sequence data (see Fig. 3(a) for its structure and computation routine). Within its structure, a hidden layer receives an input vector and generates an output vector. At each time iteration t, a hidden state $\mathbf{h}_t$ is maintained and updated based on traffic detection input $\mathbf{x}_t$ and previous hidden state $\mathbf{h}_{t-1}$.

$$\mathbf{h}_t = \mathrm{RNN}_{\sigma_h}(\mathbf{x}_t, \mathbf{h}_{t-1}, W_{rnn,x}, W_{rnn,h}), \tag{5}$$

where $\mathrm{RNN}_{\sigma_h}()$ denotes the RNN's update functions for hidden states, and $W_{rnn,x}$ and $W_{rnn,h}$ represent weights for the hidden layer. The update process of an RNN is illustrated in [29].

LSTM and GRU are two typical RNN variants with network structures that remain the same as an RNN, but the internal repeating module is improved with more complex operations generating hidden states. Such improvement allows LSTM and GRU networks to remember long-term dependencies and GRU has the advantage of computationally efficiency. The LSTM update procedure is presented as

$$[\mathbf{i}_t, \mathbf{f}_t] = \mathrm{LSTM}_{\sigma_{i,f}}(\mathbf{x}_t, \mathbf{h}_{t-1}, W_{lstm,i,f}, \mathbf{b}_{lstm,i,f}) \tag{6}$$

$$\mathbf{s}_t = \mathrm{LSTM}_{cell}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{f}_t, \mathbf{i}_t, \mathbf{s}_{t-1}, W_{lstm,s}) \tag{7}$$

$$\mathbf{o}_t = \mathrm{LSTM}_{\sigma_o}(\mathbf{x}_t, \mathbf{h}_{t-1}, W_{lstm,o}) \tag{8}$$

$$\mathbf{h}_t = \mathrm{LSTM}_{hidden}(\mathbf{o}_t, \mathbf{c}_t) \tag{9}$$

where the LSTM update functions include $\mathrm{LSTM}_{\sigma_{i,f}}()$, $\mathrm{LSTM}_{cell}()$, $\mathrm{LSTM}_{\sigma_o}()$, and $\mathrm{LSTM}_{hidden}()$. $W_{lstm,i,f}$ and $W_{lstm,o}$ are the weight matrices mapping hidden layer and traffic detection series to input, forget and output gates ($\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$). $\mathbf{b}_{lstm,i,f}$ are biases used for the input and forget gate updates. $\mathbf{s}_t$ is the cell state at t and $W_{lstm,s}$ is the trainable

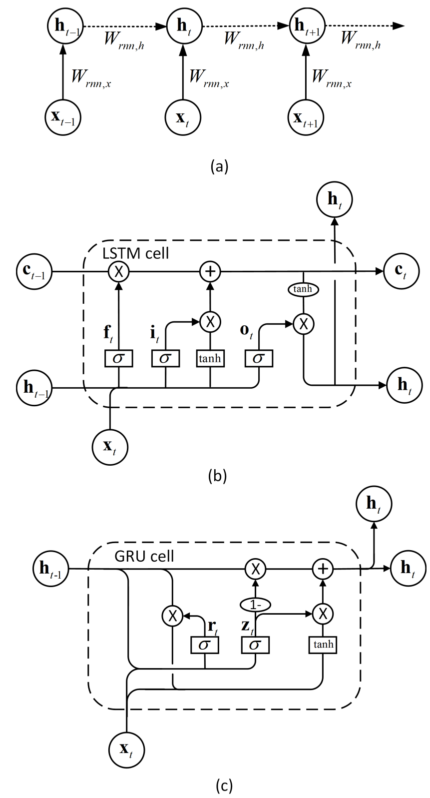

Fig. 3. Neural structures of the implemented RNN variants, (a) RNN structure; (b) LSTM cell structure; (c) GRU cell structure.

weights for updating the cell states. GRU is updated through the following equations:

$$[\mathbf{z}_t, \mathbf{r}_t] = \mathrm{GRU}_{\sigma_{z,r}}(\mathbf{x}_t, \mathbf{h}_{t-1}, W_{gru,z,r}), \tag{10}$$

$$\mathbf{h}_t = \mathrm{GRU}_{hidden}(\mathbf{z}_t, \mathbf{r}_t, \mathbf{h}_{t-1}). \tag{11}$$

Similar to LSTM, $\mathrm{GRU}_{\sigma_{z,r}}$ and GRU_{hidden} are GRU update functions. $W_{gru,z,r}$ refers to the trainable weights of a GRU cell. The update functions for LSTM and GRU are both detailed in [30].

The future control hyperparameters for the three RNN variants are all obtained through a rectified linear unit (RELU), denoting

$$\hat{\mathbf{y}}_T = \mathrm{RELU}(W_{rnn,y}\mathbf{h}_T + \mathbf{b}_{rnn,y}) \tag{12}$$

$$\mathrm{RELU}(x) = \max(0, x) \tag{13}$$

where $W_{rnn,y}$ and $\mathbf{b}_{rnn,y}$ are weights and biases of the output layer, respectively.

This study also implements a bidirectional RNN (BiRNN) model that captures longer-term correlations in practice trained using input data in the past and future from a specific time frame [31]. Fig. 4 demonstrates the general structure of a BiLSTM or a BiGRU model where either LSTM cells or GRU cells are implemented. The idea of a BiRNN is to split state neurons into two parts, with one responsible for the forward states (positive time direction) and the other for backward states (negative time direction). Outputs from the forward states are not connected to the inputs of backward states and vice versa. With both time directions involved in the same

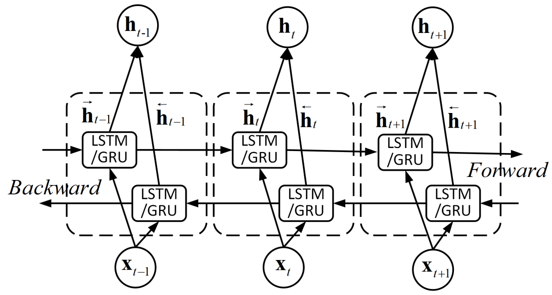

Fig. 4. The calculation process of a bidirectional RNN model.

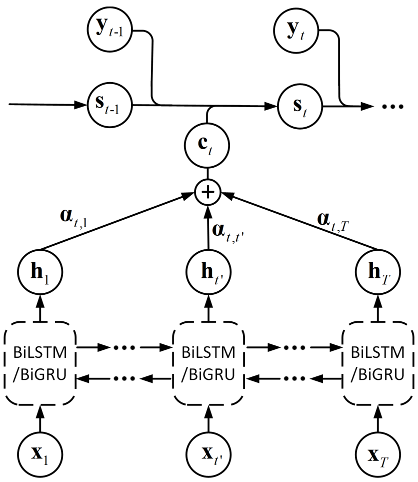

Fig. 5. The update process of an attention-based BiRNN.

network, input information in the future can be directly used in the modeling process without delays. The detailed update equations are given in [31].

3) Attention-Based RNNs: Attention mechanisms have become a popular component for sequence modeling and transduction models, especially in conjunction with a recurrent network [32]. Fig. 5 demonstrates the update process of an attention-based BiRNN (AttBiRNN). Here, a hidden state is generated from the aforementioned BiLSTM or BiGRU cells based on section III-B.2. The weighted sum of all the hidden state with T timespan constructs a context vector $\mathbf{c}_t$ in the encoder module:

$$\mathbf{c}_t = \sum_{t'=1}^{T} \alpha_{t,t'} \mathbf{h}_{t'}, \tag{14}$$

where $\alpha_{t,t'}$ denotes the attention weight between data at t and data at t', computed as

$$\alpha_{t,t'} = \frac{\exp(e_{t,t'})}{\sum_{t''=1}^{T} \exp(e_{t,t''})}, \tag{15}$$

$$e_{t,t'} = a(\mathbf{d}_{t-1}, \mathbf{h}_{t'}, W_{attbirnn,e}), \tag{16}$$

with $a()$ being a feed-forward neural network. $W_{attbirnn,e}$ contains trainable weights. In Eq. 16, $e_{t,t'}$ scores the importance of the hidden state at time t' with respect to the output at time t, given the previous decoder hidden state (i.e., $\mathbf{d}_{t-1}$). By using an attention mechanism, the model concentrates on certain parts of the traffic detection history to predict control hyperparameters. The decoder parallels the encoder that associates each output to a state vector by

$$\mathbf{d}_t = \text{LSTM}(\mathbf{d}_{t-1}, \tilde{\mathbf{y}}_{t-1}), \tag{17}$$

$$\tilde{\mathbf{y}}_{t-1} = W_{attbirnn,y}^{T}[\mathbf{y}_{t-1}; \mathbf{c}_{t-1}] + \mathbf{b}_{attbirnn,y}, \tag{18}$$

where $W_{attbirnn,y}$ and $\mathbf{b}_{attbirnn,y}$ are trainable parameters. The LSTM() function is then with an explicit context added.

Recently, a dual-stage attention-based recurrent neural network (DARNN) was proposed to not only capture long-term temporal dependencies but also to select the relevant exogenous factors to make predictions [33]. As the recommendation task for urban traffic control utilizes spatiotemporal data, the applied DARNN is slightly modified with a structure summarized in Fig. 6.

The DARNN model is broken down into two stages: spatial operations and temporal operations. In the first stage, a spatial attention mechanism is introduced to adaptively extract spatially relevant traffic pattern features at each time step.

$$\tilde{\mathbf{x}}_t = \text{Att}(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{s}_{t-1}), \tag{19}$$

where $\tilde{\mathbf{x}}_t$ refers to the derived exogenous (i.e., traffic detection information) series taking attention weights into account. Att() refers to an attention operation similar to Eq. 14-16 and the calculation procedure is provided in [33]. The encoder is essentially an LSTM unit that transfers the exogenous time series into a feature representation. With the proposed input attention mechanism, the encoder selectively focuses on certain exogenous series rather than treating all series equally by using the following equation:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \tilde{\mathbf{x}}_t, W_{darnn,h}), \tag{20}$$

where $W_{darnn,h}$ denotes the weights for obtaining hidden states of a DARNN model.

In the second stage, a temporal attention mechanism selects encoder hidden states across all time steps. The attention weight β_{t-1}^{τ} of each encoder' s hidden state at each time step represents the importance of the τ^{th} encoder hidden state. Each weight is calculated based upon the previous decoder' s hidden state, which is similar to Eqs.16 - 18. As each encoder hidden state is mapped to a temporal component of the input, the context vector is a weighted sum of all the previous encoder hidden states with timespan T:

$$\mathbf{c}_{t-1} = \sum_{\tau=1}^{T} \beta_{t-1}^{\tau} \mathbf{h}_{\tau}. \tag{21}$$

For both attention-based RNNs, the target control hyperparameters $\hat{\mathbf{y}}_T$ are computed according to a linear function:

$$\hat{\mathbf{y}}_T = \mathbf{v}_{attrnns,y}^{T}(W_{attrnns,y}[\mathbf{d}_T; \mathbf{c}_T] + \mathbf{b}_{attrnns,w}) + \mathbf{b}_{attrnns,v} \tag{22}$$

where $[\mathbf{d}_T; \mathbf{c}_T]$ is a concatenation of the decoder's hidden state and context vector. $\mathbf{v}_{attrnns,y}$, $W_{attrnns,y}$, $\mathbf{b}_{attrnns,w}$ and $\mathbf{b}_{attrnns,v}$ are trainable weight parameters.

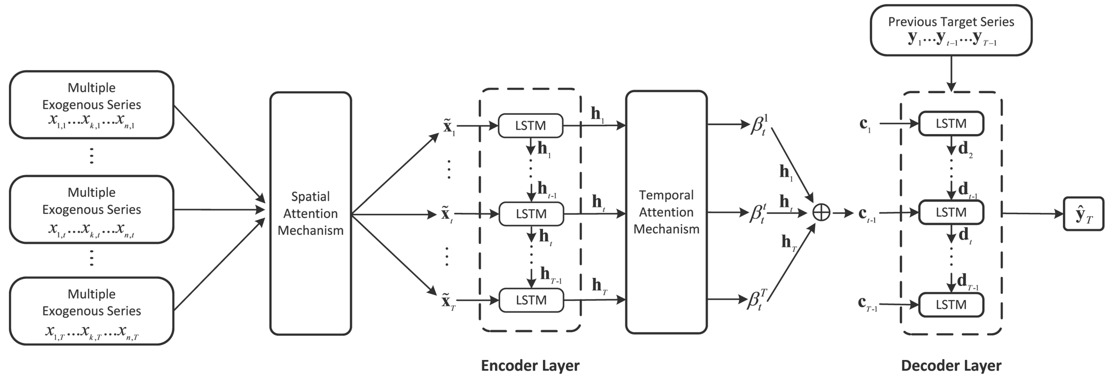

Fig. 6. The neural structure of a dual-stage attention-based recurrent neural network model.

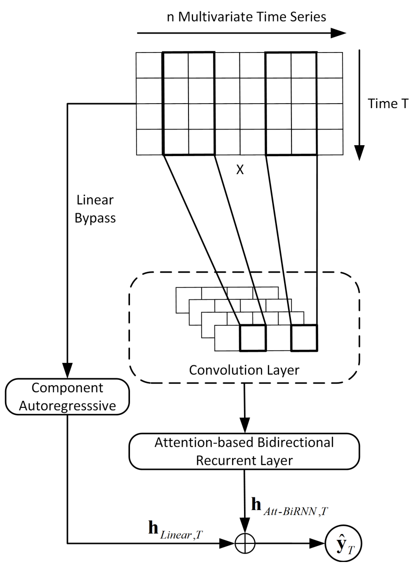

Fig. 7. The structure of a combined CNN and AttBiRNN model.

4) Combined CNNs and RNNs: CNNs have shown outstanding performance by successfully extracting local and shift-invariant features [34]. Unlike RNNs, CNNs automatically learn features from data while exploiting certain temporal invariance. Therefore, combining RNN with CNN can leverage the strengths of both neural structures to discover the local dependency patterns among multi-dimensional input variables by convolutional layers and to capture complex time-variant dependencies by recurrent layers.

This study implements a combined CNN and RNN model (henceforth called a CNN-AttBiRNN) and its structure is shown in Fig. 7. In addition to the CNN and AttBiRNN layers, a classical autoregressive (AR) model, AR(), is incorporated as a linear component to avoid the local scaling issue caused by DNN models. The output of linear component, $\mathbf{y}_{linear,T}$, is computed by

$$\mathbf{y}_{linear,T} = \mathrm{AR}(X, W_{linear}), \tag{23}$$

where W_{linear} denotes the trainable weights for the linear component.

The first layer of the CNN-AttBiRNN is a convolutional network without pooling that aims to extract short-term patterns in the time dimension and local dependencies between variables. The convolutional layer consists of multiple filters of width ω and height n. It should be noted that the height is set to be the same as the number of variables. The k^{th} filter sweeps through the input matrix X and produces

$$\mathbf{h}_k = \mathrm{RELU}(\mathrm{CNN}(W_{cnn,k}, X) + \mathbf{b}_{cnn,k}) \tag{24}$$

where CNN() denotes the convolutional operator. Each vector $\mathbf{h}_k$ is made by zero-padding on the left of the input matrix X. The output matrix of the convolutional layer is of size $K \times T$ where K denotes the number of a filter. The output of the convolutional layer is fed into the AttBiRNN layer.

$$\mathbf{y}_{attbirnn,T} = \mathrm{AttBiRNN}(\mathbf{h}_1, \ldots, \mathbf{h}_k, \ldots, \mathbf{h}_K), \tag{25}$$

where AttBiRNN() denotes the connected recurrent component that adopts the updated equations in Eqs. 14-18. Notably, the AttBiRNN layer learns the weighted combination of the hidden representations at each time-window position of the input traffic detection information. The final prediction of the CNN-AttBiRNN is then obtained by integrating the outputs of the DNN and the AR components:

$$\hat{\mathbf{y}}_T = \mathbf{y}_{attbirnn,T} + \mathbf{y}_{linear,T} \tag{26}$$

IV. Case Study

A. Overview

To justify the effectiveness of the proposed system, a case study was carried out for urban traffic controls in Hangzhou, China, where more than 1,200 SCATS signal controllers are deployed, and most of them are remotely controlled. SCATS is a typical adaptive signal control system that can fine-tune signal timing parameters according to the detected traffic information. Specifically, a predefined base signal plan consists of a nominal cycle length and nominal portions of time allocated to each signal phase, i.e., a split plan. The variation of a split is around 2-5% depending on the detected traffic conditions. Also, the base signal plan is selected from a predetermined set by a deterministic logic. The control logic of SCATS can be found in [2].

Fig. 8. The graphical user interface (GUI) of the implemented end-to-end urban traffic control recommendation platform connected with a traffic video surveillance system and SCATS system.

As mentioned in section I-A, signal control hyperparameters were manually adjusted for the whole of the Hangzhou urban area using the observed traffic patterns by professional signal control engineers. The engineers usually gather real-time traffic pattern information by frequently checking traffic efficiency metrics and remotely watching traffic monitoring videos. Thereafter, an engineer decides on the control hyperparameters for congested intersections based on his/her adjustment experiences. This decision-making process usually takes a professional engineer more than one signal cycle (around 5 to ten minutes). The decision should fulfill two minimum requirements if traffic patterns are not substantially changed within a short period: 1. the congestion problem should be dissipated within three subsequent cycles and 2. no additional problems should occur for the neighboring intersections of the adjusted intersection. Therefore, along with repeating daily operations, a large number of manual interventions are carried out on the signal control hyperparameters.

According to the proposed urban traffic control recommendation system, a web-based platform has been developed and implemented. Fig. 8 shows the graphical user interface (GUI) of the recommendation platform. The recommendation process is automatically performed and a control engineer can easily oversee the recommendation performance using the platform. In particular, the engineer can select any intersection to interact with. Then, the essential traffic efficiency metrics are provided on the main page, meanwhile, the corresponding traffic monitoring videos for the selected area are automatically displayed on the traffic video surveillance system (see Fig. 8). According to the real-time traffic pattern information, the engineer checks whether the recommended control hyperparameters are in line with his/her experience and knowledge. If not, the engineer can directly adjust the control hyperparameters of a specific SCATS controller via the platform. Moreover, such adjustment feedback is recorded in the database, and the feedback is employed to further improve the DNN models in the deep recommendation pool.

B. Data Descriptions

Table I summarizes the descriptions of multidimensional data sources for model development for the DNN models. The real-time traffic analytics platform generates traffic efficiency metric data ($trametric$) every two minutes for all roads in Hangzhou, whereas a signal controller sends out the applied control hyperparameters and detection information when a cycle ends, leading to a varied recording frequency of data sources for each intersection. For model training purposes, the timestamped detection and adaptive signal control information needs to be matched to the closest time concerning the traffic analytics data. The detection information consists of car equivalent flow (vk) and degree of saturation (ds). Therefore, $trametric$, vk, and ds are extracted as the exogenous input factors $\mathbf{x}_t$ each time. Control hyperparameters are the nominal phase and cycle lengths ($phalen$) associated with the signal controller.

TABLE I

DESCRIPTIONS OF THE INVOLVED MULTIDIMENSIONAL DATA SOURCES

Data source	Symbol	Description
Traffic analytics	$date$	Received date
	$timestamp$	Received timestamp
	$intid$	Intersection ID
	$iname$	Intersection name
	$alertid$	Alert notification ID
	$lanid$	Approach lane ID that causes alert notification
	$trametric$	Traffic efficiency metric on the lane $lanid$
Detection information	$date$	Received date
	$cstart$	Starting timestamp of a signal cycle
	$iname$	Intersection name
	$detid$	Detector ID
	$lanid$	Approach lane ID that the detector $detid$ is placed on
	$intid$	Intersection ID that the lane $lanid$ is associated with
	vk	Car equivalent flow (vehicles/cycle) for the lane $lanid$
	ds	Degree of saturation for the lane $lanid$
Adaptive signal control system	$date$	Received date
	$cstart$	Starting timestamp of a signal cycle
	$clen$	Nominal cycle length (seconds)
	$phaid$	Phase ID
	$phalen$	Nominal phase length (seconds)
	$lanids$	IDs of the lanes that are controlled by the phase $phaid$

Nowadays, the proposed recommendation platform is deployed for around 1,000 signalized intersections (see Fig. 9 (a)). The following analysis takes a highly congested arterial road (the Yan An arterial) as an example (marked with a rectangle in Fig. 9). This arterial is in a local political, business, and financial center and is close to the most popular tourist attraction spot in Hangzhou: West Lake. The traffic patterns here are extremely complex and uncertain which involve commuters, tourists, and other travelers. Six SCATS controllers are deployed on the arterial road and the corresponding intersections are Ping Hai-Yan An (SCATS131), Jie Fang-Yan An (SCATS96), Jie Fang-Yan An (SCATS19), Gym-Yan An (SCATS8), Kai Yuan-Yan An (SCATS58), and Xi Hu-Yan An (SCATS34). Their control hyperparameters need to be frequently adjusted to fit the continuously changing traffic patterns.

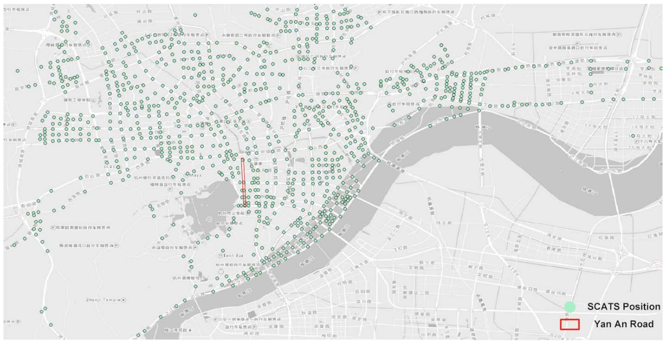

Fig. 9. Locations of the remotely connected SCATS signal controllers and the selected Yan An arterial road (marked with a rectangle).

TABLE II

THE NEURAL ARCHITECTURE OF THE DARNN MODEL USED FOR THE SCATS34 SIGNAL CONTROLLER

Neural architecture hyperparameter	Value
Encoder hidden size	128
Decoder hidden size	128
Time window size	10
Activation function	$tanh$
Learning rate	0.001

C. Training Process

Six months of data from July to December 2018 were used for training and testing the applied DNN models. The data were randomly split into training (70%), validation (15%), and testing (15%) datasets. All the involved models were trained with a commonly used optimizer: ADAM [35]. The training and validation data were used for the neural architecture search and model parameter learning, while the model performance was evaluated using the held-out testing data.

Mean squared error (MSE) is the loss function that applies to all DNN models during training:

$$MSE = \frac{\sum_{t=1}^{N}(\mathbf{y}_t - \hat{\mathbf{y}}_t)^2}{N} \tag{27}$$

where $\mathbf{y}_t$ denotes the actual control hyperparameters adjusted by humans at t and N refers to the number of timesteps in the training set. In general, after the best neural architecture is determined, the training processes ultimately converge at $1,000$ epochs with a batch size of 50 for all the nine DNN models. For example, Fig. 10 shows the training curve of the DARNN model associated with the six SCATS controllers. Most of the loss values appear to converge at 700 epochs and the converged values are all below one. As the generated neural architecture varies among different intersections, Table II provides an example of the optimal neural architecture of the DARNN model for the SCATS34 signal controller, which led to $450,098$ trainable parameters through 66 neural layers.

Root mean square error (RMSE) was used to examine the recommendation performance. The RMSE statistic expresses the differences between the control hyperparameters recommended by a model and the values observed, and is calculated

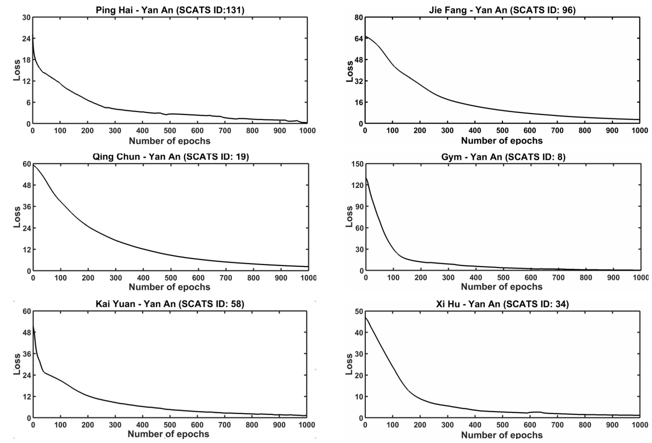

Fig. 10. The training curves for the six signalized intersections (i.e., Ping Hai-Yan An (SCATS131), Jie Fang-Yan An (SCATS96), Jie Fang-Yan An (SCATS19), Gym-Yan An (SCATS8), Kai Yuan-Yan An (SCATS58), and Xi Hu-Yan An (SCATS34).

through:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(\mathbf{y}_t - \hat{\mathbf{y}}_t)^2}{N}}. \tag{28}$$

Table III depicts the test results for all the tested DNN models regarding the six signalized intersections on the Yan An arterial. The lowest RMSE among the tested models for each SCATS controller is highlighted with a star. The results indicate that the difference in control hyperparameters are all below four seconds (around 2% of cycle length) which is reasonably accurate in this recommendation task. As expected, no single model produces the best performance for all cases with respect to the RMSE statistics. In terms of the adjustment frequency, SCATS131 is the highest and SCATS96 is the lowest out of the deployed six SCATS controllers. It can be observed that the recommendation system generates relatively higher errors for the SCATS96 controller. This can be explained by that the decreasing data size has an increasingly detrimental effect on the recommendation performance.

D. Experiment Performance

For further validation, the human's decision and the system recommendation were compared for the "Yan An-Qing Chun" intersection on October 1, 2018. The intersection layout and phase definitions are shown in Fig. 11. Seven phases (A-G) were pre-specified and different phases synthesized the phase sequence during operations. It should be noted that a SCATS controller can also switch nominal signal timing plans from a pre-determined set, which should not be considered a recommendation task.

Table IV describes a control hyperparameter (i.e., nominal phase lengths) comparison between actual executions by a signal control professional and system recommendation. It is observed that the recommended nominal phase lengths are consistent with human's actual decisions by providing the same adjusting trend. The difference in hyperparameter values can be ignored in the urban traffic control context

TABLE III

TEST RESULTS FOR THE NINE DNN MODELS TESTED FOR THE SIGNALIZED INTERSECTIONS ON THE YAN AN ARTERIAL ROAD

ID	Name	RNN	LSTM	GRU	BiLSTM	BiGRU	AttBiRNN	DARNN	CNN-BiLSTM	CNN-BiGRU
SCATS131	Ping Hai-Yan An	1.30	1.93	1.24	1.21(*)	1.45	1.95	1.34	1.81	2.22
SCATS96	Jie Fang-Yan An	4.61	3.69(*)	4.51	4.58	4.41	4.95	4.15	5.01	6.84
SCATS19	Qing Chun-Yan An	3.04	3.34	1.69(*)	2.83	1.96	1.91	2.95	3.45	2.95
SCATS8	Gym-Yan An	3.36	2.70	2.46	1.75	3.92	1.13	3.11	1.05(*)	3.90
SCATS58	Kai Yuan-Yan An	2.13	1.97	1.32	2.08	1.48	1.29(*)	2.16	3.02	4.36
SCATS34	Xi Hu-Yan An	2.17	2.35	2.01	3.97	4.29	2.81	1.63(*)	4.16	5.19

TABLE IV

CONTROL HYPERPARAMETER (I.E., NOMINAL PHASE LENGTHS) COMPARISON BETWEEN ACTUAL EXECUTIONS BY A SIGNAL CONTROL PROFESSIONAL AND SYSTEM RECOMMENDATIONS ON OCTOBER 1, 2018

Time	Current signal hyperparameters	Human's decision	System recommendation
06:27	A:35 B:24 C:34 D:28	A:44 B:30 C:38 D:28	A:43 B:29 C:38 D:29
07:20	A:44 B:30 C:35 D:28	A:45 B:32 C:44 D:35	A:46 B:32 C:45 D:36
12:38	A:53 B:34 C:48 D:36	A:51 B:32 C:35 F:16 D:36	A:51 B:34 C:32 F:17 D:36
13:00	A:51 B:32 C:35 F:16 D:36	A:51 B:33 C:46 D:41	A:49 B:34 C:48 D:39
13:36	A:51 B:33 C:46 D:41	A:48 B:34 C:32 D:38	A:49 B:34 C:31 D:39
14:15	A:48 B:34 C:32 D:38	A:50 B:33 C:45 D:40	A:49 B:34 C:48 D:39
14:44	A:50 B:33 C:44 D:40	A:41 G:14 B:34 C:43 D:45	A:41 G:15 B:36 C:40 D:45
15:25	A:41 G:14 B:34 C:43 D:45	A:47 G:17 B:34 C:45 D:33	A:49 G:18 B:34 C:47 D:31
17:40	A:47 G:18 B:38 C:43 D:38	A:38 G:18 B:35 C:34 F:19 D:37	A:38 G:18 B:32 C:34 F:20 D:38
18:31	A:29 G:18 B:29 C:36 F:17 D:43	A:31 G:18 B:30 C:54 D:46	A:31 G:18 B:31 C:56 D:45
18:49	A:31 G:18 B:30 C:54 D:46	A:31 G:18 B:30 C:41 F:16 D:44	A:31 G:18 B:30 C:40 F:17 D:41
19:51	A:29 G:17 B:29 C:37 F:19 D:39	A:53 B:34 C:48 D:36	A:53 B:32 C:49 D:36
20:33	A:53 B:34 C:48 D:36	A:38 G:15 B:31 C:39 F:14 D:36	A:36 G:14 B:32 C:36 F:17 D:34
21:39	A:27 G:15 B:27 C:29 F:15 D:27	A:43 B:32 C:38 D:31	A:43 B:32 C:39 D:29

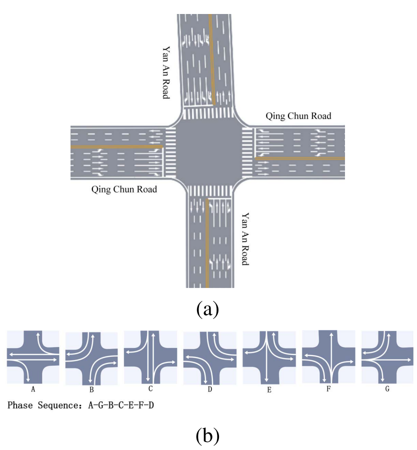

Fig. 11. The layout of the Qing Chun-Yan An intersection (a) and phase definitions of the SCATS19 signal controller (b).

(i.e. three seconds difference at maximum). Thus, the system can provide signal control engineers with a reasonably accurate recommendation. Usually, changing phase structure is a relatively difficult recommendation task while imitating engineers' behaviors. Indeed, the system can successfully handle such a task by giving the same changed phase with a similar amount of phase duration. As a result, shown in Table IV, both a professional engineer and the recommendation system add F and G phases at 20:33 and remove the G phase at 19:51 and 21:39 on October 1, 2018.

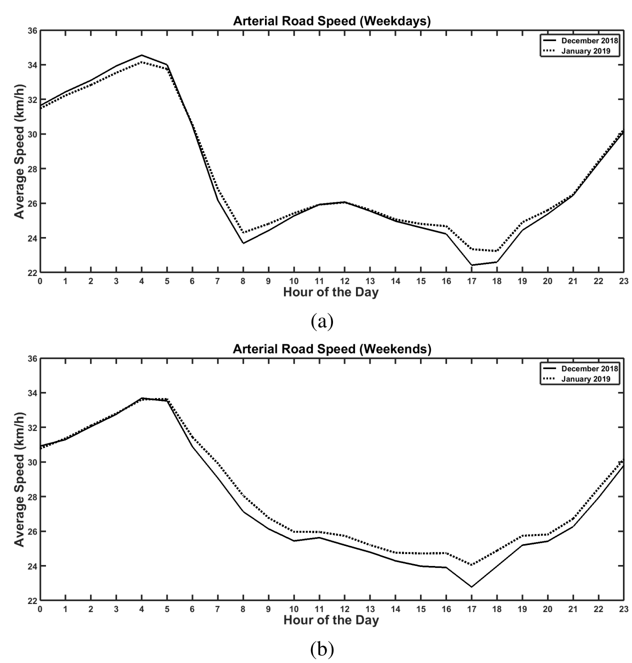

Fig. 12. Average arterial road speed for weekdays (a) and weekends (b) in December 2018 and January 2019 in Hangzhou, China.

Moreover, traffic efficiency performance was continuously monitored for all arterial roads in Hangzhou after the system was deployed. Fig. 12 displays the main characteristics of average arterial road speed in January 2019 after the system was deployed compared to historical statistics in December 2018 when control hyperparameters were adjusted by human's decision. The speed data of arterial roads are measured through a third-party navigation system, AutoNavi [5]. Although the average speed from 00:00 to 06:00 during weekdays results in a slight reduction of around 0.5% compared to the historical data, no abrupt changes in traffic conditions occurred when the recommendation system is adopted. This reduction can be explained by uncertain driver behaviors under free-flow scenarios during midnights.

In contrast, a noticeable increase in speed is reported during critical periods. For instance, the average speeds during the morning peaks associated with arterial roads were 25.56 km/h and 23.58 km/h during weekdays and weekends in January 2019, respectively. The recommendation provided increased average arterial speeds of 2.55% and 3.16% compared with those in December 2018. Moreover, slightly higher increases of 3.19% and 4.45% were reported for evening peaks during weekdays and weekends, respectively. Therefore, the phenomenon implies that proper settings of signal hyperparameters enhanced traffic efficiency over the deployed adaptive signal control systems.

V. Conclusion

Although agile and adaptive urban traffic control systems have been deployed in cities, many of them still need human involvement when traffic patterns vary in a short-term period, which is still small in scale for city-level urban control. In this work, a novel end-to-end recommendation system under the PL framework has been introduced for urban traffic controls incorporating the significance of human knowledge. The system utilizes deep learning techniques with optimally determined model types and neural architectures to imitate engineers' preferences for the generation of control hyperparameters. A real-world case was tested in Hangzhou, China and the recommended control hyperparameters of an adaptive signal control system were in line with human adjustment experiences. After the deployment of the recommendation system, traffic efficiencies are improved for the Hangzhou urban areas on average.

Currently, instantaneous event signals are solely based on traffic analytics of local intersections, although appropriate control hyperparameters can improve and maintain system performance at a certain level. The recommendation system will be further developed in two ways by integrating predictive learning and perspective learning. First, a large amount of data is generated to reveal the consequences of certain policies upon the recommendation system with the concept of predictive learning. The data augmentation process is based on a generative adversarial network-based architecture [36]. Second, the system will optimally evolve using a special trial-and-error manner instructed by perspective learning. An optimal control policy will be generated by a deep reinforcement learning algorithm to guide the signal control engineer professionals and, hence, promote the recommendation system with growing knowledge [37].

References

[1] P. Hunt, D. Robertson, R. Bretherton, and M. C. Royle, "The SCOOT on-line traffic signal optimisation technique," *Traffic Eng. Control*, vol. 23, no. 4, pp. 190–192, 1982.

[2] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Trans. Veh. Technol.*, vol. VT-29, no. 2, pp. 130–137, May 1980.

[3] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a connected vehicle environment," *Transp. Res. C, Emerg. Technol.*, vol. 55, pp. 460–473, Jun. 2015.

[4] X.-S. Hua, "The city brain: Towards real-time search for the real-world," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 1343–1344.

[5] AutoNavi. (2018). *Traffic Analysis Report, Q3 2018*. [Online]. Available: https://trp.autonavi.com/download.do

[6] L. Li, Y. Lin, N. Zheng, and F.-Y. Wang, "Parallel learning: A perspective and a framework," *IEEE/CAA J. Automat. Sinica*, vol. 4, no. 3, pp. 389–395, Jul. 2017.

[7] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[8] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[9] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 485–497, Jun. 2010.

[10] Y. Wang, X. Yang, H. Liang, and Y. Liu, "A review of the self-adaptive traffic signal control system based on future traffic environment," *J. Adv. Transp.*, vol. 2018, pp. 1–12, Jun. 2018.

[11] J. Jin, X. Ma, and I. Kosonen, "A stochastic optimization framework for road traffic controls based on evolutionary algorithms and traffic simulation," *Adv. Eng. Softw.*, vol. 114, pp. 348–360, Dec. 2017.

[12] Q. Guo, L. Li, and X. J. Ban, "Urban traffic signal control with connected and automated vehicles: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 313–334, Apr. 2019.

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

[14] J. Jin and X. Ma, "Hierarchical multi-agent control of traffic lights based on collective learning," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 236–248, Feb. 2018.

[15] Y. Zhao, H. Gao, S. Wang, and F.-Y. Wang, "A novel approach for traffic signal control: A recommendation perspective," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 127–135, Jul. 2017.

[16] S. Feng, X. Wang, H. Sun, Y. Zhang, and L. Li, "A better understanding of long-range temporal dependence of traffic flow time series," *Phys. A, Stat. Mech. Appl.*, vol. 492, pp. 639–650, Feb. 2018.

[17] C. Yu, Y. Feng, H. X. Liu, W. Ma, and X. Yang, "Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections," *Transp. Res. B, Methodol.*, vol. 112, pp. 89–112, Jun. 2018.

[18] J. Jin and X. Ma, "A group-based traffic signal control with adaptive learning ability," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 282–293, Oct. 2017.

[19] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," 2017, *arXiv:1707.07435*. [Online]. Available: http://arxiv.org/abs/1707.07435

[20] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 191–198.

[21] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[22] E. Diaconescu, "The use of NARX neural networks to predict chaotic time series," *WSEAS Trans. Comput. Res.*, vol. 3, no. 3, pp. 182–191, Mar. 2008.

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

[24] K. Cho *et al.*, "Learning Phrase Representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[25] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[26] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017. [Online]. Available: https://openreview.net/forum?id=r1Ue8Hcxg

[27] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.

[28] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4095–4104.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[30] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[31] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[32] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2249–2255.

[33] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2627–2633.

[34] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 95–104.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., May 2015.

[36] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Automat. Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.

[37] J. Jin and X. Ma, "A multi-objective agent-based control approach with application in intelligent traffic signal system," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3900–3912, Oct. 2019.

Junchen Jin received the B.Eng. degree in traffic engineering from Beijing Jiaotong University, Beijing, China, and the M.Sc. and Ph.D. degrees in transport science from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2014 and 2018, respectively. He holds the Vice Director position at the Smart Transportation Research Institute, Enjoyor Co., Ltd., Hangzhou, China. He is also a Post-Doctoral Researcher with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include intelligent transport systems, traffic simulation and control, recommender systems, artificial intelligence, deep learning, and reinforcement learning.

Haifeng Guo received the Ph.D. degree in traffic information and control from Jilin University in 2008. From 2011 to 2013, he was a Post-Doctoral Research Associate with the Zhejiang University of Technology, Hangzhou, China, where he is currently an Associate Professor with the College of Information Engineering. He is also affiliated with Enjoyor Co., Ltd., Hangzhou. His research interests broadly lie in traffic operation and control, smart transportation systems, and traffic AI.

Jia Xu received the M.Sc. and Ph.D. degrees in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2016, respectively. He holds the Vice Director position at the Smart Transportation Research Institute, Enjoyor Co., Ltd., Hangzhou, China. He is also a Post-Doctoral Researcher with the Institute of Intelligent Transportation System, Zhejiang University, Hangzhou. His research interest includes big data, traffic control and optimization, and artificial intelligence.

Xiao Wang (Member, IEEE) received the B.S. degree in network engineering from the Dalian University of Technology, Dalian, China, in 2011, and the Ph.D. degree in social computing from the University of Chinese Academy of Sciences, Beijing, China, in 2016. She is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. Her current research interests include social transportation, cyber movement organization, artificial intelligence, and social network analysis.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined the University of Arizona, Tucson, AZ, USA, as a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program from the State Planning Council and 100 Talent Program from CAS. In 2002, he joined the Laboratory of Complex Systems and Intelligence Science, CAS, as the Director, where he was the Vice President of Research, Education, and Academic Exchanges, Institute of Automation, from 2006 to 2010. In 2011, he was a State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems, Beijing. His current research interests include methods and applications for parallel systems, social computing, parallel intelligence, and knowledge automation.

Dr. Wang was elected as a fellow of INCOSE, IFAC, ASME, and AAAS. He was a recipient of the National Prize in Natural Sciences of China in 2007, the Outstanding Scientist Award from ACM for his research contributions in intelligent control and social computing in 2007, the IEEE Intelligent Transportation Systems (ITS) Outstanding Application and Research awards in 2009, 2011, and 2015, and the IEEE SMC Norbert Wiener Award in 2014. He was the General Chair or Program Chair of more than 30 IEEE, INFORMS, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, and the American Zhu Kezhen Education Foundation from 2007 to 2008. He was the Vice President of the ACM China Council from 2010 to 2011 and the Chair of IFAC TC on Economic and Social Systems from 2008 to 2011. He is currently the President-Elect of the IEEE Council on Radio Frequency Identification (RFID). Since 2008, he has been the Vice President and the Secretary General of the Chinese Association of Automation. He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000 and the *IEEE Intelligent Transportation Systems Magazine* from 2006 to 2007. He was the Editor-in-Chief of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012 and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS and the Founding Editor-in-Chief of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA and the *Chinese Journal of Command and Control*.