NCAE and ELM Based Enhanced Ensemble Optimized Model for Traffic Flow Forecasting

XINXUE QI1, XIZHONG QIN1, ZHENHONG JIA1, MENGMENG LIN1, AND YUELI LIU2

1College of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China
2State Grid Wuluming Electric Power Supply Company, Ürümqi 830001, China
Corresponding author: Xizhong Qin (qmqqxz@163.com)

This work was supported by the Xinjiang Natural Science Foundation of China under Grant 2019D01C058.

ABSTRACT Accurate and timely traffic flow forecasting is the key technology to address the issue of urban traffic congestion, which is significant for intelligent transportation system. However, it is a quite challenging task to develop an efficient, robust and automatically generated forecasting model. In this paper, we propose a novel model, ensemble forecasting architecture based on non-negative constrained sparse autoencoder and extreme learning machine (NCAE-ELM-EA). This model is designed using NCAE to extract traffic characteristics layer-by-layer through a greedy unsupervised algorithm, and the prediction network formed by connecting NCAE and ELM serves as base learner of each lag pool. Then, employing adaptive enhanced integration algorithm to build an ensemble optimized forecasting model with scalable size suitable for each road segment. The model was applied to the actual data collected from I5 NB highways in Portland, USA, and J6-J7(N) freeways in the United Kingdom, with higher accuracy and lower labor costs compared with existing predictors.

INDEX TERMS Traffic flow forecasting, non-negative constrained sparse autoencoder, extreme learning machine, ensemble learning, feedback enhancement.

I. INTRODUCTION

With the acceleration of urbanization process and the sharp increase in the total number of automobiles, the increasingly serious traffic congestion problem and the resulting environmental pollution have become the most important factors affecting the lives of citizens in many large cities. How to alleviate congestion, improve road utilization, and reduce fuel consumption and emissions are the problems that Intelligent Transportation System (ITS) has been researching and solving [1]–[4]. Traffic flow forecasting is a process of estimating traffic state at a certain moment in the future based on current and past traffic conditions. Accurate and timely forecasting is not only a vital component of ITS [5], but the key to the successful deployment of current ITS. However, due to the inherent dynamic randomness and large variation of traffic flow, establishing an optimal model with high accuracy, strong robustness and auto-bulidability is a very challenging task [6], [7].

In the past few decades, many researchers have published numerous articles in this field. In general, techniques used for traffic flow forecasting can be divided into three categories: time series analysis methods, traditional machine learning and deep learning methods.

Time series analysis methods mainly take the autoregressive moving average (ARMA) [8] as basic model, followed by the autoregressive integrated moving average (ARIMA) model [9] and the seasonal ARIMA model [10], [11]. In order to quantify the uncertainty of traffic flow sequences, GARCH was used to model conditional variance of traffic flow. Subsequently, other GARCH models, including multi-variable GARCH model [12] and asymmetric GARCH model [13], [14], were successively applied to traffic prediction and made considerable progress. However, with the deepening of people's understanding of the complex nonlinearity and non-stationarity of traffic flow, these linear methods have been unable to meet the needs.

With the continuous progress of technology, the forecasting methods based on machine learning flourishes. Sun et al. [15] used the gaussian mixture model to estimate...
the joint probability distribution to predict traffic flow. Fei et al. [16] proposed a dynamic linear model (DLM) based on bayesian inference to predict travel time. Zheng and Su [17] applied the k-Nearest Neighbor (KNN) to forecast traffic volume. Support vector machine (SVM) [18], [19], which has been gradually paid attention to in this field due to its outstanding nonlinear modeling ability. However, although traditional machine learning approaches have a solid mathematical foundation which can assist us understand the mechanism of data generation, they lack the ability to process high-dimensional data [20], especially when the regularity of traffic data is time-varying. It is difficult for them to describe complex nonlinear changes, so the improvement of prediction accuracy is limited.

In recent years, deep learning models with strong multidimensional nonlinear data processing ability has attracted more and more researchers to apply it to traffic data mining. Guo et al. [7] developed a technique for short-term urban traffic flow forecasting based on convolutional neural network (CNN). In the literature [21], LSTM NN is introduced to capture nonlinear dynamic characteristics effectively. LSTM NN solves the problem of back propagation error attenuation by means of memory module, and outperforms ordinary NN. Yang et al. [22] used sparse autoencoder (SAE) to extract features from traffic flow, then obtained a robust prediction model. Generally, it is hard for single model to consistently perform well in diverse and complex scenarios, resulting in poor generalization of the model. Zhou et al. [6] introduced the $\delta$-agree strategy to train the boosting module of stacked SAE, which constituted the ensemble model of $\delta$-agree AdaBoost regression, and got superior performance in highway traffic flow forecasting.

According to the above discussions, due to the dynamic and complex nature of traffic flow, it is a promising research direction to combine advanced models that can extract hidden features from data with integration strategies. Moreover, existing studies on traffic prediction have not completely solved the problem of choosing appropriate input variables [23] and how to establish a unified framework of the best model flexibly and automatically. A novel idea is to use a set of networks with different lags to learn the characteristics of data from various aspects. In the stage of integration, it automatically selects accurate and diverse base predictors for each data set to ensemble. By means of enhanced integration, the selection of input variables is unified with the selection of base learners, which is more conducive to obtaining an excellent ensemble model under a consistent architecture.

In this paper, the NCAE-ELM-EA adaptive enhancement forecasting model is proposed, which reduces the labor cost of redesigning models for different road sections, and improves the generalization ability and accuracy of the model. The principal contributions of this work lie in the following four aspects.

1) To extract the characteristics from historical data, NCAEs with different time-lag are used as the basic module.
2) During this process, Taguchi method is applied to design the parameters of NCAEs, so as to receive the optimal parameters configuration corresponding to different lag values.
3) Features extracted by NCAEs with optimal structure are used as input to the predictor ELM, and NCAE connects ELM to form a prediction network as the base learner of each lag pool.
4) Using the enhanced integration algorithm, the base learners are selectively integrated to generate the optimal ensemble prediction model.

The rest of this paper is organized as follows, Section II introduces the NCAE network, Taguchi method, and adaptive enhanced integration algorithm. Section III presents the NCAE-ELM-EA model architecture. Section IV describes the design factors of the experiment, analyzes the experimental results and the performance of the model. Section V conclusion.

II. METHODOLOGIES

The best ensemble forecasting model that can be automatically generated is designed, trained, and integrated by Taguchi method, NCAE network, ELM, and adaptive enhancement algorithm. The schematic flow of the model is shown in Fig. 1.

A. NON-NEGATIVE CONSTRAINT AUTOENCODER (NCAE)

A sparse autoencoder (SAE) [24], [25] is essentially a NN network with the number of input nodes equals to the number of output nodes. By reconstructing the input in the output layer, the codes of input data, that is, the features, are got in the hidden layer. The cost function $J_{SAE}$ includes the average reconstruction error between all inputs and outputs, weight decay term and sparse penalty term.

$$J_{SAE} = \frac{1}{2m} \sum_{i=1}^{m} \| \hat{x}_i - x_i \|^2 + \lambda \cdot \Omega_{weights} + \beta \cdot \Omega_{sparsity}$$

where, $x$ is the input vector, $\hat{x}$ is the reconstructed vector in the output layer. $\lambda$ controls weight decay term, $\beta$ controls sparse penalty term. Weight decay term imposes nonnegativity constraint on the connecting weights.

$$\Omega_{weights} = \frac{1}{2} \sum_{l=1}^{2} g(w)$$

$$g(w_{ij}) = \begin{cases} w_{ij}^2, & w_{ij} < 0 \\ 0, & w_{ij} \geq 0 \end{cases}$$

The sparse penalty term is based on the concept of KL divergence,

$$\Omega_{sparsity} = \sum_{j=1}^{n_h} KL(\rho \| \hat{p}_j)$$
Sparse penalty term is added to ensure that useful data representation can be extracted in the hidden layer when the hidden units are greater than or equal to the input units. KL divergence is to force the neurons to be inactive in most of the time, so that the hidden layer only responds to a small number of samples in the input data, thereby increasing the sparsity of the hidden layer coding and obtaining a simple and effective representation of the input data.

The average activation $\tilde{\rho}_j$ of hidden neuron $j$ was defined as,

$$
\tilde{\rho}_j = \frac{1}{n} \sum_{i=1}^{n} [a_i(x_i) + b_j]
$$

(5)

$a_j, b_j$ are the activation and bias of the hidden neuron $j$, respectively.

KL divergence measures the difference between $\tilde{\rho}_j$ and $\rho$, and ensures sparsity by forcing $\tilde{\rho}_j = \rho$. 

$$
KL(\rho || \tilde{\rho}_j) = \rho \log(\rho / \tilde{\rho}_j) + (1 - \rho) \log((1 - \rho)/(1 - \tilde{\rho}_j))
$$

(6)

The training process of NCAE aims at minimizing the cost function $J_{SAE}$ to optimize weights and bias by using the back propagation algorithm. After a certain period of iterative training, the latent patterns hidden in the high-dimensional training data are fully extracted in the hidden layer.

### B. EXTREME LEARNING MACHINE

For the common feedforward neural network(FNN), the most frequently used training method is to adjust the weights and bias through BP algorithm based on gradient descent, so as to minimize the cost function in successive iteration to approach the ideal output. However, gradient-based methods are prone to fall into local minima, excessive training will lead to poor generalization, and when the number of hidden nodes is large, training will be time-consuming and convergence is slow. In order to solve these issues, Huang et al. [26] proposed Extreme Learning Machine (ELM) in 2005. In this method, the input weights and the hidden layer bias are given randomly, and the output weights are calculated according to the generalized inverse operation of the hidden layer output matrix.

#### C. TAGUCHI METHOD AND PARAMETER SETTINGS OF NCAE

In general, time series forecasting is based on past observations of the series to capture potential features, so an appropriate number of past values, termed lag, is the most critical part of the prediction [27]. If the lag is small, important information may be missed. On the other hand, if the lag is large, useless inputs may introduce noise. Both of these conditions are not conducive to better prediction [28], [29]. Furthermore, the optimal lag values for different data sets are generally different. For the same data set, the potential data patterns learned by the networks with different lags are also various. It is obviously unreasonable to consider only result corresponding to the optimal lag and ignore the contribution of other lags to the final prediction. Therefore, we need to adaptively select a set of appropriate lags according to various data sets, and then integrate the characteristics learned by different networks to obtain an optimal result.

In addition, the number of hidden nodes is also closely correlated with the number of input variables, especially autoencoder. It affects the dimension of the mapped features space directly, which in turn affects whether the extraction of the input data is correct, sufficient and or not. When NCAE is
trained, the effect of sparsity penalty term, as well as the size of the sparse degree and the weight constraint are related to the features extraction of NCAE. Therefore, how to set these design parameters reasonably and effectively, and analyze their correlation will determine the performance of the model.

In this paper, Taguchi method is used to select the levels of NCAE’s parameters. Taguchi method [30] is actually a series of orthogonal experiments, in which the parameters are configured by orthogonal array. Through these representative experiments, the effect of each factor’s levels is analyzed to determine the optimal design parameters combination. This method not only handles the complex adjustment of parameters and large workload of trial-and-error method, but also separates the effect of each design factor through the orthogonal combination of parameters, and understands the contribution of each factor to the model more effectively. For instance, there are five design factors contained five levels in each lag pool. Using trial-and-error method requires 3125 \((5^5)\) experiments. When an orthogonal array \(L_{25}\) \((5^5)\) is applied, only 25 experiments are needed to evaluate the influence of each factor to determine the optimum condition. Compared the number of trials required by the Taguchi method with trial-and-error method, 3100 trials \((3125-25)\) can be saved. There were 10 lag pools of the NCAE-ELM-EA, which saved a total of 31,000 \((10 \times 3100)\) experiments. This demonstrates the effectiveness of Taguchi method in the designed of model.

**D. ADAPTIVE ENHANCED ENSEMBLE**

Ensemble learning has been proved to be an effective method to improve accuracy or decompose complex problems into easier subproblems [31]. Ensemble learning [32], [33] is a general learning strategy, rather than a specific learning algorithm, which integrates the results of multiple complementary basic models by certain rules to obtain better prediction than a single model. In other ensemble models with regard to traffic flow forecasting, one method is to select the best parameters through different layers of the ensemble model. Once the optimized model is determined, other poor performance models are discarded [27]. The other method is to artificially determine a set of appropriate lags through abundant experiments to train and obtain base learners, and integrate the predictions of these models [34]. Both these methods have their own flaws. On the one hand, although the abandoned sub-optimal models have poor performance compared with the best one, they still have learned the underlying data features from different levels and provided some useful information about the future traffic condition. It is unreasonable to discard them directly. On the other hand, determining a group of base learners according to experiences is illogical, which cannot fully capture patterns of training samples and result in the model lacks flexibility, thus affecting the further improvement of prediction performance.

In this paper, the model not only makes prediction based on the features extracted by NCAEs, but also focuses on the accuracy of the base learners. At the stage of integration, an adaptive enhancement strategy is adopted to verify whether the addition of each lag pool’s network has made a contribution, so that the accuracy of the ensemble output is improved, as well as the flexibility to deal with different data sets is increased.

Therefore, a series of lags are set to constitute basic lag pools for integration. In each lag pool, Taguchi method is used to design a set of parameters’ levels to configure different base learners. Then using bagging method to further ensure the diversity among base learners in each lag pool. Finally, the adaptive enhanced ensemble algorithm (Algorithm 2) is applied to generate the ensemble prediction model.

**Algorithm 2 Adaptive Enhanced Ensemble Algorithm**

**Input:** The optimal base learner of each lag pool, its MAPE on the valid sets.

**Step1:** Arrange all base learners in ascending order according to their MAPE

**Step2:** Integrate the top two base learners to get the first ensemble model.

**Step3:** Each base learner in the sequence is added to the ensemble model one by one. Finally, according to the prediction performance of the model after each integration operation, the best ensemble model is determined.

**Output:** Optimal ensemble prediction model.

---

**III. NCAE-ELM-EA FORECASTING MODEL**

This paper proposes an enhanced ensemble architecture prediction model (NCAE-ELM-EA) based on stacked NCAE and ELM. The major steps of model can be described in Fig. 2.

The model is mainly composed of three parts: (1) preprocessing module, sample filter was used to process the missing values and outliers of the original data sequences, then the data series was divided into samples belonging to different lags. (2) Training module, trains NCAEs and its subsequent connected regression layer, analyzes the relationship between factors’ levels and the prediction performance, thus determines the optimal NCAE structure, in each lag pool. The selected NCAE with optimal structure connects to the ELM to constitute a network, called NCAE-ELM prediction network, as the basic learner for each lag pool. (3) The adaptive enhancement integration module, which uses the enhanced integration algorithm to selectively integrate the best base learners of each lag pool, forms an ensemble prediction model with scalable model size.

---

**IV. EXPERIMENTAL DESIGN AND PERFORMANCE EVALUATION**

**A. DATA DESCRIPTION**

The traffic data used in this paper are collected from the I5 NB freeway in Portland, USA [34] and the J6-J7(N) road segment of M6 highway in Britain in the U.K [22], [23]. (I5 NB road: http://portal.its.pdx.edu/. J6-J7(N) road: ia
http://data.gov.uk/data/search). We use the 15 minutes aggregation data (veh/s/15 minutes), the collected traffic patterns and the sampling time of data are more suitable for short-term traffic flow forecasting.

In experiments, we chose data from I5 NB on the working days between September 11, 2015 and October 5, 2015 (17 working days) as data set. The first 13 weekdays are grouped as the training set, the last 4 weekdays is treated as the test set. We regard data from J6-J7 (N) on the March 7, 2016 and March 31, 2016 (19 working days) as data set. Then, the beginning 15 weekdays are seen as the training set, the remaining 4 days is used as the test set. We discard the data belonging to weekends because the traffic states on weekends vary differently weekdays. Since we adopt the 10-fold cross-validation in the training process, the validation set is composed of 10% samples selected from the training set randomly.

### B. PARAMETER SETTING

There are 6 design parameters of NCAE network, and 5 parameters of each lag pool are designed using Taguchi method which considers the design factors and their corresponding levels are shown in Table 1.

#### TABLE 1. Design factors for the NCAE model.

| Design factors | levels |
|----------------|--------|
| $N_1$          | 10     | 20     | 30     | 40     | 50     |
| $N_2$          | 5      | 10     | 20     | 30     | 40     |
| $\lambda$      | 0.03   | 0.3    | 2      | 5      | 10     |
| $\beta$        | 0.5    | 1      | 3      | 5      | 10     |
| $\rho$         | 0.01   | 0.03   | 0.05   | 0.1    | 0.3    |

1) Design Factor lag: The appropriate lag, termed the number of input nodes of NCAE. We set a group of lag values $lag = \{1, 2, 3, 4, 8, 12, 16, 24, 48, 96\}$.
2) Design Factor i: $N_1$, the number of hidden nodes in the first-layer NCAE, which is determined according to the selection rule of hidden nodes of MLP, the minimum is $\log_2 N$, $2 \times \log_2 N$, $\ldots$, the maximum is 50, $N$ is the number of training nodes [35, 36], as shown in Table 1.
3) Design Factor ii: $N_2$, the number of hidden nodes of the second-layer NCAE.
4) Design Factor iii: $\lambda$, the coefficient of the $L_2$ weight regularizer in the cost function.
### TABLE 2. Lag = 12 pool, orthogonal array \(L_{25}(5^5)\) and experimental results.

| Main Trials | Design factor/Level | J6-J7(N) road section | IS NB road section |
|-------------|---------------------|------------------------|-------------------|
| i | ii | iii | iv | v | MAPE | VAPE | MAPE | VAPE | MAPE | VAPE |
| 1 | 1 | 1 | 1 | 1 | 1 | 9.96 | 2.46 | 13.06 | 3.90 | 26.56 | 57.82 | 32.28 | 69.73 |
| 2 | 1 | 2 | 2 | 2 | 2 | 9.81 | 2.61 | 10.64 | 2.98 | 26.25 | 55.18 | 28.01 | 55.49 |
| 3 | 1 | 3 | 3 | 3 | 3 | 13.88 | 4.55 | 10.26 | 2.65 | 40.24 | 84.59 | 26.63 | 61.94 |
| 4 | 1 | 4 | 4 | 4 | 4 | 12.83 | 4.01 | 8.64 | 1.92 | 34.41 | 72.31 | 23.20 | 56.22 |
| 5 | 1 | 5 | 5 | 5 | 5 | 10.18 | 2.62 | \textbf{8.13} | \textbf{1.87} | 29.02 | 62.35 | 22.37 | 52.75 |
| 6 | 2 | 1 | 2 | 3 | 4 | 8.49 | 2.07 | 15.11 | 5.06 | 24.06 | 55.88 | 38.12 | 84.19 |
| 7 | 2 | 2 | 3 | 4 | 5 | 8.42 | 2.00 | 9.08 | 2.15 | 23.94 | 58.07 | 24.82 | 60.12 |
| 8 | 2 | 3 | 4 | 5 | 1 | 10.69 | 2.68 | 11.63 | 3.09 | 32.94 | 79.24 | 35.38 | 83.43 |
| 9 | 2 | 4 | 5 | 1 | 2 | 11.13 | 2.94 | 10.53 | 2.62 | 33.33 | 75.72 | 32.10 | 72.46 |
| 10 | 2 | 5 | 1 | 2 | 3 | \textbf{8.31} | \textbf{1.96} | 9.47 | 2.28 | 24.05 | 54.48 | 25.11 | 51.33 |
| 11 | 3 | 1 | 3 | 5 | 2 | 8.53 | 1.98 | 17.06 | 6.50 | 23.24 | 53.03 | 43.92 | 94.58 |
| 12 | 3 | 2 | 4 | 1 | 3 | 8.63 | 1.98 | 21.60 | 9.89 | 24.45 | 58.23 | 55.52 | 142.37 |
| 13 | 3 | 3 | 5 | 2 | 4 | 8.70 | 2.08 | 11.99 | 3.35 | 23.77 | 56.95 | 31.04 | 70.52 |
| 14 | 3 | 4 | 1 | 3 | 5 | 8.45 | 2.01 | 8.52 | 1.93 | 23.75 | 55.24 | \textbf{22.29} | \textbf{53.30} |
| 15 | 3 | 5 | 2 | 4 | 1 | 8.54 | 2.03 | 13.32 | 4.08 | 23.76 | 53.89 | 31.16 | 62.82 |
| 16 | 4 | 1 | 4 | 2 | 5 | 8.45 | 2.07 | 15.62 | 5.52 | 22.60 | 53.97 | 38.28 | 86.11 |
| 17 | 4 | 2 | 5 | 3 | 1 | 10.16 | 2.54 | 21.39 | 9.98 | 26.99 | 61.34 | 53.44 | 130.80 |
| 18 | 4 | 3 | 1 | 4 | 2 | 8.40 | 2.01 | 10.46 | 2.65 | 22.71 | 51.16 | 26.44 | 58.03 |
| 19 | 4 | 4 | 2 | 5 | 3 | 8.53 | 1.95 | 8.65 | 1.91 | 22.88 | 53.31 | 23.40 | 55.81 |
| 20 | 4 | 5 | 3 | 1 | 4 | 8.53 | 2.02 | 8.72 | 1.97 | 22.49 | 52.23 | 23.99 | 54.94 |
| 21 | 5 | 1 | 5 | 4 | 3 | 8.45 | 1.90 | 28.46 | 17.25 | 22.65 | 50.23 | 64.52 | 178.92 |
| 22 | 5 | 2 | 1 | 5 | 4 | 8.55 | 2.00 | 9.95 | 2.44 | \textbf{22.37} | \textbf{49.26} | 24.73 | 54.83 |
| 23 | 5 | 3 | 2 | 1 | 5 | 8.56 | 2.05 | 8.27 | 2.00 | 22.87 | 50.05 | 23.47 | 54.16 |
| 24 | 5 | 4 | 3 | 2 | 1 | 8.43 | 1.84 | 9.39 | 2.18 | 24.08 | 54.51 | 26.80 | 61.17 |
| 25 | 5 | 5 | 4 | 3 | 2 | 8.51 | 1.92 | 8.64 | 1.97 | 22.92 | 53.60 | 24.61 | 56.98 |

5) Design Factor iv: \(\beta\), the coefficient for the sparsity regularization term.

6) Design Factor v: Sparsity Proportion \(\rho\), desired proportion of training examples a neuron reacts to, is a parameter of the sparsity regularizer. It controls the sparsity of the output of the hidden layer.

### C. PERFORMANCE ANALYSIS

In order to evaluate the accuracy and efficiency of the stacked NCAE-ELM-EA model proposed in this paper, its performance was evaluated by two commonly used metrics, including mean absolute percentage error (MAPE), variance absolute percentage error (VAPE).

\[
MAPE = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \tag{7}
\]

\[
VAPE = \text{Var} \left( \left| \frac{y_i - \hat{y}_i}{y_i} \right| \right) \times 100\% \tag{8}
\]

where \(y_i\) is the observed traffic flow condition, \(\hat{y}_i\) is the predicted traffic flow condition, and \(m\) is the number of forecasted points. An orthogonal array \(L_{25}(5^5)\) is used for the trial design. Based on the combination of the levels of design factors, 25 main trials are conducted to obtain the optimized structure of the certain lag pool. Each main trial corresponds to each row of the adopted orthogonal array. On the two traffic data sets (J6-J7(N), IS NB), the results of 25 main trials of the base learners in the base learner pool with lag 12 are shown in Table 2.

As shown in Table 2, on the J6-J7(N) road, the 10th main trial with 12 input nodes (that is lag), 20 hidden nodes for the NCAE, setting \(3e^{-2}, 1\) and \(0.05\) are the regularizer weight, sparsity term coefficient and sparsity parameter respectively, achieves the smallest MAPE(8.31). In contrast, on the IS NB road, When an NCAE is connected to the regression layer, the 22th main trial obtains the smallest MAPE(22.37) and VAPE(49.26). When stack two layers of NCAE is connected to regression layer, the 14th main trial, with 30 hidden nodes for two NCAEs, setting \(3e^{-2}, 3\) and \(0.3\) are the regularizer weight, sparsity term coefficient and Sparsity parameter respectively, achieves the smallest MAPE(22.29).
According to above discussion, in the same lag pool, the prediction accuracy of base learners with different structures on the same data set is different, and the performance of base learners with same structure on different data sets also varies greatly. If the model can be generated adaptively according to different data sets, it can not only improve the forecasting accuracy, but also reduce the modeling workload, reduce human intervention and improve the universality of the model.

The effect of design factors can be separated since the combinations of design factors of trials are orthogonal. The effect of each design factor at each of the five levels are calculated by taking the average from Table 2 for a design factor at a given level. For example, the level 3 of the design factor i is in the 11th, 12th, 13th,14th, and 15th main trials, on the J6-J7 (N) segment, when the network is a layer of NCAE and regression layer, the average of this level is 8.57% (MAPE) and 2.02%(VAPE). Lower value of MAPE and VAPE indicate better performance.

On I5 NB road, when lag is 12, for two kind of network(first-layer NCAE or second-layer NCAE connect to regression layer), the effects of design factors at five levels are shown in Fig. 3.

Based on above analysis, which level of parameters can get the best effect will be known, we can obtain the optimized parameters combination of base learners in each lag pool. The result of optimal parameters’ level on two data sets, is shown in table 3 and 4.

It can be seen from Table 3 and 4 that the parameters of the selected optimal base learners vary greatly on different data sets. As a result, the NCAE with different structures in base learners can extract diverse characteristics of input data and mine potential traits of traffic flow sequence. On the J6-J7(N) and I5 NB road section data sets, the base learners that constitute the best ensemble prediction model is shown in the table 5.
TABLE 3. J6-J7 road section, The optimized parameters’ level of the base learners of each lag pool.

| Network | the network of first-layer NCAE + ReLayer | the network of second-layer NCAE + ReLayer |
|---------|------------------------------------------|------------------------------------------|
| Design factor | Factor i | Factor ii | Factor iii | Factor iv | Factor v | Factor i | Factor ii | Factor iii | Factor iv | Factor v |
| Lag = 1 | 5 | - | 5 | 5 | 5 | 5 | 4 | 5 | 4 | 5 |
| Lag = 2 | 5 | - | 4 | 4 | 4 | 5 | 3 | 5 | 3 | 5 |
| Lag = 3 | 4 | - | 4 | 3 | 4 | 4 | 4 | 3 | 5 | 5 |
| Lag = 4 | 4 | - | 5 | 5 | 2 | 1 | 4 | 3 | 3 | 4 |
| Lag = 8 | 5 | - | 1 | 4 | 5 | 1 | 4 | 3 | 5 | 5 |
| Lag = 12 | 5 | - | 1 | 2 | 5 | 1 | 4 | 3 | 5 | 5 |
| Lag = 16 | 5 | - | 2 | 2 | 5 | 1 | 5 | 3 | 5 | 4 |
| Lag = 24 | 5 | - | 2 | 2 | 5 | 1 | 5 | 3 | 2 | 4 |
| Lag = 48 | 3 | - | 2 | 4 | 5 | 1 | 5 | 3 | 2 | 5 |
| Lag = 96 | 5 | - | 1 | 1 | 5 | 1 | 5 | 3 | 2 | 5 |

TABLE 4. I5 NB road section, The optimized parameters’ level of the base learners of each lag pool.

| Network | the network of first-layer NCAE + ReLayer | the network of second-layer NCAE + ReLayer |
|---------|------------------------------------------|------------------------------------------|
| Design factor | Factor i | Factor ii | Factor iii | Factor iv | Factor v | Factor i | Factor ii | Factor iii | Factor iv | Factor v |
| Lag = 1 | 3 | - | 4 | 5 | 4 | 5 | 4 | 5 | 5 | 5 |
| Lag = 2 | 2 | - | 4 | 1 | 5 | 5 | 4 | 4 | 3 | 5 |
| Lag = 3 | 2 | - | 1 | 1 | 5 | 4 | 4 | 4 | 3 | 5 |
| Lag = 4 | 2 | - | 2 | 3 | 5 | 4 | 4 | 3 | 3 | 4 |
| Lag = 8 | 5 | - | 1 | 2 | 5 | 1 | 4 | 3 | 5 | 5 |
| Lag = 12 | 5 | - | 1 | 2 | 5 | 1 | 5 | 3 | 2 | 5 |
| Lag = 16 | 5 | - | 1 | 2 | 5 | 1 | 5 | 3 | 5 | 5 |
| Lag = 24 | 5 | - | 1 | 2 | 5 | 1 | 5 | 3 | 5 | 5 |
| Lag = 48 | 3 | - | 2 | 2 | 5 | 1 | 5 | 3 | 2 | 5 |
| Lag = 96 | 5 | - | 1 | 2 | 5 | 1 | 5 | 3 | 2 | 5 |

TABLE 5. Base learners who constitute NCAE-ELM-EA.

| Best ensemble forecasting model | base learners that make up the ensemble model (in ascending MAPE order) |
|--------------------------------|---------------------------------------------------------------------|
| J6-J7(N) road data set | 1lag = 12, base learner B, 7.44%; 2lag = 16, base learner A, 7.98%; 3lag = 12, base learner A, 8.09%; 4lag = 8, base learner A, 8.12%; 5lag = 16, base learner B, 8.12%; 6lag = 24, base learner A, 8.21%; 7lag = 3, base learner A, 8.63%; |
| I5 NB road data set | 1lag = 24, base learner B, 14.16%; 2lag = 16, base learner B, 15.44%; 3lag = 8, base learner B, 16.12%; 4lag = 8, base learner A, 16.20%; 5lag = 24, base learner A, 16.33%; 6lag = 12, base learner B, 16.53%; 7lag = 12, base learner A, 16.81%; 8lag = 4, base learner B, 16.86%; 9lag = 16, base learner A, 17.00%; 10lag = 96, base learner B, 17.32%; 11lag = 4, base learner A, 17.42%; 12lag = 3, base learner B, 17.51%; |

Each subgraph in figure 4 represents the features obtained by the NCAE of a certain base learner, the row represents the number of hidden nodes of NCAE, namely the dimension of features, and each column denotes a sample. From the gray-scale map, we can find that the extracted features show obvious daily periodicity. When taking different lag values, the characteristics obtained have similar rules, but there are differences in details.

In Figure 5, we choose base learners whose performance is the first and third on the I5 NB road data set, and unfold their features on a daily basis. It can be clearly seen from (C) and (D) that different NCAE structures acquire various...
features on the same test set. They are all stacked two-layer NCAE networks. When the lag value is smaller, the granularity of the features’ grey scale image is larger during peak hours, while when the lag value is larger in the morning and evening non-peak hours, the granularity is larger. Networks with different lags capture the short-term and long-term trends of traffic flow, and diverse features will be the basis for improving the prediction accuracy of the integrated model.

This paper adopts an adaptive enhanced integration strategy to establish a unified model NCAE-ELM-EA that can adaptively determine the best network structure based on the data set. When the ensemble model selects the individuals to be integrated, it arranges the prepared base learners according to their performance, and then integrates them into the model one by one. On the J6-J7(N) and I5 NB data set, the prediction results of the best ensemble prediction model on the test set are shown in Figure 6.

On the different data sets, although the prediction accuracy of base learners varies, even some of them have poor performance, the model obtained by integrating these base learners with adaptive enhancement strategy has better prediction performance. NCAE-ELM-EA synthesizes the advantages of networks with different structure, that is, by taking various lag values, the corresponding optimal network is used to fully extract data features from multiple time correlation angles, and the data rules are grasped, so as to improve the prediction accuracy. For different data sets, the ensemble model selects different integration individuals, but the selected networks are all based on the performance of networks in the training process. This approach makes the model more flexible and robust, which is more obvious on I5 NB data set with frequent and drastic data fluctuations. Moreover, the size of the ensemble model is adaptive and scalable, because the accuracy of each base learner will slightly fluctuate in different training processes, resulting in differences in performance ranking and contribution to the final prediction result, but the final prediction result with high accuracy tend to be stable.

By comparing the five prediction models of the proposed NCAE-ELM-EA, MLP, E-ELM [28], LSTM [22] and EnLSSVR [31], the accuracy and effectiveness of networks based on NCAE with the best structure and the

---

**FIGURE 4.** Greyscale image of features. (A): Features extracted by NCAE of the base learners (the hidden layer output of NCAE), on the training set of J6-J7(N) road section. (B): Features on the test set of J6-J7(N) road section. (C): Features on the training set of I5 NB road section. (D): Features on the test set of I5 NB road section.
enhanced integration method in traffic flow prediction are proved. The parameter setting of the competition models: MLP has two hidden layers, whose the number of input and hidden nodes are selected from the parameter settings in this paper. Some of the same parameters in the E-ELM model follow the setting of literature [25], and the remaining parameters are determined according to the experimental data set training in this paper. The time delay of LSTM is also determined according to the lag set adaptive in this paper for different data sets. EnLSSVR model is an
ensemble model, and its parameters are set according to literature [34].

The predicted results of all methods on test sets of each section are shown in Figure 7 and Figure 8. On all data sets, the evaluation tests of the NCAE-ELM-EA model and the comparative prediction model were repeated 10 times, and the performance results were shown in Table 6.

From these results, we can find that, on the J6-J7(N) road section, the performance of the NCAE-ELM-EA model proposed in this paper is very close to that of E-ELM and MLP, with high accuracy and small error fluctuation. The MAPE

| Dataset | index | NCAE-ELM-EA | MLP | LSTM | E-ELM | EnLSSVR |
|---------|-------|-------------|-----|------|-------|---------|
| J6-J7(N) section | MAPE | 7.58% | 7.69% | 9.88% | 7.97% | 8.44% |
| IS NB section | VAPE | 0.99 | 1.07 | 1.77 | 1.14 | 1.22 |
| | MAPE | 13.56% | 15.56% | 18.91% | 17.58% | 16.02% |
| | VAPE | 4.86 | 7.64 | 6.89 | 7.25 | 6.99 |
of NCAE-ELM-EA was only 7.58%, which was 2.3% and 0.86% higher than LSTM and EnLSSVR respectively. On the I5 NB road section, compared with the comparison model, the MAPE of this model decreased by 2%, 5.35%, 3.82% and 2.46% respectively, and the error fluctuation (VAPE) was also the smallest. On the I5 NB road section, the performance of E-ELM model who has excellent performance on the J6-J7(N) road section, is significantly worse than that of the NCAE-ELM-EA, which proves that NCAE-ELM-EA not only has high prediction accuracy, but also has good robustness. When meet road data with frequent and violent fluctuations, the NCAE-ELM-EA model can better grasp the trend of data sequence and more accurately capture the characteristics of the fluctuation segment. Compared with other models, this model can adapt to different data flexibly, showing the advantages of enhanced integration.

V. CONCLUSION
In this paper, we proposed a novel model for short-term traffic flow forecasting. The method is the first time to use the non-negative constraint autoencoder as the base learner, the Taguchi method is used to design and obtain the optimized basic networks in each lag pool, and the ensemble model is constructed through the enhanced integration algorithm. The NCAE-ELM-EA model is applied to the real traffic flow scenario. The experimental results show that this model can be applied to different traffic flow data sets flexibly, and can achieve better prediction performance than other widely used models.

REFERENCES
[1] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, “Adaptive multi-kernel SVM with spatial–temporal correlation for short-term traffic flow prediction,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2001–2013, Jun. 2019.
[2] Z. Diao, D. Zhang, X. Wang, K. Xie, S. He, X. Lu, and Y. Li, “A hybrid model for short-term traffic volume prediction in massive transportation systems,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 3, pp. 935–946, Mar. 2019.
[3] A. Hamilton, B. Waterson, T. Cherrett, A. Robinson, and I. Snell, “The evolution of urban traffic control: Changing policy and technology,” Transp. Planning Technol., vol. 36, no. 1, pp. 24–43, Feb. 2013.
[4] A. Poole and A. Kotsialos, “Swarm intelligence algorithms for macroscopic traffic flow model validation with automatic assignment of fundamental diagrams,” Appl. Soft Comput., vol. 38, pp. 134–150, Jan. 2016.
[5] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golas, “Short-term traffic forecasting: Where we are and where we’re going,” Transp. Res. C, Emerg. Technol., vol. 43, pp. 3–19, Jun. 2014.

[6] T. Zhou, G. Han, and X. Xu, “A-graded AdaBoost stacked autoencoder for short-term traffic flow forecasting,” Neurocomputing, vol. 247, pp. 31–38, Apr. 2017.

[7] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, “Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[8] C. K. Moorthy and B. G. Ratcliffe, “Short-term traffic forecasting using time series methods,” Transp. Planning Technol., vol. 12, no. 1, pp. 45–56, Jul. 1988.

[9] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining kohonen maps with arima time series models to forecast traffic flow,” Transp. Res. C, Emerg. Technol., vol. 4, no. 5, pp. 307–318, Oct. 1996.

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

[11] M. Lippi, M. Bertini, and P. Frasconi, “Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 871–882, Jun. 2013.

[12] J. Xia, Q. Nie, W. Huang, and Z. Qian, “Reliable short-term traffic flow forecasting for urban roads: Multivariate generalized autoregressive conditional heteroscedasticity approach,” Transp. Res. Rec., vol. 2343, no. 1, pp. 77–85, Dec. 2013.

[13] Y. Zhang, R. Sun, A. Haghani, and X. Zeng, “Univariate volatility-based models for improving quality of travel time reliability forecasting,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2365, no. 1, pp. 73–81, Jan. 2013.

[14] Y. Zhang, A. Haghani, and R. Sun, “Stochastic volatility modeling approach that accounts for uncertainties in travel time reliability forecasting,” Transp. Res. Rec., J. Transp. Res. Board, vol. 2442, no. 1, pp. 62–70, Jan. 2014.

[15] S. Sun, C. Zhang, and G. Yu, “A Bayesian network approach to traffic flow forecasting,” IEEE Trans. Intell. Transp. Syst., vol. 7, no. 1, pp. 124–132, Mar. 2006.

[16] X. Fei, C.-C. Lu, and K. Liu, “A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction,” Transp. Res. C, Emerg. Technol., vol. 19, no. 6, pp. 1306–1318, Dec. 2011.

[17] Z. Zheng and D. Su, “Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm,” Transp. Res. C, Emerg. Technol., vol. 43, no. 1, pp. 143–157, Jun. 2014.

[18] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” Statist. Comput., vol. 14, no. 3, pp. 199–222, Aug. 2004.

[19] Y. Dong, Z. Zhang, and W.-C. Hong, “A hybrid seasonal mechanism with a chaotic cuckoo search algorithm with a support vector regression model for electric load forecasting,” Energies, vol. 11, no. 4, p. 1099, Apr. 2018.

[20] M. G. Karlaftis and E. I. Vlahogianni, “Statistical methods versus neural networks in transportation research: Differences, similarities and some insights,” Transp. Res. C, Emerg. Technol., vol. 19, no. 3, pp. 387–399, Jun. 2011.

[21] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” Transp. Res. C, Emerg. Technol., vol. 54, pp. 187–197, May 2015.

[22] H.-F. Yang, T. S. Dillon, and Y.-P. Chen, “Optimized structure of the traffic flow forecasting model with a deep learning approach,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2371–2381, Oct. 2017.

[23] H.-F. Yang, T. S. Dillon, E. Chang, and A. Y.-P. P. Chen, “Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting,” IEEE Trans. Ind. Informat., vol. 15, no. 1, pp. 23–34, Jan. 2019.

[24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1096–1103.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J. Mach. Learn. Res., vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[26] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: A new learning scheme of feedforward neural networks,” in Proc. IEEE Int. Joint Conf. Neural Netw., Jul. 2005, pp. 985–990.
MENGMENG LIN is currently pursuing the master’s degree in information and communication engineering with the College of Information Science and Engineering, Xinjiang University, China. Her current research interests include time series analysis, machine learning, and pattern recognition.

YUELI LIU received the M.S. degree from the College of Information Science and Engineering, Xinjiang University, China. She is currently with State Grid Wulumuqi Electric Power Supply Company, Ürümqi. Her research interest includes machine learning.