IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 5, 2018, accepted December 22, 2018, date of publication December 25, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2889814

A Cohesion-Based Heuristic Feature Selection for Short-Term Traffic Forecasting

LISHAN LIU[1], NING JIA[1], LEI LIN[2], AND ZHENGBING HE[3]
[1]College of Management and Economics, Tianjin University, Tianjin 300072, China
[2]Goergen Institute for Data Science, University of Rochester, Rochester, NY 14620, USA
[3]Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, Beijing 100024, China

Corresponding author: Zhengbing He (he.zb@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71871010, Grant 71871011, and Grant 71571013 and in part by the Beijing Municipal Natural Science Foundation under Grant 9174027.

ABSTRACT An input vector composed of various features plays an important role in short-term traffic forecasting. However, there is limited research on the optimal feature selection of an input vector for a certain forecasting task. To fill the gap, this paper proposes a cohesion-based heuristic feature selection method by analyzing the nature of the forecasting methods. This method is able to determine which features should be contained in an input vector to make a forecasting algorithm perform better. The proposed method is demonstrated in two experiments based on the empirical traffic flow data. The results show that the method is able to improve the performances of the short-term traffic forecasting algorithms. It is then suggested to consider the proposed method as a preprocessing procedure in practical forecasting applications.

INDEX TERMS Traffic flow, short-term forecasting, optimal feature selection, input vector.

I. INTRODUCTION

Short-term traffic forecasting is a basic and important part of intelligent transportation systems. Traffic managers make use of it to drive traffic control and guidance strategies, while travelers can benefit from it in making route choices.

By definition, short-term traffic forecasting is the process to predict key traffic parameters such as speed, flow, occupancy or travel time, with a forecasting horizon typically ranging from 5 to 30 minutes at specific locations. The general form of the traffic forecasting problem can be expressed as:

$$y(t+1) = f(x(t)) \tag{1}$$

where $x(t)$ denotes the traffic features available at time t. Normally, it includes data elements that are assumed to influence the target traffic feature. If only time series is considered, $x(t)$ usually includes $\{y(t), y(t-1), \ldots, y(t-n)\}$, where n is a measure of the time history. In some spatial temporal forecasting models, $x(t)$ includes traffic data not only from the link of interest, but also from related links that are considered to affect traffic evolution of the target link. In the field of short-term traffic forecasting, $x(t)$ is usually called a ''input vector''. The function $f(\cdot)$ in Equation 1 defines the relationship between input vectors and output results depicted by using a forecasting model. This relationship can be depicted in the form of mathematical equations, or so-called ''black-box'' nonparametric mapping models.

In the field of traffic forecasting, the forecasting algorithms have been widely studied [1]–[6], which is a key that determines the forecasting results. Beyond that, the input vector is another key, i.e., inappropriately selected input vectors easily result in non-ideal forecasting results. Unfortunately, there exist few works focusing on ''how to determine the input vector'', as it will be shown in the following section of literature review.

To facilitate the research regarding feature selection, this paper proposes a cohesion-based feature selection method for short-term traffic flow forecasting. Experiments based on real-world data are conducted, and the results show that the proposed method can reduce the forecasting error of four widely used forecasting methods. The proposed method can be treated as a data preprocessing procedure in various traffic flow forecasting applications.

The rest of this paper is organized as follows. Section II first gives a brief literature review of the short-term traffic forecasting and the research regarding feature selection. Section III proposes the cohesion-based feature selection method. Section IV conducts two experiments with real-world traffic data collected at two different locations, to demonstrate the effectiveness of the proposed method, Section V concludes the paper and offers future directions.

II. A BRIEF LITERATURE REVIEW

In general, there are two classes of traffic forecasting methods, i.e., parametric and nonparametric methods. The parametric method assumes that there is an explicit mathematical form of $f(\cdot)$ in Equation 1 that is characterized by a set of parameters. Historical data is used to determine the parameters that can minimize the forecasting error. Then, the enclosed formulas can be used in the real-time forecasting.

Due to the space limitation, we only briefly review some of the most important works, and readers can refer to the latest literature review given in [1] and [2] for more detail of various short-term forecasting research.

A. PARAMETRIC FORECASTING METHODS

Among the existing parametric methods, the most popular one is the time series-related method, which treats traffic flow as an ordinary time series. One of the most popular models is Auto Regressive Integrated Moving Average (ARIMA) method. In the early time of traffic forecasting, ARIMA was widely used [7]–[10]. However, the ARIMA method requires significant expertise to calibrate and maintain, and lacks self-learning ability. These problems hinder the wide use of the ARIMA method in real-world traffic forecasting [11]. In the recent decade, various ARIMA-based methods are developed. For example, the multi-variable ARIMA method [12] incorporates multi-sites data into the traditional model. Reference [12] states that their method improves the forecasting accuracy, and can achieve large-scale forecasting with low computational burden. Besides the ARIMA-based models, it is worth noting that the state-space and linear regression methods are also popular parametric methods [13], [14].

B. NONPARAMETRIC FORECASTING METHODS

Instead of finding the explicit mathematical form, nonparametric methods are driven by data and allow data to speak for itself [15]. One of the most popular nonparametric methods is nonparametric regression (NPR). The NPR method retains all historical observation and searches for the most similar case of the current state, and then makes forecasting. Reference [16] investigated the practical use of the NPR method and discussed the potential problems of the method in practice. Reference [17] studied the multi-variant NPR forecasting, as well as the influence of neighbor size and the transferability of database, which are valuable topics for the practical use of the NPR model. Reference [18] made three improvements for faster calculation and higher accuracy, including the data organization and the search mechanism. References [19] and [20] incorporated additional information into the NRP forecasting, such as historical and real-time traffic states, and stated that these incorporations help to reduce forecasting errors. The shortcomings of the NPR method (or even of all nonparametric methods) are obvious, i.e., the requirements of a large amount of data, a large storage space, and heavy computational burden.

C. FEATURE SELECTION OF AN INPUT VECTOR

As above-mentioned, an input vector is critical for the goodness of the forecasting result. For example, [17] incorporated speed and occupancy into the input vector, and the forecasting accuracy was thus promoted. The flow on surrounding roads was added into the input vector, due to the fact that the upstream flow may have influence on the downstream flow to be predicted [12], [21], [22]. Reference [23] also showed that feature selection can improve the forecasting accuracy. The p-test score was used to conduct the feature ranking and wrapper-like scheme, which aims to select the optimal number of features for traffic congestion prediction. A fuzzy entropy feature selector was applied to determine redundant factors and rank factor importance when modeling the incident duration [24]. Moreover, [25] utilized correlation-based method to choose the most relevant factors as the input of the SVM model for the prediction of the zonal crash frequency. Reference [26] employed the Recursive Feature Elimination to screen out the important factors in traffic accidents prediction. A Lasso method based Granger causality model was adopted in [27] to retrieve spatiotemporal characteristics with a form of causal relationship, based on which a multi-variable linear regression model was built to predict traffic flow prediction in a freeway. Although the methods include related features in the input vector, few of them proposes an optimized way to select input vectors that can make the forecasting results as good as possible.

III. A COHESION-BASED FEATURE SELECTION METHOD

Let L_0 be the study site in a road network, and t and $t+\delta$ be current time and the time of prediction, respectively. Denoted by $V_0(t+\delta)$ the traffic volume at time $t+\delta$ at L_0. Then, the input vector S_t is selected (commonly based on expertise), and it is composed of features assumed to influence $V_0(t+\delta)$. We define $P_t\{S_t, V_0(t+\delta)\}$ as a pattern, which consists of input vector S_t and forecasting result $V_0(t+\delta)$. Then, a pattern database P can be constructed, including the patterns for all time slices. An input vector selection strategy that impacts on the pattern database is proposed as follows.

Denote by $d(P_1, P_2)$ the distance between two patterns. For a given pattern $P_t \in P$, the nearest n similar patterns could be found, which are denoted by $P_{t_1} \sim P_{t_n}$ and named as the similar pattern set (SPS). In the field of pattern recognition, it is usually assumed that the mapping function is continuous, meaning that the more similar the input vectors are, the smaller the difference between the corresponding outputs will be. Based on this assumption, a cohesion index is proposed as follows to estimate the difference of outputs in an SPS based on Gamma test [28], [29].

$$C_i = \sum_{j=1}^{n} (V_0(t_j) - V_0(t))^2 \tag{2}$$

For each pattern P_{ti} $(1 \leqslant i \leqslant N)$ in the pattern database, the average cohesion index is written as

$$\overline{C} = \sum_{i}^{N} \frac{C_i}{2N} \tag{3}$$

The SPS for each pattern is built, and each SPS is associated with a cohesion index. Then, a relationship between the complete pattern database and the average cohesion index is written as follows.

$$\min \overline{C} = g(P) \tag{4}$$

In essence, this is a feature selection process to find the best structure of an input vector, which can minimize the average cohesion index of the pattern database. Therefore, we name the proposed method as a cohesion-based feature selection algorithm (CFSA).

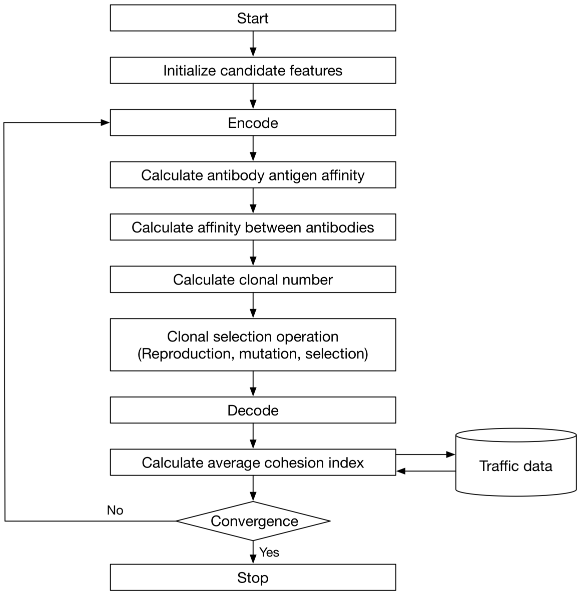
Start
Initialize candidate features
Encode
Calculate antibody antigen affinity
Calculate affinity between antibodies
Calculate clonal number
Clonal selection operation
(Reproduction, mutation, selection)
Decode
Calculate average cohesion index
Traffic data
No
Convergence
Yes
Stop

FIGURE 1. The framework of the proposed cohesion-based selection algorithm.

Since the feasible solution set is discrete and it is hard to explicitly formulate $g(\cdot)$ in Equation 4, the clonal selection algorithm[1] is employed to solve this problem. The framework of the proposed CFSA is introduced as follows (also see Figure 1).

A. CODING SCHEME

A binary coding scheme is adopted in our model. The length of chromosome equals to the length of the candidate features. Each entry of chromosome represents whether a feature is selected in an input vector (Figure 2).

B. ANTIBODY ANTIGEN AFFINITY

Antibody antigen affinity is the fitness function of a general evolutionary algorithm. It aims to minimize the average cohesion index for whole pattern database. The less the average cohesion index of each chromosome is, the larger the possibility that this chromosome is kept will be. Pseudo codes of computing the fitness function is in Algorithm 1.

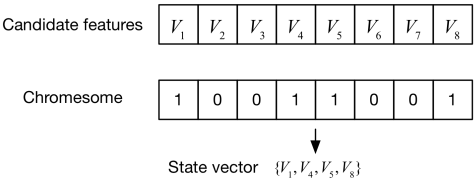
Candidate features
V_1 V_2 V_3 V_4 V_5 V_6 V_7 V_8
Chromesome
1 0 0 1 1 0 0 1
State vector $\{V_1, V_4, V_5, V_8\}$

FIGURE 2. Coding scheme.

Algorithm 1 Pseudo Codes of Computing the Fitness Function

```
Decode chromosome
Construct a pattern database in terms of decoded chromosome
foreach For each pattern in the pattern database $P_{ti}$ $(1 \leqslant i \leqslant N)$ do
    Find $k$ nearest patterns of $P_{ti}$ denoted as $P_{t_1} \sim P_{t_k}$;
    Calculate the cohesion index for each pattern by following Equation 2.
end
Calculate the average cohesion index for the pattern data by following Equation 3.
```

C. AFFINITY BETWEEN ANTIBODIES

The following Equation 5 is employed to calculate the affinity between antibodies to fully use the complementary advantages of different features and to illustrate the difference between the two feature selection combinations.

$$\theta_i = \min\{D_{ij}\} = \min\{\exp(||S_i - S_j||)\}, \quad i \neq j;\ i,\ j = 1, 2, \ldots, M \tag{5}$$

where M is the length of the vector. Note that the greater the difference, the greater the value; the better the diversity of the antibody group, the more conducive to the search of the optimal solution. In the following section, we use Hamming distance to measure the difference between two chromosomes.

D. CALCULATE CLONAL NUMBER

The clone number of each antibody is calculated by using the following Equation 6. The number of antibody clones is adjusted based the value of antibody antigen affinity and the value of affinity between antibodies.

$$q_i(k) = \left\lfloor \frac{\overline{C_i}}{\sum_{j=1}^{M} \overline{C_j}} \cdot \theta_i \right\rfloor, \quad i = 1, 2, \ldots, M \tag{6}$$

E. ACCELERATION TECHNIQUES

The calculation of one cohesion index is a typical K-nearest-neighbors (KNN) search problem, whereas the computation of the average cohesion index requires to conduct the KNN search for N times. Moreover, the genetic algorithm itself is a heuristic search procedure whose computation burden is usually high in computing the fitness function. The heavy computational burden will be a big problem when the number

[1] The clonal selection algorithm is an optimized search algorithm based on Evolutionary Algorithm and Artificial Immune System. It is suitable for finding an optimal or near optimal solution from a large-size discrete feasible set, particularly for high dimensional data.

of patterns is large. Therefore, the following two acceleration techniques are applied in the study.

1) APPLICATION OF AN ADVANCED SEARCHING DATA STRUCTURE

Suppose that a linear data structure is adopted to create the pattern database, and the time complexity is $O(N)$. Therefore, the total time complexity is $O(N^2)$. An advanced searching structure, i.e., KD tree,[2] is employed in this study.

2) REDUCTION OF THE TOTAL AMOUNTS OF THE KNN SEARCH

In the procedure of calculating the average cohesion index, all patterns in the database are taken into account. However, it may be not necessary for a pattern database with a large number of patterns. Sampling methods can be applied to reduce the number of patterns that need to search. Here, the density-biased sampling algorithm [32] is employed.

IV. EXPERIMENTS

A. EXPERIMENT DESIGN

In the Twin Cities, Minnesota, United States, over 4,000 double inductive loop detectors has been installed on its road network, and real-time traffic flow data are daily collected.[3] The data used here was collected at two sites on a section of I-35E South (i.e., Site I and Site II in Figure 3).

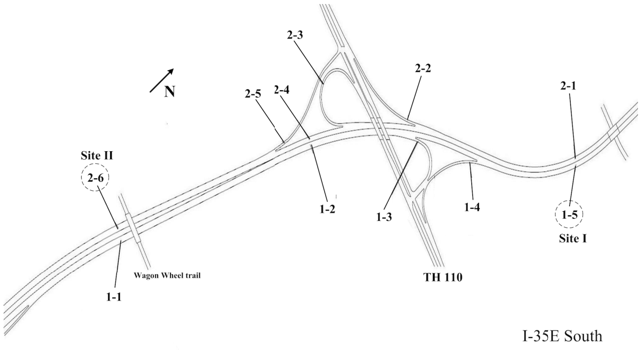

FIGURE 3. Locations where the data were collected.

For each detector, we use the data during 3/1/2012 and 3/31/2012 as the training data, and the data during 4/1/2012 and 4/5/2012 as test data. All data is averaged every five minutes. For each site, we test the one-step forecasting (i.e., forecasting the average flow of next 5 minutes, denoted by $V(t+1)$) and the two-steps forecasting (forecasting the average flow of next 5 10 minutes, denoted by $V(t+2)$). In total, there are four scenarios. Note that the data points that are less than 50 vehicles per interval are discarded in the experiment, since it is not valuable to forecast low-volume traffic flow.

For each scenario, the following three classes of input vectors are examined:

(1) **CFSA input vector**. The CFSA input vector is an optimized input vector using the proposed method. When applying the method, the neighbor number is set to 20 according to [20]. Euclidean distance is used to measure the distance between patterns, and 20 chromosomes are contained in the GA algorithm.

(2) **Time series (TS) input vector**. TS input vector refers to the classic input vector that is composed by the average flow of the current interval and previous three time intervals.

(3) **Naive spatial-temporal (NST) input vector**. This class of input vectors incorporates direct upstream flow and historical flow.

Two popular models, i.e., the KNN non-linear-regression (NPR) model and the support vector machine (SVM) model, are employed to test the effect of the three classes of input vectors.

Two indices are adopted to measure the performance of forecasting:

(1) **Mean absolute percentage error (MAPE)**. MAPE is a reflection of the overall performance. The definition is as follows:

$$\mathrm{MAPE} = \frac{1}{N}\sum_{t=0}^{N-1}\frac{|V(t+1)-\hat{V}(t+1)|}{V(t+1)} \tag{7}$$

where $\hat{V}(t+1)$ is the forecasted traffic volume, $V(t+1)$ is the ground-truth volume.

(2) **MAPE at leap points**. Tracking the abrupt change of the traffic flow is vital for traffic management, and thus the tracking capability is an important performance index of a forecasting algorithm. It can be measured by the MAPE at leap points. In detail, time t is considered to be a leap point, if the following condition is satisfied,

$$\frac{|V(t+1)-V(t)|}{V(t)} > \sigma \tag{8}$$

where σ is a threshold and set to 10% here.

B. EXPERIMENT RESULTS: SITE I

Experiments using the data collected at Site I are first conducted. The selected input vectors with three different configurations are listed in Table 1.

The performances of the one-step and two-step forecasting using Site I data are listed in Table 2. From the table, we can find that with the optimized input vector (i.e., CFSA input vector), all indices are lower for the NPR and SVM models.

To see more detail, the data collected from 4/5/2012 is taken as a sample to show the improvements. The time-series curve of traffic flow during the whole day is shown in Figure 4. It can be seen that there are two peak (congestion) periods. The forecasting results of the first two hours of morning peak (6:00-8:00, within the box) is selected to make a comparison of different input vectors.

[2] KD tree is first proposed in [30] to provide a data structure for fast multi-dimensional data search. It can be seen as an extension of the binary sort tree. One can refer to [30] and [31] for the details of creating KD tree and KD tree-based KNN search algorithm. It is proved in [30] that the time complexity of creating a KD tree is $O(N\log N)$. The time complexity of a KNN search operation on KD tree is $O(N\log N+K)$ and can be assumed to be $O(N\log N)$ since K is normally much less that N.

[3] The data can be downloaded from http://www.d.umn.edu/tdrl/traffic/

TABLE 1. Site I: selected input vectors.

		One-step forecasting	Two-step forecasting
CFSA	Candidate vector	$V_{1-1}(t) \sim V_{1-1}(t-4)$ $V_{1-2}(t) \sim V_{1-2}(t-4)$ $V_{1-3}(t) \sim V_{1-3}(t-4)$ $V_{1-4}(t) \sim V_{1-4}(t-4)$ $V_{1-5}(t) \sim V_{1-5}(t-4)$	$V_{1-1}(t) \sim V_{1-1}(t-4)$ $V_{1-2}(t) \sim V_{1-2}(t-4)$ $V_{1-3}(t) \sim V_{1-3}(t-4)$ $V_{1-4}(t) \sim V_{1-4}(t-4)$ $V_{1-5}(t) \sim V_{1-5}(t-4)$
	Optimized vector	$V_{1-2}(t), V_{1-2}(t-3)$ $V_{1-3}(t) \sim V_{1-3}(t-3)$ $V_{1-4}(t), V_{1-4}(t-2)$ $V_{1-5}(t-1)$	$V_{1-2}(t), V_{1-2}(t-3)$ $V_{1-3}(t-1), V_{1-3}(t-2)$ $V_{1-3}(t-3)$ $V_{1-4}(t) \sim V_{1-4}(t-3)$ $V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$
TS		$V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$	$V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$
NST		$V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$ $V_{1-4}(t), V_{1-4}(t-1)$ $V_{1-3}(t), V_{1-3}(t-1)$	$V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$ $V_{1-4}(t), V_{1-4}(t-1)$ $V_{1-3}(t), V_{1-3}(t-1)$

TABLE 2. Site I: forecasting errors.

		NPR		SVM	
		MAPE	MAPE at leap points	MAPE	MAPE at leap points
One-step	CFSA	0.096	0.212	0.096	0.211
	TS	0.109	0.240	0.113	0.236
	NST	0.101	0.234	0.105	0.223
Two-step	CFSA	0.118	0.231	0.116	0.238
	TS	0.133	0.250	0.131	0.261
	NST	0.127	0.246	0.129	0.251

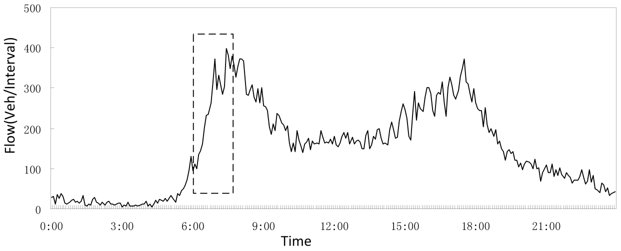

FIGURE 4. Empirical traffic flow time series of 4/5/2012 at Site I.

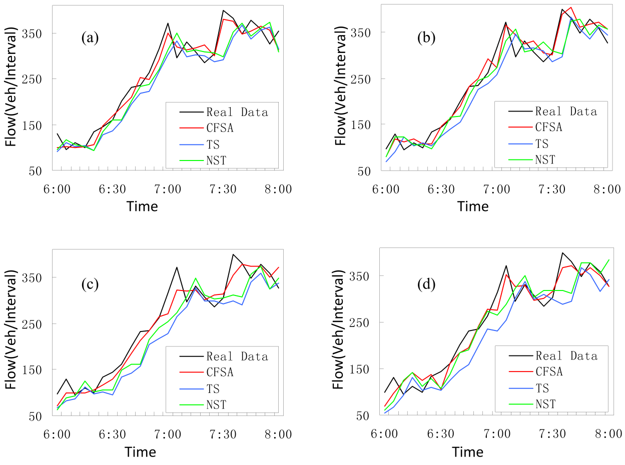

FIGURE 5. Visualized time-series of the forecasting results at Site I. (a) One-step, NPR. (b) One-step, SVM. (c) Two-step, NPR. (d) Two-step, SVM.

TABLE 3. Site II: selected input vectors.

		One-step forecasting	Two-step forecasting
CFSA	Candidate vector	$V_{2-1}(t) \sim V_{2-1}(t-4)$ $V_{2-2}(t) \sim V_{2-2}(t-4)$ $V_{2-3}(t) \sim V_{2-3}(t-4)$ $V_{2-4}(t) \sim V_{2-4}(t-4)$ $V_{2-5}(t) \sim V_{2-5}(t-4)$ $V_{2-6}(t) \sim V_{2-6}(t-4)$	$V_{2-1}(t) \sim V_{2-1}(t-4)$ $V_{2-2}(t) \sim V_{2-2}(t-4)$ $V_{2-3}(t) \sim V_{2-3}(t-4)$ $V_{2-4}(t) \sim V_{2-4}(t-4)$ $V_{2-5}(t) \sim V_{2-5}(t-4)$ $V_{2-6}(t) \sim V_{2-6}(t-4)$
	Optimized vector	$V_{2-1}(t)$ $V_{2-2}(t) \sim V_{2-2}(t-3)$ $V_{2-3}(t-2), V_{2-3}(t-3)$ $V_{2-4}(t-2)$ $V_{2-5}(t), V_{2-5}(t-1)$ $V_{2-5}(t-2)$ $V_{2-6}(t-1)$	$V_{2-1}(t), V_{2-1}(t-1)$ $V_{2-2}(t), V_{2-2}(t-3)$ $V_{2-3}(t-2), V_{2-3}(t-4)$ $V_{2-4}(t-2), V_{2-4}(t-3)$ $V_{2-4}(t-4)$ $V_{2-5}(t-1), V_{2-5}(t-2)$ $V_{2-5}(t-3)$ $V_{2-6}(t)$
TS		$V_{1-6}(t), V_{1-6}(t-1)$ $V_{1-6}(t-2)$	$V_{1-6}(t), V_{1-6}(t-1)$ $V_{1-6}(t-2)$
NST		$V_{1-6}(t), V_{1-6}(t-1)$ $V_{1-6}(t-2)$ $V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-4}(t), V_{1-4}(t-1)$	$V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-5}(t-2)$ $V_{1-5}(t), V_{1-5}(t-1)$ $V_{1-4}(t), V_{1-4}(t-1)$

TABLE 4. Site II: forecasting errors.

		NPR		SVM	
		MAPE	MAPE at leap points	MAPE	MAPE at leap points
One-step	CFSA	0.095	0.198	0.092	0.208
	TS	0.112	0.237	0.115	0.231
	NST	0.105	0.227	0.109	0.228
Two-step	CFSA	0.120	0.221	0.120	0.235
	TS	0.134	0.245	0.131	0.253
	NST	0.131	0.243	0.128	0.251

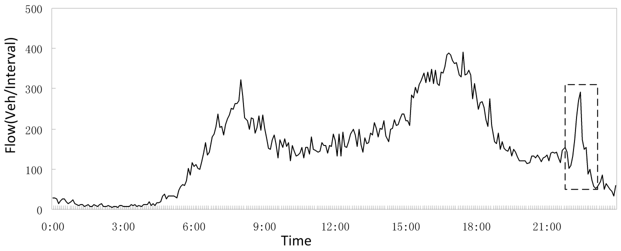

FIGURE 6. Empirical traffic flow time series of 4/5/2012 at Site II.

Figure 5 compares the forecasted traffic flow with the ground-truth data. It can be seen that in the rapid increasing region, the forecasting algorithm with the CFSA input vector performs much better than the those with the TS and NST input vectors. With the CFSA input vector, the forecasted flow tracks the empirical flow better, while the results using the other two input vectors have obvious time lags.

C. EXPERIMENT RESULTS: SITE II

To make the results solider, we use the data at Site II and re-conduct the experiments. The selected input vectors with three different configurations are listed in Table 3.

The performances of the one-step and two-step forecasting at Site II are listed in Table 4. It can be also seen from the table that the optimized input vector (i.e., CFSA input vector) results in lower performance indices.

As in the previous case, we present the improvements in Figure 6 by taking the traffic flow data on 4/5/2012 as an example. A sharp peak is found at approximately 22:00, which is a non-recurrent pattern, since it doesn't appear on the previous days. Thus, it is a good scenario to examine the performance of forecasting.

Figure 7 compares the forecasted traffic flow with the ground-truth data. For the one-step forecasting, the predicted flow during the onset phase of the congestion shows a small lag, which is not a good forecasting compared with the one during the offset phase. The forecasting results with the

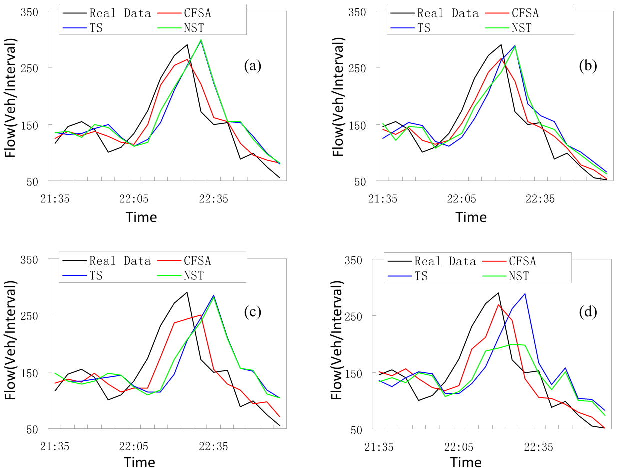

FIGURE 7. **Visualized time-series of the forecasting results at Site II. (a) One-step, NPR. (b) One-step, SVM. (c) Two-step, NPR. (d) Two-step, SVM.**

CFSA input vector are able to follow the abrupt changes of traffic flow for both the onset and offset phases of a peak. The forecasting results with the TS and NST input vectors exhibit obvious lags during the onset phase. In the two-step forecasting, all results are not quite ideal compared with the results of the one-step forecasting. However, the results with the CFSA input vector are still better than those with the TS and NST input vectors.

D. ANALYSIS OF THE EXPERIMENT RESULTS

We summarize the findings from the experiments as follows.

(1) The proposed method successfully improves the performance of both the NPR and SVM traffic flow forecasting, measured by using MAPE and MAPE at leap points. The MAPE is reduced by 1%∼2%, and the MAPE at leap points is reduced by 2%∼5%.

(2) The MAPE at leap points is very high. In fact, forecasting the abrupt change of traffic flow is still a challenge for most of the existing approaches. However, after applying our proposed input vector selection method, the forecasting method can better track the evolution of traffic flow.

(3) Different input vectors are selected for 5 min and 10 min forecasting tasks, while in some existing works, the same input vector is used. This fact indicates that we should customize input vector (maybe other parameters) for different requirement of forecasting.

V. CONCLUSION

Short-term traffic flow forecasting is important for many ITS applications. Comparing with the fruitful forecasting algorithms, the research on the feature selection (i.e. how to design the input vector) remains elusive. Moreover, with the rapid development of information technology, more and more data is collected and can be used in forecasting. A preprocess method that can ‘‘filter’’ the large amount of data has become an urgent need.

With the above motivation, this paper proposes a cohesion-based feature selection method to design the input vector for nonparametric short-term traffic flow forecasting. Experiments are conducted based on real-world traffic flow data. The results show that the proposed feature selection method can reduce forecasting errors under various scenarios.

The main shortcoming of the proposed method is the high computational burden, since it is a heuristic method and a complex searching procedure is adopted to obtain the fitness value. However, we believe that it is not a severe problem in practical environments, due to the following considerations: (1) The feature selection is an off-line preprocessing procedure; (2) The advanced data structure and distributed architectures can greatly accelerate the calculation; (3) The development of computer hardware will greatly improve the computing power; (4) The state space of this problem is relatively small, compared to many complex optimization problems. Thus there will not be many iterations before convergence.

There are several interesting research directions regarding the study. First, only one variable is employed in this study, and thus multivariate analysis can be tested in future. Second, some other factors can be considered in the objective function, e.g. the length of input vectors. This is because longer vectors mean more time consumption and less matched patterns in real-time forecasting. More factors may turn the problem into a multi-objective problem, and a multi-objective genetic algorithm such as NSGA II could be a better solution.

REFERENCES

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, ‘‘Short-term traffic forecasting: Where we are and where we're going,’’ *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[2] U. Mori, A. Mendiburu, M. Álvarez, and J. A. Lozano, ‘‘A review of travel time estimation and forecasting for Advanced Traveller Information Systems,’’ *Transportmetrica A, Transport Sci.*, vol. 11, no. 2, pp. 119–157, 2015.

[3] Z. He, L. Zheng, P. Chen, and W. Guan, ‘‘Mapping to cells: A simple method to extract traffic dynamics from probe vehicle data,’’ *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 3, pp. 252–267, 2017.

[4] Z. Zhang, Y. Wang, P. Chen, Z. He, and G. Yu, ‘‘Probe data-driven travel time forecasting for urban expressways by matching similar spatiotemporal traffic patterns,’’ *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 476–493, Dec. 2017.

[5] L. Lu, J. Wang, Z. He, and C.-Y. Chan, ‘‘Real-time estimation of freeway travel time with recurrent congestion based on sparse detector data,’’ *IET Intell. Transp. Syst.*, vol. 12, no. 1, pp. 2–11, Feb. 2018.

[6] L. Lin, Z. He, and S. Peeta, ‘‘Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach,’’ *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 258–276, Dec. 2018.

[7] M. S. Ahmed and A. R. Cook, ‘‘Analysis of freeway traffic time-series data by using Box–Jenkins techniques,’’ *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[8] M. Levin and Y.-D. Tsao, ‘‘On forecasting freeway occupancies and volumes,’’ *Transp. Res. Rec.*, no. 773, pp. 47–49, 1980.

[9] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, ‘‘Short-term prediction of traffic volume in urban arterials,’’ *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.

[10] J. W. C. Van Lint, "Online learning solutions for freeway travel time prediction," *IEEE Transp. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 38–47, Jan. 2008.

[11] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[12] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res. C Emerg. Technol.*, vol. 19, no. 4, pp. 606–616, 2011.

[13] A. Stathopoulos and G. M. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transp. Res. C, Emerg. Technol.*, vol. 11, no. 2, pp. 121–135, 2003.

[14] T. T. Tchrakian, B. Basu, and M. O'Mahony, "Real-time traffic flow forecasting using spectral analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 519–526, Jun. 2012.

[15] D. Bosq, *Nonparametric Statistics for Stochastic Processes* (Lecture Notes in Statistics). New York, NY, USA: Springer-Verlag, 1996. [Online]. Available: https://www.amazon.com/Nonparametric-Statistics-Stochastic-Processes-Estimation/dp/0387985905

[16] R. K. Oswald, "Traffic flow forecasting using approximate nearest neighbor nonparametric regression," Center Transp. Stud., Univ. Virginia, Charlottesville, VA, USA, Tech. Rep. UVA-CE-ITS_01-4, Dec. 2001. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/15834

[17] S. Clark, "Traffic prediction using multivariate nonparametric regression," *J. Transp. Eng.*, vol. 129, no. 2, pp. 161–168, Mar. 2003.

[18] X. Gong and F. Wang, "Three improvements on KNN-NPR for traffic flow forecasting," in *Proc. IEEE 5th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2002, pp. 736–740.

[19] T. Kim, H. Kim, and D. J. Lovell, "Traffic flow forecasting: Overcoming memoryless property in nearest neighbor non-parametric regression," in *Proc. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2005, pp. 965–969.

[20] R. E. Turochy, "Enhancing short-term traffic forecasting with traffic condition information," *J. Transp. Eng.*, vol. 132, no. 6, pp. 469–474, 2006.

[21] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 211–234, 2005.

[22] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Spatio-temporal short-term urban traffic volume forecasting using genetically optimized modular networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 22, no. 5, pp. 317–325, 2007.

[23] S. Yang, "On feature selection for traffic congestion prediction," *Transp. Res. C, Emerg. Technol.*, vol. 26, pp. 160–169, Jan. 2013.

[24] E. I. Vlahogianni and M. G. Karlaftis, "Fuzzy-entropy neural network freeway incident duration modeling with single and competing uncertainties," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 28, no. 6, pp. 420–433, 2013.

[25] N. Dong, H. Huang, and L. Zheng, "Support vector machine in crash prediction at the level of traffic analysis zones: Assessing the spatial proximity effects," *Accident Anal. Prevention*, vol. 82, pp. 192–198, Sep. 2015.

[26] Y.-R. Shiau, C.-H. Tsai, Y.-H. Hung, and Y.-T. Kuo, "The application of data mining technology to build a forecasting model for classification of road traffic accidents," *Math. Problems Eng.*, vol. 2015, Jun. 2015, Art. no. 170635.

[27] L. Li, X. Su, Y. Wang, Y. Lin, Z. Li, and Y. Li, "Robust causal dependence mining in big data network and its application to traffic flow predictions," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 292–307, Sep. 2015.

[28] A. Stefánsson, N. Končar, and A. J. Jones, "A note on the Gamma test," *Neural Comput. Appl.*, vol. 5, no. 3, pp. 131–133, 1997.

[29] A. P. M. Tsui, A. J. Jones, and A. G. De Oliveira, "The construction of smooth models using irregular embeddings determined by a Gamma test analysis," *Neural Comput. Appl.*, vol. 10, no. 4, pp. 318–329, 2002.

[30] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1997.

[31] J. L. Bentley and J. H. Friedman, "Data structures for range searching," *ACM Comput. Surv.*, vol. 11, no. 4, pp. 397–409, 1979.

[32] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold, "Efficient biased sampling for approximate clustering and outlier detection in large data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 5, pp. 1170–1187, Sep. 2003.

LISHAN LIU received the B.S. degree from the College of Automation Engineering, Tianjin University, China, in 2009, where he is currently pursuing the Ph.D. degree with the College of Management and Economics. His research interest includes urban traffic control and management.

NING JIA received the B.S. degree from the Department of Management, Shandong University, China, in 2005, and the Ph.D. degree from the Institute of Systems Engineering, Tianjin University, China, in 2010, with a thesis on traffic flow modeling. Since 2010, he has been with the Institute of Systems Engineering, Tianjin University, where he is currently a Professor with the College of Management and Economics. He has authored more than 20 peer-reviewed papers. His research interests include ITS and urban traffic control and management.

LEI LIN received the B.S. degree in traffic and transportation and the M.S. degree in systems engineering from Beijing Jiaotong University, China, in 2008 and 2010, respectively, and the M.S. degree in computer science and the Ph.D. degree in transportation systems engineering from the University at Buffalo, The State University of New York, Buffalo, in 2013 and 2015, respectively. From 2013 to 2015, he was a Research Assistant with the Transportation Informatics Tier 1 University Transportation Center. He was a Researcher with Xerox from 2015 to 2017. He was a Research Associate with the NEXTRANS Center, Purdue University. Since 2018, he has been a Research Scientist with the Goergen Institute for Data Science, University of Rochester. His research interests include transportation big data, machine learning applications in transportation, and connected and automated transportation.

ZHENGBING HE received the B.A. degree in English language and literature from the Dalian University of Foreign Languages, China, in 2006, and the Ph.D. degree in systems engineering from Tianjin University, China, in 2011. From 2011 to 2017, he was a Postdoctoral Researcher and an Assistant Professor with the School of Traffic and Transportation, Beijing Jiaotong University, China. He is currently a Distinguished Young Professor with the College of Metropolitan Transportation, Beijing University of Technology, China.

He is always interested in solving various transportation problems by combining empirical (big) data with the knowledge of traffic flow theory and intelligent transportation systems. Over the last five years, his first-author papers have been published in the IEEE Transactions on Intelligent Transportation Systems, the IEEE Transactions on Intelligent Vehicles, *Transportation Research Part B*, *Transportation Research Part C*, *Computer-aided Civil and Infrastructure Engineering*, *Transportmetrica B*, the *ASCE Journal of Transportation Engineering*, the *Journal of Intelligent Transportation Systems*, and *Transportation Letters*. He is an Associate Editor of the IEEE Access and a Guest Editor of *Transportation Research Part C*, the *Journal of Intelligent Transportation Systems*, and *Transportmetrica A*. His webpage is http://www.zhengbinghe.com.