

An Automated Hyperparameter Search-Based Deep Learning Model for Highway Traffic Prediction

Hongsuk Yi and Khac-Hoai Nam Bui^{ID}

Abstract—Auto machine learning recently has been introduced as a trending technique for learning applications, including smart transportation. In this study, we focus on applying auto-machine learning for hyperparameter tuning to learn traffic datasets at the main regions of highway systems. Particularly, deep learning models have been recently introduced as emergent methods for traffic prediction. However, training deep learning models requires expensive works (e.g., time-consuming and human expertise), especially in terms of determining the configurations of hyperparameters in the models. In this regard, this paper introduces an automated framework for hyperparameter tuning to learn traffic datasets at an ecosystem in terms of reducing time-consuming tasks. Specifically, we first propose the *HyperNet* framework, using advanced data science techniques (e.g., Bayesian optimization and meta-learning) for the automated hyperparameter search process. Then, a deep learning model with the long short term memory network based on the *HyperNet* framework has presented for learning the temporal variation of traffic datasets at main regions of highway traffic systems. Regarding the experiment, we take data from the Korean highway system into account as a case study to evaluate the proposed approach. The evaluation indicates promising results of the proposed framework for learning multiple datasets of the traffic highway systems.

Index Terms—Highway traffic prediction, deep learning, automated machine learning, long-short term memory, hyperparameters optimization, meta learning.

I. INTRODUCTION

As the ever-increasing traffic demand, cities around the world have to face the traffic congestion problem. Traffic flow on the highway system is no exception. For instance, traffic congestion has occurred on the main roads at rush hours such as weekend morning and evening [1]. Hence, with the development of the Intelligent Transportation System (ITS), where a large amount of data generated every day, many approaches have been proposed to deal with traffic congestion problems based on data collection from ITS [2], [3]. In the

Manuscript received June 4, 2019; revised October 13, 2019, December 29, 2019, and February 25, 2020; accepted April 9, 2020. This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) Grant funded by the Korea Government (Development of Deep Learning-Based Urban Traffic Congestion Prediction and Signal Control Solution System) under Grant MSIT 2018-0-00494 and in part by the Korea Institute of Science and Technology Information (KISTI) Grant funded by the Korea Government under Grant MSIT K-19-L02-C07-S01. The Associate Editor for this article was X. Si Ma. (Corresponding author: Khac-Hoai Nam Bui.)

The authors are with the Korea Institute of Science and Technology Information, Daejeon, 34141, South Korea (e-mail: hsyi@kisti.re.kr; hoainam.bk2012@gmail.com).

Digital Object Identifier 10.1109/TITS.2020.2987614

case of highway systems, traffic flow prediction becomes the main component for improving the traffic flow [4]. However, the rapid growth of generated data becomes an emergent challenge that the traditional processing systems are not able to deal with the data analytics requirements [5].

Recently, Deep Learning (DL), an evolution of Machine Learning (ML), which contains multiple hidden layers, has been introduced as a promising method to improve the accuracy of traffic prediction [6]. Specifically, some well-known DL models (e.g., Deep Belief Networks (DBNs), Stacked Autoencoder (SAE), Convolution Neural Networks (CNN), and Long-Short Term Memory (LSTM)), a variant of Recurrent Neural Network (RNN), have demonstrated capabilities for traffic prediction. For instance, Yisheng *et al.* [7] have proposed a DL model for traffic flow prediction using SAE to train generic features of traffic data. In [8], Huang *et al.* have introduced a DBN model for multitask learning with promising results. Furthermore, the work in [9] indicates that LSTM networks can achieve good performance for time series forecasting of traffic data. Nevertheless, configuring neural networks is still a challenging problem in which we have to explore different configurations to provide an appropriate model. Moreover, with the integration of Artificial Intelligence (AI), DL methods emerge as potential solutions in terms of shifting to computational intelligence approaches which are able to automatically turn hyperparameters with the best configurations [10]. Those aforementioned issues are still open researches in this field.

Particularly, the successful applications of DL are currently far from the fully automated process since according to *No Free Lunch* theorems, there are no algorithms/configurations that can archive good performance with all possible problems. Therefore, human experts play a critical role for the development of ML applications (e.g., Feature extraction, model/algorithm selections, and hyperparameter tuning) [11]. Furthermore, training DL requires expensive works since the models involve a highly iterative process (e.g., high-dimensional hyperparameters) which are frustratingly difficult and time-consuming tasks [12]. In this regard, the trend research focuses on the automated training process to determine appropriate values of these hyperparameters for training DL models with acceptable time [13]. Specifically, Fig. 1 depicts the process of Auto Machine Learning (AutoML) for the training process.

In order to overcome the aforementioned problems to apply DL models for training the temporal variation of traffic data

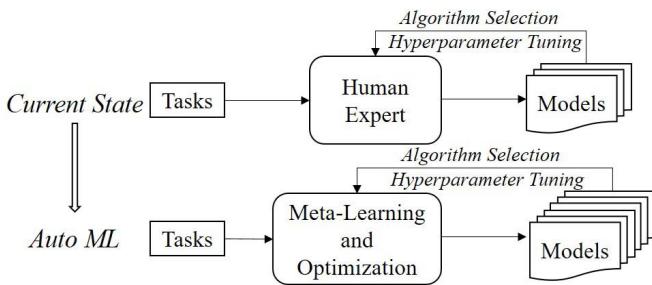


Fig. 1. AutoML for training process.

on highway systems, this study focuses on developing an automated training process for predicting traffic conditions at the main regions of a certain highway system. In particular, we take the automated hyperparameters tuning into account to determine the optimal configuration for the LSTM model to analyze highway traffic time-series datasets. Specifically, several advanced techniques such as Bayesian and meta-learning have been applied for drastically speeding up the hyperparameter search process [14]. To the best of our knowledge, this paper is the first study to apply automated hyperparameter tuning problems for training traffic datasets. Specifically, our contributions include as follows:

- We collect and analyze data of the Korean highway system, which are important routes of public transportation in Korea, to predict the congestion in each particular region. Specifically, in highway systems, the traffic flow patterns (e.g., rush and non-rush hours) are quite similar at several specific locations. Therefore, a promising meta-learning method can be applied in order to reduce significantly the time consumption for learning traffic datasets at highway systems [15].
- We propose an automated framework, which is named *HyperNet* for hyperparameter tuning to train large-scale traffic datasets in terms of enabling automated processes and reducing time-consuming tasks. Particularly, we apply state-of-the-art data analytics techniques such as Bayesian optimization and meta-learning for tuning hyperparameters of LSTM, a recent DL model which has proved the capabilities for training time-series traffic datasets [16]. In addition, the framework can be extended for other DL models which depends on the characteristic of the input datasets and the target of the predictions (e.g., large-scale traffic forecasting at an urban area).
- The implementations indicate that the proposed approach is able to provide promising results for the traffic congestion prediction at the main regions (entrance/exit area) of the highway systems. Technically, the proposed approach is an important step to develop the AutoML for learning applications which is a state-of-the-art technique in this research area [17].

The rest of this paper is organized as follows: In Section 2, we present some literature reviews of traffic prediction using DL models. Moreover, the hyperparameter optimization problem has been introduced in this section. In Section 3,

we propose *HyperNet* framework for short term traffic speed prediction. Consequently, applying the proposed framework for traffic prediction on highway systems is described in Section 4. The experimental results compared with previous works are shown in Section 5. Section 6 includes some discussions and future works of this study.

II. LITERATURE REVIEW

A. Long-Short Term Memory for Time Series Prediction

Traffic prediction has been regarded as the main component in ITS. The ARIMA model first introduced to predict the short-term freeway traffic flow. Since then, many models have been proposed for traffic prediction [4]. Recent studies reveal that DL models, which include multi-network layers work together, are promising approaches for traffic prediction. Specifically, It depends on the assumptions of training datasets, there are several well-known DL methods that can use to develop time series forecasting such as Multilayer Perceptrons(MLP), CNN, and LSTM models [7], [9]. Furthermore, hybrids models (e.g., CNN-LSTM and ConvLSTM) have been introduced for traffic prediction in which traffic data have processed as images for the input [18], [19]. Recently, Graph Neural Networks (GNN) has proposed as a promising concept for improving the ability to learn hidden patterns from spatial-temporal graphs, which is an important issue for various applications such as human action recognition and traffic prediction [20].

In this study, we use the LSTM model, which have been proved the capability for learning the temporal variation in ITS [21], to train the collected datasets on highway systems. Technically, LSTM model is a variant of RNN architecture that is able to learn information with long time spans and determine the optimal time lags in an automatic manner. Particularly, the LSTM cell comprises an input gate x , an output gate o , and a forget gate f for regulating the flow of information (e.i., into and out) of the cell. Therefore, LSTM is able to overcome the vanishing gradient problem that occurs with most RNN models. Specifically, Fig. 2 depicts the difference between LSTM and standard RNN structures.

Supporting the input X is the historical traffic data which is denoted as follows:

$$X = \{x_1, x_2, \dots, x_n\} \quad (1)$$

The hidden state of memory cells H and predict output time series Y are respectively determined as follows:

$$\begin{aligned} H &= \{h_1, h_2, \dots, h_n\} \\ Y &= \{y_1, y_2, \dots, y_n\} \end{aligned} \quad (2)$$

where each $h_t \in H$ and $y_t \in Y$ can be sequentially calculated as follow:

$$\begin{aligned} h_t &= H(W_h x_t + W_h' h_{t-1} + b_h) \\ y_t &= W_y h_t + b_y \end{aligned} \quad (3)$$

where W and b denotes weight matrices and bias vectors, respectively. Consequently, the hidden state of memory

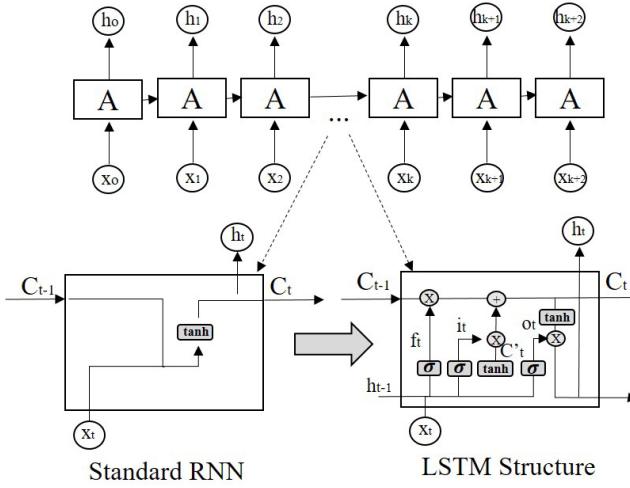


Fig. 2. The difference between traditional RNN and LSTM cell.

cells H is computed based on the formulations of each gate in the cell:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (4)$$

where σ and C denote the sigmoid function and activation vectors, respectively, which can be sequentially calculated as follows:

$$\begin{aligned} \sigma(x) &= \frac{1}{1 + e^{-x}} \\ C_t &= f_t * C_{t-1} + i_t * \widehat{C}_t \end{aligned} \quad (5)$$

where \widehat{C} denotes the candidate value in each computed process of the cell which can be calculated as follows:

$$\widehat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (6)$$

Nevertheless, the limitation of LSTM model is that there is no direction from Constant Error Carousels (CECs) to LSTM cell for supporting to control which means the gate layers are not able to look at the cell state. Hence, in this study, we apply the work in [22] which has added ‘peepholes connection’ to all the gates. Therefore, the gates of the LSTM cell in Eq. 4 can be re-calculated as follows:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t, C_{t-1}] + b_i) \\ f_t &= \sigma(W_f \cdot [h_{t-1}, x_t, C_{t-1}] + b_f) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t, C_t] + b_o) \end{aligned} \quad (7)$$

B. Hyperparameters Optimization in Deep Learning Models

Determining the best configuration is one of the most challenges in implementing machine learning (ML) solutions. Normally, we minimize the testing error by regularly modifying the model during programming as shown in Fig. 3. However, applying this issue for DL models is computationally expensive since difference with conventional ML, DL involves

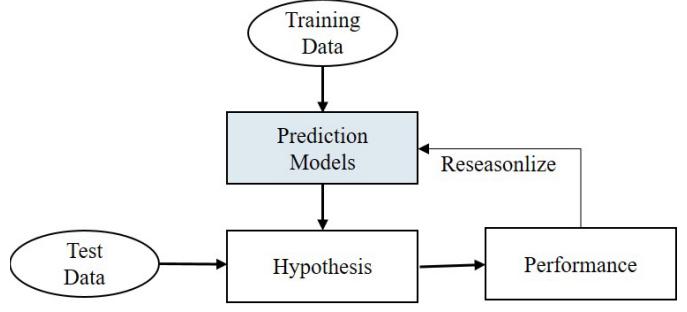


Fig. 3. The predictive analytics process.

TABLE I
SEARCH STRATEGIES FOR HYPERPARAMETERS OPTIMIZATION

Approach	Type	Space Exploration	Parallel	History
Manual Search	Manual	Low	No	Yes
Grid Search	Auto	High	Yes	No
Random Search	Auto	Medium	Yes	No
Bayesian Search	Auto	Medium-Driven	Yes	Yes

many hyperparameters. Recently, automated hyperparameter optimization (HPO) for DL models has been introduced with promising results [23]. Although there has not a clear definition, hyperparameters are generally described as follows:

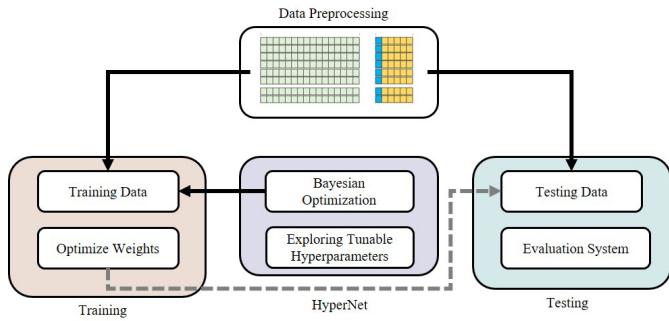
Definition 1: In machine learning, hyperparameters are variables that define network structure and determine how the model is trained. They are specified before beginning the learning process.

Technically, searching for the best configuration of hyperparameters is an iterative process. Currently, there have four well-known strategies for searching the configuration which include: i) Manual Search; ii) Grid Search; iii) Random Search; and iv) Bayesian Search. Specifically, Tab. I shows the properties of each strategy [24]. In our framework, we apply Bayesian optimization approach for automated searching of the hyperparameters due to following reasons:

- The approach has been proved the efficient and robust automated capability for hyperparameter tuning in recent works [23].
- Grid and Random searches have a limitation in which the new prediction is independent with previous runs. In this regard, using the Bayesian approach is able to take less training steps and achieve a comparable result with a sufficiently high number of experiments.

C. Meta Learning for Fast Automated Machine Learning

HPO using the Bayesian approach has been shown good performance for the hyperparameter tuning problem, especially in case of high dimensions of the hyperparameters [25]. However, applying this approach to real-life problems still remain several challenging issues since Bayesian hyperparameter optimization requires a substantial number of evaluations to detect high-performance regions when started on a new optimization problem. Especially, in case of the transportation domains, where multiple datasets have collected and processed from various sources in the system (e.g., Sensors on the road).

Fig. 4. Proposed system for traffic prediction using *HyperNet* framework.

In this regard, meta-learning (learning to learn) has introduced to address the aforementioned problems in which ML algorithms are developed to learn from data without being explicitly programmed [26]. Consequently, recent works have successfully applied meta-learning for DL models (i.e., Neural Architecture Search (NAS) and hyperparameter tuning) to automate the learning process and reduce time-consuming tasks [27], [28].

In this study, we take hyperparameter tuning using meta-learning into account for multiple traffic datasets on highway systems. Specifically, the key concept by applying meta-learning for the hyperparameter search is to suggest the configuration for a new similar dataset based on the configurations that are known from previous evaluations. This concept is suitable for the datasets from highway traffic systems in which multiple traffic datasets have collected from different sensors locating on the same route and the traffic flow patterns are quite similar at several specific locations. In this regard, although this paper implements the LSTM model as the case study, however, we are able to apply the proposed framework for several state-of-the-art DL models of traffic forecasting such as ConvLSTM and GNN. The main difference is that based on the characteristic of each model, we have to determine the number of hyperparameters and their search spaces for turning as the input of the framework.

III. HYPERNET FRAMEWORK FOR AUTOMATED HYPERPARAMETER SEARCH

A. Problem Formulation

Fig. 4 illustrates the overview of our proposed framework for traffic prediction in ITS. Specifically, according to the Def. 1, hyperparameters are automatically set before training regarding weight optimization.

Supporting $\theta = \theta_1, \theta_2, \dots, \theta_m$ and $\Theta = \Theta_1 \times \Theta_2 \times \dots \times \Theta_m$ denote the set and configuration space of the hyperparameters, respectively, the objective function of *HyperNet* is to find the hyperparameters of a given model that return the best performance as measured on a validation set. In this regard, the objective function can be formulated as follows:

$$\min_{\theta \in \mathbb{R}^\Theta} f(\theta, \mathcal{D}_{val}) \quad (8)$$

where $f(\Phi, \mathcal{D}_{val})$ represents the objective score to minimize regarding validation dataset \mathcal{D}_{val} . Therefore, the

Algorithm 1 Sequential Model-Based Optimization
SMBO(f, θ, Θ, T)

Data: Target function f , Hyperparameters θ , Search Space Θ , Iteration T

Result: Best Hyperparameters θ^*

```

1 Initialize model  $\mathcal{M}$ ;
2  $\mathcal{H} \leftarrow \emptyset$ 
3 for  $t \leftarrow 1$  to  $T$  do
4    $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} EI(\theta, \mathcal{M}_{t-1})$  ;
5   Evaluate  $f(\theta^*)$  ;
6    $\mathcal{H} \leftarrow \mathcal{H} \cup (\theta^*, f(\theta^*))$  ;
7   Fit model  $\mathcal{M}_t$  by  $\mathcal{H}$ 
8 end
9 Return  $\theta^*$  given  $\mathcal{H}$ 

```

hyperparameter optimization problem can be formulated as follows:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} f(\theta, \mathcal{D}_{train}) \quad (9)$$

which means that we need to determine the maximum value θ^* from training datasets \mathcal{D}_{train} .

B. Bayesian Hyperparameter Optimization for a Single Dataset

For solving objective function in Eq. 9, we apply Bayesian Optimization (BO) which has recently introduced as a promising approach for the hyperparameter optimization problems. Specifically, Sequential Model-Based Optimization (SMBO), a formalization of BO, has emerged as a successful method for hyperparameter tuning [23]. Technically, the idea of this approach is to place a surrogate model \mathcal{M} over the target black-box, and update it sequentially by querying $f(\theta)$ at new points that optimize an acquisition function by using the expected Positive Improvement (EI) which can be calculated as follows:

$$EI_{y^*}(\theta, \mathcal{M}) = \int_{-\infty}^{y^*} \max(y^* - y, 0) p_{\mathcal{M}}(y|\theta) dy \quad (10)$$

where y and y^* denotes the actual and threshold values, respectively. More detail of the generic SMBO is demonstrated in Algorithm 1.

C. Exploring Tunable Hyperparameters via Meta Learning for Multiple Datasets

Since SMBO is regarded as a generic function optimization framework, scalability becomes a serious problem, especially in case the transportation domain which involves multiple datasets from various source. In this regard, we propose a new approach for training multiple datasets using meta-learning in which focus on determining the initialization of tunable hyperparameters in the SMBO algorithm [15] in which we refer to the new algorithm for training multiple datasets as M-SMBO.

Considering a set of similar datasets $\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^n$ (e.g., datasets from an ecosystem) in which each dataset divides

into a target variable Y and a feature vector X . In this regard, a predictive model has produced for learning the function relationship between X and Y . Particularly, the function is defined by a performance metric that is controlled by m-dimension hyperparameter configuration θ . Therefore, the objective of M-SMBO is to find the best hyperparameter configuration for a new dataset \mathcal{D}^{k+1} based on the set of benchmark datasets $\mathcal{D}^1, \mathcal{D}^2 \dots \mathcal{D}^k$. Specifically, this approach includes three steps as follows:

1) *Benchmark Datasets*: In order to apply meta-learning for hyperparameter tuning, we first gather meta-data that evaluate on given k ($k \in n$) datasets. Each dataset is supported to optimize hyperparameters using the SMBO function (Algorithm 1). Specifically, given a dataset \mathcal{D}^i ($i \in k$), the hyperparameter configurations is to minimize the loss function \mathcal{L} as follows:

$$\theta^{*,i} = \underset{\theta \in \Theta}{\operatorname{argmin}} \mathcal{L}(\theta, \mathcal{D}^i) \quad (11)$$

where $\mathcal{L}(\theta, \mathcal{D}^i)$ indicates the loss function of dataset \mathcal{D}^i using hyperparameter θ . In this study, we determine the benchmark datasets based on several analysis techniques on all datasets. Furthermore, since BO contains random factors, we execute 10 times on each benchmark dataset.

2) *Default Hyperparameter Configuration*: Default configurations refer to the hyperparameters that work well across multiple datasets. Normally, they have provided by software packages (e.g., Auto-Weka [29]). In this study, we define the default hyperparameters based on the empirical experiments from benchmarks datasets. Specifically, a subset hyperparameter θ^s ($\theta^s \in \Theta$) is determined as a default configuration in case of satisfying the following equation:

$$\theta^s = \underset{\theta \in \Theta}{\operatorname{argmin}} \bar{h}(\mathcal{L}(\theta^{*,1}, \mathcal{D}^1), \mathcal{L}(\theta^{*,2}, \mathcal{D}^2), \dots, \mathcal{L}(\theta^{*,k}, \mathcal{D}^k)) \quad (12)$$

where the function \bar{h} (e.g, median) has used to perform the single scalar value [30].

3) *Search Space for Hyperparameter Tuning*: BO has attracted many studies comparing with other approaches such as grid search and random search since BO can use the information gained by all the previous tries to explore the search space of hyperparameters in an efficient way [25]. However, this process is still expensive work. In this regard, we try to determine the reasonable search space Θ^* ($\Theta^* \in \Theta$) by optimal the configuration of each hyperparameter θ_k with high probability for the new datasets. Therefore, we apply the quantile function to determine the threshold of the search space [31]. In this regard, supporting $q_{i,p}$ denotes the quantile of the discrete distribution of hyperparameter θ_i , the optimal search domain Θ^* can be defined as follows:

$$\Theta^* = \{\theta \in \Theta \mid \forall i \in 1 \dots m : q_{i,p_1} \leq \theta_i \leq q_{i,p_2}\} \quad (13)$$

where p_1 and p_2 denote the upper and lower quartile values, respectively. Note that this concept can be applied for both *numerical* and *categorical* hyperparameters. Particularly, Algorithm 2 depicts the pseudo-code of the M-SMBO algorithm for hyperparameter search across multiple datasets.

Algorithm 2 Meta-Learning-Based Hyperparameter Tuning for Traffic Datasets M-SMBO($D^{k+1}, f, \theta^{*,1:k}, \theta, \Theta, T$)

Data: New Dataset D^{k+1} , Target function f , Hyperparameter θ , Best Hyperparameters of benchmark Datasets $\theta^{*,1:k} = \theta_1^*, \theta_2^* \dots \theta_k^*$, Search Space Θ , Iteration T , upper and lower quantile values p_1, p_2

Result: Best Hyperparameters $\theta^{*,k+1}$

1 **Set** $\theta^{s,k+1} = \underset{\theta \in \Theta}{\operatorname{argmin}} \bar{h}(\mathcal{L}(\theta^{*,1}, \mathcal{D}^1), \mathcal{L}(\theta^{*,2}, \mathcal{D}^2), \dots, \mathcal{L}(\theta^{*,k}, \mathcal{D}^k))$;

2 $\theta^{new,k+1} = \theta / \theta^{s,k+1}$;

3 **Set** $\Theta^{new,k+1} = \{\theta^{new,k+1} \in \Theta \mid \forall i \in 1 \dots m : q_{i,p_1} \leq \theta_i \leq q_{i,p_2}\}$

4 **Return** $\theta^{*,k+1} \leftarrow \theta^{s,k+1} \wedge \text{SMBO}(f, \theta^{new,k+1}, \Theta^{new,k+1}, T)$

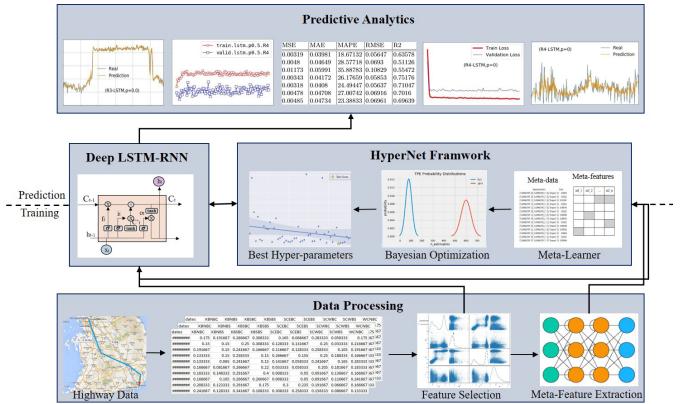


Fig. 5. System architecture for traffic prediction on highway system.

IV. HYPERNET FRAMEWORK-BASED DEEP RNN-LSTM MODEL FOR TRAFFIC PREDICTION ON HIGHWAY SYSTEM

A. System Architecture

Fig. 5 depicts the system architecture of our approach for traffic prediction of highway system based on *HyperNet* framework. Specifically, our approach includes three main processes: i) Data Processing; ii) *HyperNet* Framework-based Deep LSTM-RNN Model; iii) Predictive Analytics.

1) *Data Processing*: This process is proposed for understanding traffic data that are collected from the highway system. Among having many features, we focus on three important issues as follows:

- Data pre-processing: performing statistical checks for the quality of the collected data in terms of avoiding serial correlation where error terms in a time series transfer from one period to another.
- Features selection: defining features (e.g., main points) that has explanatory power to predict and using them for the training in DL models.
- Meta-features extraction: since we only consider training DL model for traffic datasets at a certain area of highway systems, we assume that simple meta-features such as

the number of features, traffic patterns (e.g., time interval and rush hours), and structure of all considered datasets are similar. Therefore, defining benchmark datasets and calculating the similarity among datasets also include in this process. For instance, the input datasets from sensors which are located on the same route with the same level of traffic condition can be regarded as similar datasets. More detail of this concept is explained in the following section with the real data that we have collected in this study.

2) HyperNet Framework-Based Deep LSTM-RNN Model:

In this study, we apply an LSTM network which has been proved the capabilities for predicting traffic data [9], [16]. However, the difference with previous works, instead of manually configuration, we proposed a framework for training the LSTM model in which the hyperparameters are automatically turned. In this regard, this process is able to improve the performance of the training model in terms of accuracy and time complexity.

3) Predictive Analytics: This process demonstrates the performances of the proposed model in terms of minimizing the square error. Specifically, we use 70% daily data to train (train data), 10% of the validation set and predict the next 20% (test data). Then, we will compare the predicted results with the test data.

B. Objective Function

As we mentioned above, by applying the automated hyperparameters tuning problem, the objective function of this study is to minimize the square error for predicting traffic flow, which is formulated as follows:

$$e = \sum_{t=1}^n (y_t - p_t)^2 \quad (14)$$

where e denotes the square error value and p is the predicted traffic flow value. Consequently, two performance indexes for evaluating the model are taken into consideration which include: i) mean square error (MSE); and ii) mean absolute percentage error (MAPE). Specifically, the performance indexes are sequentially calculated as follows:

$$\begin{aligned} MSE &= \frac{1}{n} \sum_{i=1}^n (f_i - \hat{f}_i)^2 \\ MAPE &= \frac{1}{n} \sum_{i=1}^n \frac{|f_i - \hat{f}_i|}{f_i} \end{aligned} \quad (15)$$

where f and \hat{f} are observation and prediction values, respectively.

C. LSTM Hyperparameters for Tuning

The challenging issue by using DL models is that they include many hyperparameters. Specifically, adjusting all hyperparameters can make a complicated process. Moreover, search algorithms (Tab. I) treat hyperparameters equality to find the best results which can make time-consuming and computationally expensive problems [32].

TABLE II
HYPERPARAMETERS TUNING FOR TRAINING BIG TRAFFIC ATA

Hyperparameter	Relative	Description	Type/Search space
Learning rate	Training algorithm	A configuration by adjusting the weights based on the error gradient.	Float / Log-uniform [0.005, 0.2]
Hidden layers	Network structure	The layers between Input and Output layers to increase the accuracy.	Int / [0,2]
Number of neurons	Network structure	The units within a hidden layer with regularization techniques to increasing the accuracy.	Int / Log-uniform [20, 200]
Batch size	Training algorithm	Defining the number of samples that will be propagated through the process.	Int / [1, 512]
Dropout	Network structure	Reducing overfitting in neural networks.	Float / [0, 1]

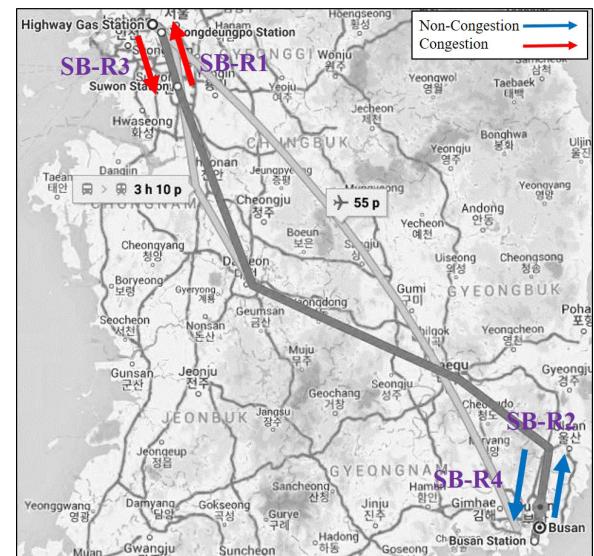


Fig. 6. Frequency of TPI values in each region.

Recently, several methods have introduced the importance of hyperparameters for the training process [27], [33]. In this study, regarding the traffic data from the highway system, based on the experience in our previous work [16], [34], five hyperparameters are considered for tuning in LSTM models such as learning rate, number of LSTM layers, number of neurons, batch size, and dropout. Particularly, Tab. II depicts the explanations and search spaces of the aforementioned hyperparameters.

V. IMPLEMENTATION

A. Data Description and Processing

Regarding the data for the evaluation, we collected data from four main routes of Korea highway systems such as Gyeongbu (SB), Gyeongchun (SC), Seohaean (SM), and Yeongdong (IG) routes. Specifically, in each route, we take four major regions, which include Northbound and Southbound of two terminus, into account. For instance, Fig. 6¹

¹<https://www.google.com/maps/>

TABLE III
FORMAT OF HIGHWAY TRAFFIC DATASETS

Variable	Type	Notation
Route	Spring (ID)	$ROUTE_{ID}$
Location	Spring (ID)	LOC_{ID}
Time	YYYYMMDDhhmm	REG_{HM}
TPI	Integer	TPI

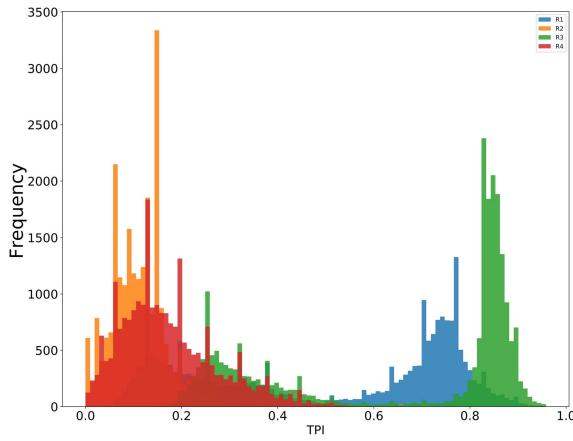


Fig. 7. Frequency of TPI values in each region of SB route.

depicts the map and main regions of SB highway that we consider in this study.

Hence, our evaluation totally includes 16 input datasets. Consequently, each dataset obtains the average traffic speed every 5 minutes which have collected from May to July, 2017 (3 months). Therefore, the usage data includes 23616 average traffic speed data which means that 288 average traffic speed data had aggregated every day and stored them for 82 days excepting holidays. In particular, Tab. III depicts the format of highway traffic data that we have processed before training.

Specifically, since there have several posted limited speed from 80 KPH (Kilometers per hour) to 110 KPH, the 5 minutes average speed has converted to the Traffic Performance Index (TPI) [35], which is calculated as follows:

$$TPI_t = \frac{v_f - v_t}{v_f} \quad (16)$$

where v_f , v_t represents the maximum speed and the average traffic speed in each 5 minutes, respectively. Particularly, the maximum speed is determined by calculating the speed of a vehicle without interruption by any surrounding vehicles. Basically, the maximum speed can be calculated as follows:

$$v_f = c * v_l \quad (17)$$

where v_l denotes the posted limited speed of a specific link and c is a constant number which can be varied from 1.2 to 1.5. Consequently, by converting the 5 minutes average speed to TPI values, the data is logically normalized which is used as the input data for the prediction.

Fundamental analysis from data collection shows the traffic conditions in each main region of a route. For instance, in the case of SB highway, traffic congestion has occurred at regions that are nearby Seoul station (SB-R1 and SB-R3) as shown in Fig. 6. Specifically, Fig. 7 shows the frequency of TPI values

TABLE IV
TRAFFIC CONDITIONS OF ALL CONSIDERED REGIONS

Route	Region	Traffic Condition
SB	R1, R3	Congestion
	R2, R4	Non-Congestion
SC	R1, R3	Congestion
	R2, R4	Non-Congestion
SM	R1, R3	Congestion
	R2, R4	Non-Congestion
IG	R1, R3	Congestion
	R2, R4	Non-Congestion

in each region of SB highway. Technically, in continuous traffic flow such as highway systems, the TPI values less than 0.4 and greater than 0.8 represents *non-congested condition* and *serious congested condition*, respectively.

For more detail, Tab. IV indicates the traffic conditions of all datasets in each route that we take into count for the evaluation. In this regard, for training multiple datasets, we will take two regions that are in different traffic condition level in each route (e.g., R1 and R4) for the benchmark datasets.

B. Experimental Settings

For the implementation, we use *Keras*,² a python DL library for developing our model. The experiments work well by a PC with Core i7 16-GB CPU and 32GB GPU memories in which we used the GPU for acceleration. Specifically, our implementation includes three main processes as follows:

- *Hyperparameter tuning*: We first implement an auto hyperparameter search based on Bayesian optimization for determining hyperparameters values which are regarded as the best configuration.
- *Traffic Speed Prediction*: By applying a Deep LSTM-RNN model with the aforementioned hyperparameters, the results are the prediction values of TPI for the next time interval in each considered region.
- *Comparison*: For evaluating the effectiveness of the proposed approach, we compare Bayesian Optimization with *Random Search* in terms of both accuracy and time-consuming tasks. Furthermore, we also evaluate our proposed framework for hyperparameter search across multiple datasets to show the capability of the scalability of the proposed approach for traffic prediction on highway systems.

More detail of the experimental settings are described as follows:

- *Input and Output*: The input includes the average traffic speed of vehicles in previous time intervals. The output is the traffic speed prediction in the next time interval (short-term prediction) of each certain region.
- *Training and Testing Data*: Data have been collected for 82 days (excepting holidays) in 2017. In this regard, for each dataset, we use the data of the first 66 days as the training set (including 8 days for the validation) and the last 16 days as the testing set.

²<https://keras.io/>

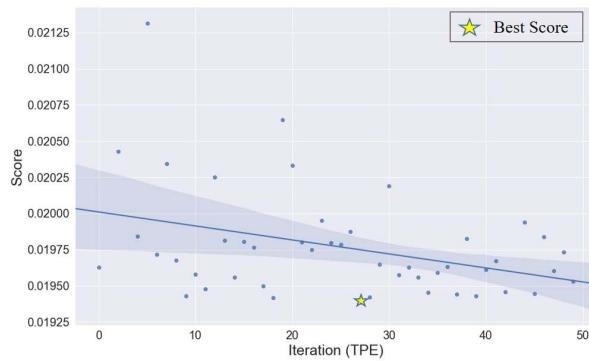


Fig. 8. Best found values of SB-R4 datasets using Bayesian hyperparameter optimization.

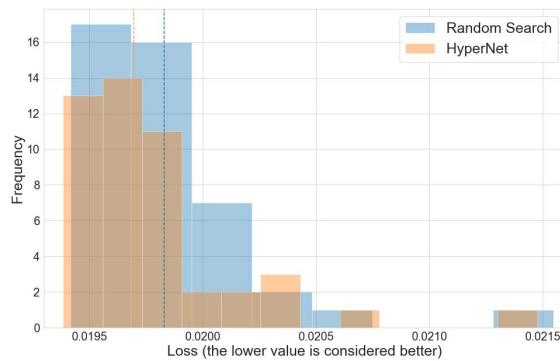


Fig. 9. Distribution of best values over 50 optimizations.

- **Evaluation Matrix:** In this study, we use MAPE for error measurement. Moreover, MSE (Loss) has been considered for the overall performance evaluation to imply the accuracy of the proposed model.

C. Single Dataset Result Analysis

In order to evaluate the performance, we first implement the proposed framework for each single benchmark dataset. Specifically, we consider 50 iterations ($T = 50$) per each run to find the best optimal values. For instance, Fig. 8 demonstrates the result of the best-found values after 50 optimizations for training *SB-R4* dataset. Furthermore, we compare the results with *Random Search* which is another well-known approach for hyperparameter tuning [36]. Fig. 9 demonstrates the distribution of 50 optimizations between two algorithms. In particular, the dashed lines, which indicate the medians of the distributions, denote that *Bayesian Search* is better than *Random Search* regarding the considered datasets.

For more detail, Tab. V depicts the summary results by applying the two approaches for all benchmark datasets in which BS, L, and TS denote the Best Score, the training step (Iteration) to find Best Score, and the Test Score, respectively. As results, using *Bayesian Search* for hyperparameter optimization outperforms *Random Search*, especially in the case of the Iteration to find the best configuration. Specifically, although there are no big differences between the loss values, however, as we mentioned above, the advantage of Bayesian optimization is able to take fewer training steps in order to

TABLE V
THE COMPARISON BETWEEN RANDOM SEARCH AND HYPERNET FOR TRAINING BENCHMARK DATASETS

Dataset	Random Search			HyperNet		
	BS	I	TS	BS	I	TS
SB-R1	0.0025	35	0.0021	0.0024	6	0.0021
SB-R4	0.0194	30	0.0181	0.0193	28	0.0178
SC-R1	0.0061	19	0.0103	0.0061	19	0.0103
SC-R4	0.0136	39	0.0139	0.0136	11	0.0139
SM-R1	0.004	42	0.0056	0.0039	37	0.0044
SM-R4	0.0137	36	0.0102	0.0137	19	0.0102
IG-R1	0.0031	9	0.0034	0.0031	1	0.0034
IG-R4	0.011	50	0.0096	0.0107	28	0.0085

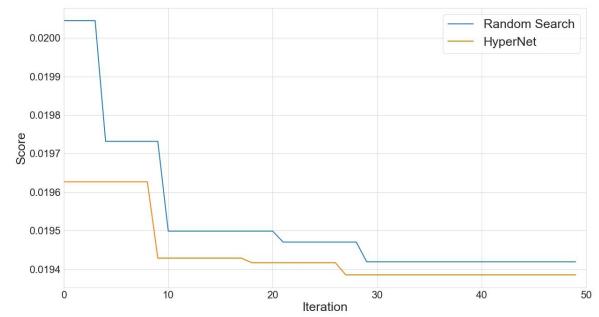
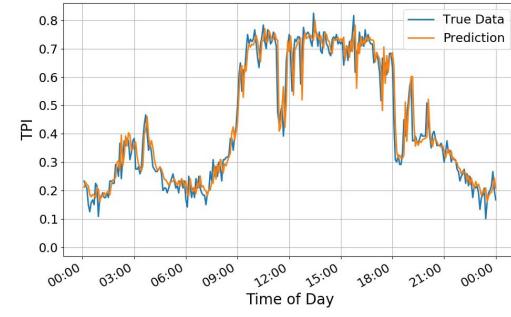
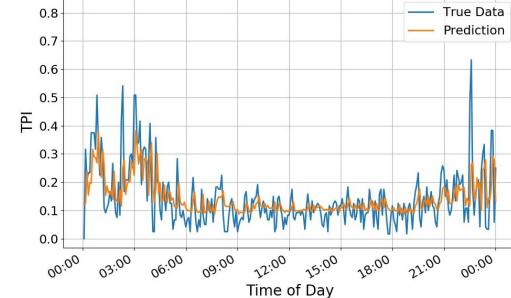


Fig. 10. Convergence graph for finding the optimum value of SB-R4 datasets.



(a) TPI Prediction of SB-R1



(b) TPI Prediction of SB-R4

Fig. 11. TPI predictions.

achieve comparable results. Particularly, this is an important concept for the further development of AutoML for learning applications in the transportation domain.

For more detail, Fig 10 demonstrates the graph convergence to find the optimum between two approaches on *SB-R4* dataset. We can see that the *Bayesian Search* has been able to converge

TABLE VI

THE COMPARISON BETWEEN SMBO AND HYPERNET FOR HYPERPARAMETER OPTIMIZATION ACROSS MULTIPLE DATASETS

Dataset	SMBO		HyperNet	
	Loss	Validation	Loss	Validation
SB-R2	0.0104	0.0095	0.0116	0.0095
SB-R3	0.0021	0.0019	0.0025	0.0019
SC-R2	0.0124	0.0138	0.016	0.0151
SC-R3	0.0048	0.0044	0.007	0.0063
SM-R2	0.019	0.0175	0.0191	0.0178
SM-R3	0.0044	0.0043	0.0051	0.0048
IG-R2	0.01	0.0111	0.0122	0.0117
IG-R3	0.0106	0.0104	0.0129	0.0111

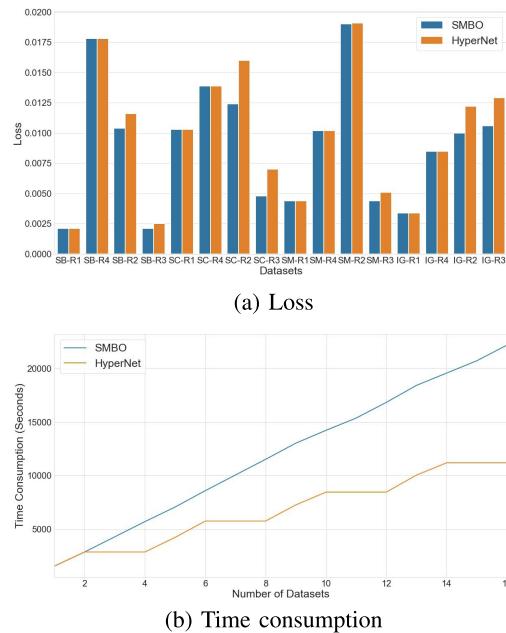


Fig. 12. Loss and time consumption.

better than the *Random Search*. Consequently, Fig. 11 demonstrates the TPI predictions of SB-R1 (congestion) and SB-R4 (non-congestion) of the SB route based on the best-found hyperparameter values by applying the proposed framework.

D. Multiple Datasets Result Analysis

The main disadvantage of applying the Bayesian approach (SMBO) is the time-consuming problem. Specifically, it is difficult to extend this approach with a large number of datasets in which we have to train each dataset independently.

Therefore, in this study, we propose the hyperparameter optimization across multiple datasets using meta-learning for highway traffic system. The main reason is that in each route of highway systems, based on traffic conditions, we are able to classify similar datasets in which we do not need to train similar datasets from scratch. In this regard, Tab. VI demonstrate the result of our implementation for the proposed framework comparing with using SMBO for training each dataset independently. Specifically, by applying the proposed framework, we are able to achieve comparable results with

the best values. However, the advantage of the proposed framework is that we are able to reduce significantly time consumption with acceptable results. In particular, Fig. 12 depicts the results in terms of accuracy and time consumption between two approaches, respectively. Note that for training multiple datasets, to simplify the problem, we assume that the input datasets are trained in sequence.

VI. CONCLUSION AND FUTURE WORK

In ITS, DL for time series prediction has exploited the effectiveness in terms of analyzing data sequence and extracting traffic flow features. In this regard, this paper focuses on implementing a DL method using LSTM-RNN models for predicting the congestion in highway systems. Specifically, the main challenge of applying DL models is that it requires the human expert to determine the appropriate model for traffic prediction. In this regard, this study proposes the *HyperNet* framework for improving the automated training process in terms of optimizing the configuration of hyperparameters across multiple datasets at the main regions of a certain highway system. Furthermore, we take the Korean highway traffic system into account as the case study of our work. The experiment indicates promising results with the highway traffic datasets that we have collected from four main routes in the Korea transportation system. For the future work of this study, we are going to extend the proposed framework for other applications of the traffic forecasting (e.g., traffic flow at urban areas or multi-channels predictions) by using another DL model (e.g., ConvLSTM or Graph Neural Networks). Specifically, based on the objective and problem definition of the DL models, the input datasets and the hyperparameters need to pre-process and determine, respectively. Moreover, the exploration in more detail for the hierarchy of the hyperparameters to train models also need to take into account. Specifically, based on our experiences, the hyperparameters should not be treated equally by applying search algorithms which can make the process more complex. There are interesting issues that we are considering regarding this research area.

REFERENCES

- [1] L. D. Baskar, J. Hellendoorn, B. De Schutter, and Z. Papp, "Traffic control and intelligent vehicle highway systems: A survey," *IET Intell. Transp. Syst.*, vol. 5, no. 1, pp. 38–52, Mar. 2011.
- [2] K.-H.-N. Bui, D. Camacho, and J. E. Jung, "Real-time traffic flow management based on inter-object communication: A case study at intersection," *Mobile Netw. Appl.*, vol. 22, no. 4, pp. 613–624, Aug. 2017.
- [3] K.-H.-N. Bui and J. J. Jung, "Internet of agents framework for connected vehicles: A case study on distributed traffic control system," *J. Parallel Distrib. Comput.*, vol. 116, pp. 89–95, Jun. 2018.
- [4] T. Seo, A. M. Bayen, T. Kusakabe, and Y. Asakura, "Traffic state estimation on highway: A comprehensive survey," *Annu. Rev. Control*, vol. 43, pp. 128–151, Mar. 2017.
- [5] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.
- [6] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, pp. 197–387, Jun. 2014.
- [7] 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.
- [8] 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.

- [9] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.
- [10] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, pp. 55:1–55:21, Mar. 2019.
- [11] Q. Yao *et al.*, "Taking human out of learning applications: A survey on automated machine learning," 2018, *arXiv:1810.13306*. [Online]. Available: <http://arxiv.org/abs/1810.13306>
- [12] F. Hutter, J. Lücke, and L. Schmidt-Thieme, "Beyond manual tuning of hyperparameters," *KI-Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, Nov. 2015.
- [13] F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., "Automated machine learning—Methods, systems, challenges," *The Springer Series on Challenges in Machine Learning*. Cham, Switzerland: Springer, 2019.
- [14] J. Snoek *et al.*, "Scalable Bayesian optimization using deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jun. 2015, pp. 2171–2180.
- [15] M. Feurer, J. T. Springenberg, and F. Hutter, "Initializing Bayesian hyperparameter optimization via meta-learning," in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, Texas, USA, Jan. 2015, pp. 1128–1135.
- [16] H. Yi, K.-H.-N. Bui, and H. Jung, "Implementing a deep learning framework for short term traffic flow prediction," in *Proc. 9th Int. Conf. Web Intell., Mining Semantics WIMS*, South Korea, Jun. 2019, pp. 7:1–7:8
- [17] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. 29th Annu. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 2962–2970.
- [18] 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, 2017.
- [19] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Oct. 2017, pp. 1–6.
- [20] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–16.
- [21] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.
- [22] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks. IJCNN. Neural Comput., New Challenges Perspect. New Millennium*, Como, Italy, Jul. 2000, pp. 189–194.
- [23] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 2960–2968.
- [24] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 2546–2554.
- [25] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, GA, USA, Jun. 2013, pp. 115–123.
- [26] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: A survey of trends and technologies," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 117–130, Jun. 2015.
- [27] J. N. van Rijn and F. Hutter, "Hyperparameter importance across datasets," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 2367–2376.
- [28] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, Aug. 2017, pp. 1126–1135.
- [29] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 826–830, Mar. 2017.
- [30] F. Pfisterer, J. N. van Rijn, P. Probst, A. Müller, and B. Bischl, "Learning multiple defaults for machine learning algorithms," 2018, *arXiv:1811.09409*. [Online]. Available: <http://arxiv.org/abs/1811.09409>
- [31] P. Probst, A. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *J. Mach. Learn. Res.*, vol. 20, pp. 53:1–53:32, Feb. 2019.
- [32] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," 2012, *arXiv:1206.5533*. [Online]. Available: <http://arxiv.org/abs/1206.5533>
- [33] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
- [34] H. Yi and K. N. Bui, "VDS data-based deep learning approach for traffic forecasting using LSTM network," in *Proc. 19th EPIA Conf. Artif. Intell. (EPIA)*, Vila Real, Portugal, Sep. 2019, pp. 547–558.
- [35] F. He, X. Yan, Y. Liu, and L. Ma, "A traffic congestion assessment method for urban road networks based on speed performance index," *Procedia Eng.*, vol. 137, pp. 425–433, Feb. 2016.
- [36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.



Hongsuk Yi received the Ph.D. degree from Sogang University, Seoul, South Korea, in 1997. He has been a Principal Researcher with the Korea Institute of Science and Technology Information (KISTI), South Korea, since 2000. His research interests are traffic congestion problem, smart cities, supercomputing, and heterogeneous computing by using deep learning techniques. Recently, his research topic focuses on developing traffic signal control solutions by using deep learning methodologies.



Khac-Hoai Nam Bui received the M.S. degree in computer science and information engineering from Aletheia University, New Taipei, Taiwan, in 2014, and the Ph.D. degree in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2018.

He has been a Post-Doctoral Researcher with the Korea Institute of Science and Technology Information (KISTI), Daejeon, South Korea, since February 2019. Before joining KISTI, he was a Postdoctoral Researcher with Chung-Ang university, Seoul, in September 2018. His research interests are the Internet of Things and ambient intelligence by using AI methodologies such as data mining, machine learning, and logical reasoning. Recently, he focuses on applying deep learning models for smart applications, such as transportation and home energy management systems.