

Research Article

Urban Traffic Flow Prediction Model with CPSO/SSVM Algorithm under the Edge Computing Framework

Fengkai Liu ^{1,2}, Xingmin Ma,² Xingshuo An,² and Guangnan Liang²

¹College of Computer Science and Technology, Zhejiang University, 310027 Hangzhou, China

²North China Institute of Computing Technology, NCI, 100083 Beijing, China

Correspondence should be addressed to Fengkai Liu; 11821127@zju.edu.cn

Received 23 April 2020; Revised 28 June 2020; Accepted 3 August 2020; Published 1 September 2020

Academic Editor: Ke Xiong

Copyright © 2020 Fengkai Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Urban traffic flow prediction has always been an important realm for smart city build-up. With the development of edge computing technology in recent years, the network edge nodes of smart cities are able to collect and process various types of urban traffic data in real time, which leads to the possibility of deploying intelligent traffic prediction technology with real-time analysis and timely feedback on the edge. In view of the strong nonlinear characteristics of urban traffic flow, multiple dynamic and static influencing factors involved, and increasing difficulty of short-term traffic flow prediction in a metropolitan area, this paper proposes an urban traffic flow prediction model based on chaotic particle swarm optimization algorithm-smooth support vector machine (CPSO/SSVM). The prediction model has built a new second-order smooth function to achieve better approximation and regression effects and has further improved the computational efficiency of the smooth support vector machine algorithm through chaotic particle swarm optimization. Simulation experiment results show that this model can accurately predict urban traffic flow.

1. Introduction

The concept of a smart city has been quite popular in recent years as it demonstrates great potential for improving urban management and people's life [1]. Various sensing and computing technologies have gradually outlined the future of a smart city. In the smart city scenario, the combined application of multiple technologies has generated a huge amount of data, which has laid the basic foundation of a smart city.

With the surge of data volume, cloud computing has encountered unforeseen challenges [2]. Firstly, due to the huge amounts of data instantly acquired from multiple sources, the sensing layer of a smart city has shown special characteristics such as strong redundancy, inner and interconnectivity, sensitivity to transmission delay, and heterogeneous. Therefore, it is difficult for cloud-centralized service to handle such large-scale data under high pressure [3]. Secondly, cloud computing is a cohesion model; it will lead to

network delay and waste of bandwidth once data is sent from edge to cloud through a remote network. Finally, the data security and privacy protection of a smart city during long-distance transmission are facing great challenge [4], since a centralized cloud-computing model makes it difficult to ensure security and privacy of data transmitted from distributed edges.

The paradigm to handle the aforementioned issues is edge computing. "Edge" indicates proximity to the user or the sources which generate data. Therefore, edge computing mainly provides computation, data storage, and network support from the edge. The tasks of storage and computation have been transferred to the edge instead of servers from the cloud, as indicated in Figure 1.

Edge computing has demonstrated great advantages from the perspectives of wide connection, distributed computation, proximity to a data source, and low latency. Edge computing can provide better capability in the areas of data

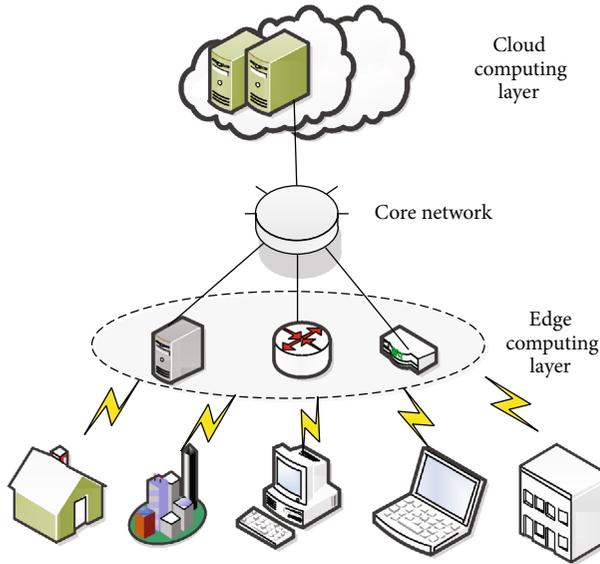


FIGURE 1: Technical architecture for smart city edge computing.

filtering and compression, situational awareness, and data classification, which has laid a strong technical foundation for the application of big data analysis, traffic management, and urban environmental monitoring [5].

There are some differences between edge data processing and traditional data processing techniques. Firstly, the core of smart city data is heterogeneous in nature. Smart city data normally come from areas of governance, public security, environment, transportation, internet, and IOT, all of which generate data from multiple sources in different modalities. Secondly, restriction on the access of the edge node should be taken into consideration during the process of edge processing. Lastly, the cooperation between edge and cloud needs to be taken into consideration [6].

Therefore, the paper focuses on the urban data processing on the edge side, particular small-scale urban data though not huge but significant in daily urban governance operation, and successful application of these data would eventually lead to the informed decision during urban governance and planning. The contribution of the article includes the following: (1) proposed a data application framework for urban edge computing based on the research of requirements of urban data processing and (2) proposed a new CPSO/SSVM algorithm, which has built a new 2nd-order smooth function to achieve a better effect on approximation and regression. Meanwhile, CPSO optimization has further improved the efficiency of SVM. The prediction result is satisfactory on accuracy and stability when processing traffic flow data from the edge. The remaining structure of the paper is as follows: part 2 has introduced related work on edge computing and traffic flow prediction, part 3 designed the urban traffic prediction model under the edge computing framework, part 4 has tested the model, and part 5 has given the conclusion.

2. Related Works

Big data and edge computing are both hotspots in the academia, and some scholars have begun to pay attention to big

data processing at the edge. Yang and Liu [7] discussed the advantages of edge computing. To maximize the application of edge computing, data fusion is used in the framework of edge computing. The authors have proposed a Gaussian process-based temporal data fusion (GPTDF) method aimed for the issue of sequential online forecasting at the edge. This approach provides an effective data fusion tool for data capture and privacy protection of edge computing. Big data applications operating under an edge computing environment have their own unique characteristics. Ndikumana et al. [8] put forward the concept of joint 4C (computing, caching, communication, and control) to describe this feature. Joint 4C is transformed into an optimization problem and solved in the research. Considering the complexity and difference of mobile big data, Xu et al. [9] propose a computation offloading method, named COM, for IoT-enabled cloud-edge computing. In order to solve the issue of data sharing and collaboration in edge computing, Zhang et al. [10] propose a new computing paradigm named Firework. Firework provides a balanced environment where data can be shared, distributed, and processed for big data application; in the meantime, computing power is kept within stakeholder's information infrastructure.

The application of data processing in smart cities has also received considerable attention in recent years. Lau et al. [11] introduce a multiperspective classification of the data processing to evaluate the smart city applications in their overview article on smart cities. Then, the aforementioned method has been applied to evaluate a group of selective applications in each domain of the smart city. Jara et al. [12] fused traffic and temperature data in their study. They also proposed a temperature-traffic prediction model that obeys the Poisson distribution.

The prediction algorithm for urban traffic has also received considerable attentions in recent years. Jiang and Adeli [13] have proposed a novel nonparametric dynamic time-delay recurrent wavelet neural network model to predict traffic flow. Lv et al. [14] have proposed a novel method to predict a traffic flow-based deep-learning algorithm, which considers the correlation within the spatiotemporal traffic network. The test result has shown a superior outcome in traffic flow prediction. Zhang et al. [15] has put forward an algorithm framework based on deep learning; the algorithm predicts the traffic network's node flow and edge flow concurrently under a spatiotemporal scenario. Test results have demonstrated the advantages of methodologies with an outcome above 11 baselines, such as ConvLSTM, CNN, and Markov random field. Yuan et al. [16] have put forward a novel method to analyze Shanghai urban traffic by designing a key mode of urban traffic and predicting transition between different modes. Experimental results have demonstrated the proposed method to predict that traffic mode is more accurate and comprehensive.

3. Materials and Methods

3.1. Urban Traffic Data Application Architecture under the Edge Computing Framework. Based on the inherent properties of traffic flow in the urban area as well as factors

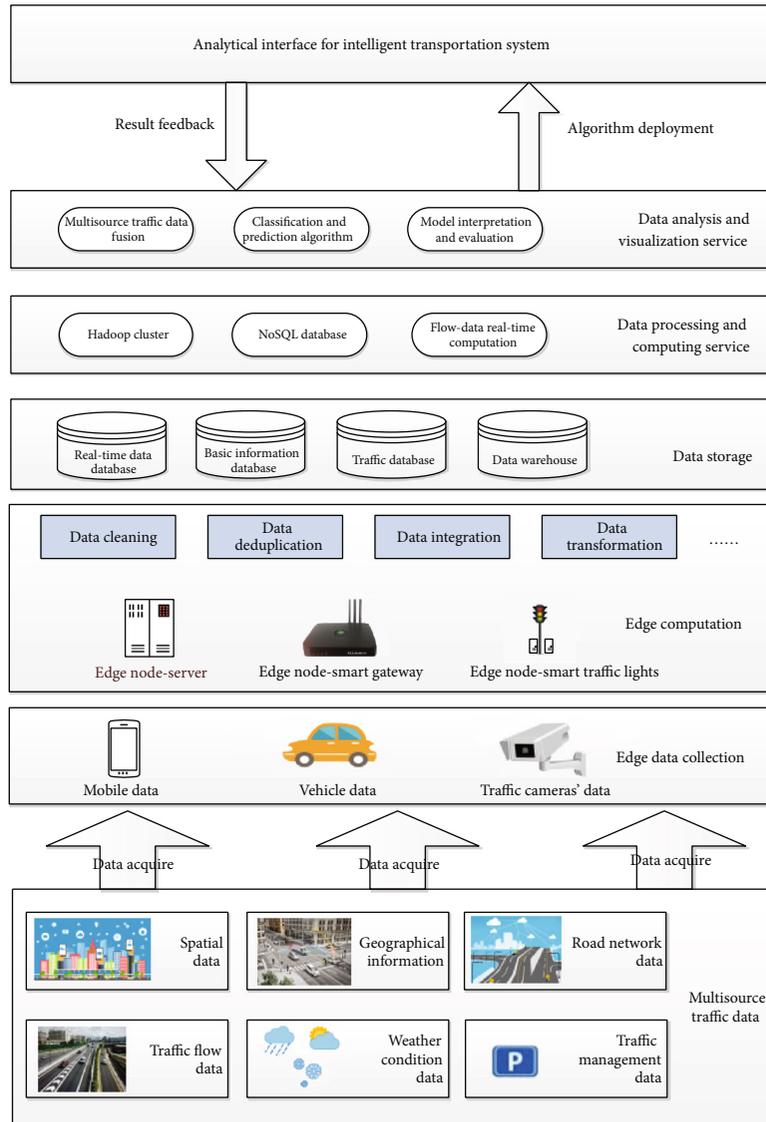


FIGURE 2: Technical architecture for urban traffic big data.

influencing the efficiency of the intelligent transportation system (ITS), such as proximity to the edge, flowability, and heterogeneous, the paper proposed an application architecture of urban traffic data and designed a traffic prediction model to better handle traffic flow data from the edge. The overall architecture includes five layers including edge data collection, edge computation, data storage, data processing and computing, and data analysis and visualization, as illustrated in Figure 2.

Edge data collection layer is the “entrance” of data for the overall technical architecture; it is mainly responsible for the collection of data from road traffic network. Real-time traffic data are mainly generated by ring induction coils, toll bayonet, car GPS, etc. The amount of data is on a large scale with heterogeneous in nature. At this layer, it is necessary to build a channel for data acquisition, so that traffic big data can converge to the edge nodes.

Edge computation layer will provide a timely response from close proximity at the edge of the network; data fusion

methodologies are normally applied at this layer to deal with heterogeneous traffic information collected from multiple sources. At the same time, preliminary data preprocessing is also required at the edge computing layer for data quality control. The edge node is responsible for processing data in a larger area of the local area network and provides extensible data processing capabilities. The edge computing layer mainly performs tasks such as data cleaning, data integration, and data deduplication.

The data storage layer will store the data extracted from the edge data collection layer, and the extracted data will be temporarily or permanently stored in the edge device. In the urban traffic big data scenario, the data stored at this layer is divided into three categories, including traffic flow data, weather data, and street view data from different POIs. Historical traffic flow data is used to train traffic flow prediction models, and real-time traffic flow data is used to evaluate prediction effects; improvements on the model will be made

TABLE 1: Experimental data type.

Data source	Data type	No. of indicators	Content of indicators
Traffic flow data	Continuous variable	1	Traffic flow
Weather data	Discrete variable	4	Fog, haze, rain, snow
POI data	Discrete variable	10	Adjacent traffic nodes with 500 meters, no. of shopping malls, no. of schools, no. of hospitals, no. of tourist sites, no. of bus stations, no. of restaurants, no. of hotels, no. of supermarkets

accordingly based on evaluation results. This layer provides data support for the computing service layer. The data required by the computing service layer comes from this layer.

The data processing and computing service layer is the core functional layer of the architecture. The purpose of this architecture is to provide users with accurate traffic flow prediction. This layer provides traffic flow preprocessing function and traffic flow prediction algorithm library. The preprocessing function is mainly realized by the system automatically. A variety of prediction algorithms will be provided including SVM smoothing algorithm, traditional support vector regression (SVR) algorithm, SVR with Chaotic Genetic Algorithm (CGA-SVR), Back Propagation Neural Network (BPNN), Autoregressive Integrated Moving Average model (ARIMA), and other traffic flow prediction models; users can apply the corresponding algorithm for traffic flow prediction based on different scenarios.

Data analysis and visualization service layer will interact with the analytical interface for ITS. With the support of the data processing and computing service layer, prediction results can be obtained in a more efficient way. Meanwhile, analytical methods and visualization methods enhance users' ability to acquire in-depth information. The RESTful architecture is used between the computing service layer and the visualization service layer, which can achieve loosely coupled connections between modules. Through the data analysis and visualization service layer, a more flexible and comprehensive interaction between the user and the data is achieved.

The combination of edge computing and data analytics is powerful as it can provide edge users a timely and accurate decision support when it comes to traffic management during rush hours. By deploying intelligent algorithms close to the edge computing layer, analytical results can be quickly shared across edge networks, which is vital to ITS as traffic flow management is highly related to efficient information sharing, and safety on the road can be guaranteed by providing accurate and timely road traffic feedback to the drivers. The urban traffic flow prediction model can be applied in the urban traffic data application architecture in two ways. Firstly, it can be deployed in the edge computing layer to conduct some instant analytical tasks in order to predict fluctuating traffic flow; the results can be shared with drivers through edge networks in a timely manner. Secondly, the model can be applied in the data analysis and visualization service layer in ITS's enterprise cloud in order to conduct analysis over heterogeneous traffic data from multiple sources and put forward informed suggestions to key stake-

holders who have overseen the entire metropolitan traffic network.

3.2. Design of Urban Traffic Indicators. Urban traffic flow is a crucial part of smart city management with multiple influencing factors involved. When it comes to the metropolitan area, the issue can be even more complex. The factors influencing urban traffic flow normally include flow from adjacent traffic nodes, weather condition, and point of interest (POI), such as nearby school, hospital, and shopping mall, thanks to the multiple traffic data acquired from the edge node, which includes spatial data, geographical information, road network data, traffic flow data, weather condition data, and traffic management data. The paper is based on traffic flow data from Guiyang City, with a spatial span of 717 intersections and a temporal span of 6 months; the experimental data type is shown in Table 1. A group of urban traffic indicators is built to evaluate factors that are influencing urban traffic flow.

The indicators of weather data are as follows:

- (1) Fog: the level of fog can be graded as minor fog, fog, heavy fog, dense fog, and heavy dense fog
- (2) Haze: the level of haze can be graded as light haze, haze, and heavy haze
- (3) Rain: the level of rain can be graded as light rain, medium rain, heavy rain, storm rain, and heavy storm rain
- (4) Snow: the level of snow can be graded as light snow, medium snow, and heavy snow

3.3. Methodology for the CPSO/SSVM Urban Traffic Flow Prediction Model. The urban traffic prediction model based on various kinds of machine learning methodologies is key to the construction of ITS. It is able to provide technical support for urban traffic management especially flow control of busy traffic nodes in the metropolitan area. Urban traffic is a complex system with high dynamic, which makes it difficult to analyze in a short-time manner. Increasing randomness within the urban intelligent system makes it difficult for traffic flow prediction. Short-time traffic flow is the key part of urban traffic big data; the basic characteristics of short-time traffic flow include non-linearity, randomness, and uncertainty.

Currently, urban traffic control and route guidance is mostly applied on a preset manner; only a few cities apply self-adaptive control mode during traffic flow detection. In

order to make up for the inefficiency, various machine learning methodologies have been introduced to build up related prediction techniques. SVM has been widely applied in the area of traffic flow prediction. With the method of structural risk minimization, SVM has great advantages on overcoming problems such as a small sample, nonlinearity, curse of dimensionality, oversimulation, and local minimization, which simplifies the problem of classification and regression during traffic flow analysis. Therefore, SVM has shown a promising future in intelligent traffic control and guidance, which can ease the issue of traffic congestion in the metropolitan area.

The paper has proposed an urban traffic prediction model based on CPSO/SSVM, which is able to predict short-time traffic flow at city intersection by considering multiple factors including POI and weather condition and acquire better prediction result compared to tradition SVM algorithm.

The standard SVM algorithm is as follows [17]:

$$\begin{aligned} \min_{w,b,C} \frac{\|w\|^2}{2} + Ce^T y \\ \text{s.t. } D \cdot (Aw + eb) \geq e \\ y \geq 0. \end{aligned} \quad (1)$$

In 2005, Lee et al. from Taiwan University introduced the concept of smooth function to improve SVM by introducing nondifferentiable function [18]; the formula is as follows:

$$\min_{w,b,C} \frac{1}{2} * C \| (e - D(Aw - eb))_+ \|_2^2 + \frac{1}{2} * (w^T w + b^2). \quad (2)$$

There is a nondifferentiable function in the objective function, which has shown a strong rotundity and unique solution; however, its nondifferentiable function is not smooth; therefore, a smooth function is required to infinitely approach a nondifferentiable function. Lee et al. have performed integral processing over a sigmoid function [18] and acquire

$$p(x, k) = x + \frac{1}{k} * \lg \left(1 + e^{-kx} \right). \quad (3)$$

By taking an integral function of S function as smooth function, smooth processing has been performed over a nondifferentiable part in formula (2), which therefore acquired initial SSVM.

$$\min_{w,b,C} \frac{1}{2} * C \| p(e - D(Aw - eb), k) \|_2^2 + \frac{1}{2} * (w^T w + b^2). \quad (4)$$

With the introduction of SSVM, it can replace nondifferentiable function by introducing a different smooth function to achieve the effect of smooth processing, which resulted to lots of smooth functions with a good approximation effect, which in turn resulted to several SSVM algorithms.

In 2005, Yuan et al. from UESTC has proposed a function as follows [19]:

$$T(x, k) = \begin{cases} 0, & x < -\frac{1}{k}, \\ \frac{k^2}{6}x^3 + \frac{k}{2}x^2 + \frac{1}{2}x + \frac{1}{6k}, & -\frac{1}{k} < x < 0, \\ -\frac{k^2}{6}x^3 + \frac{k}{2}x^2 + \frac{1}{2}x + \frac{1}{6k}, & 0 \leq x \leq \frac{1}{k}, \\ x, & x > \frac{1}{k}. \end{cases} \quad (5)$$

In 2013, Wu et al. from XUPT has proposed two 2nd-order smooth functions as follows [20, 21]:

$$\varphi(x, k) = \begin{cases} 0, & x < -\frac{1}{3k}, \\ \frac{3}{2}k^2 \left(\frac{1}{3k} + x \right)^3, & -\frac{1}{3k} \leq x < 0, \\ x + \frac{3}{2}k^2 \left(\frac{1}{3k} - x \right)^3, & 0 \leq x \leq \frac{1}{3k}, \\ x, & x > \frac{1}{3k}, \end{cases} \quad (6)$$

$$G(x, k) = \begin{cases} 0, & x < -\frac{1}{4k}, \\ \frac{8}{3}k^2x^3 + 2kx^2 + \frac{1}{2}x + \frac{1}{24k}, & -\frac{1}{4k} \leq x < 0, \\ -\frac{8}{3}k^2x^3 + 2kx^2 + \frac{1}{2}x + \frac{1}{24k}, & 0 \leq x \leq \frac{1}{4k}, \\ x, & x > \frac{1}{4k}. \end{cases} \quad (7)$$

By a piecewise smooth function from formula (6) and (7) to an approximate nondifferentiable part in formula (2), the approximation effect and final regression effect of the function have shown that the accuracy of formula (7) is more superior than formula (6) and (7). The paper proposed a new type of 2nd smooth function; the approximation effect and final regression effect of the function are more superior than formula (6) and (7); the proposed smooth function is as follows:

$$M(x, k) = \begin{cases} 0, & x < -\frac{1}{7k}, \\ \frac{49}{6}k^2x^3 + \frac{7}{2}kx^2 + \frac{1}{2}x + \frac{1}{42k}, & -\frac{1}{7k} \leq x < 0, \\ -\frac{49}{6}k^2x^3 + \frac{7}{2}kx^2 + \frac{1}{2}x + \frac{1}{42k}, & 0 \leq x \leq \frac{1}{7k}, \\ x, & x > \frac{1}{7k}. \end{cases} \quad (8)$$

The paper has set up a new SSVM algorithm (Ma-Liu Piecewise Smooth Support Vector Machine, MLSSVM) as follows:

$$\min_{w,b,C} \frac{1}{2} * C \|M(e - D(Aw - eb), k)\|_2^2 + \frac{1}{2} * (w^T w + b^2). \quad (9)$$

Theorem 1. The smooth function $M(x, k)$ is as formula (8); $x_+ = \max\{0, x\}$ indicates the nondifferentiable function, and smooth function $M(x, k)$ satisfies the following:

- (1) $M(x, k)$ is about x 2^{nd} -order smooth
- (2) $M(x, k) \geq x_+$
- (3) For $x \in R$, $M(x, k)^2 - x_+^2 \leq 1/821k^2$

Proof.

- (1) When $x = \pm 1/7k$ and $x = 0$, condition holds true for function $M(x, k)$ as follows: $M(-1/7k, k) = 0$, $\lim_{x \rightarrow 0^-} M(x, k) = \lim_{x \rightarrow 0^+} M(x, k)$, $M(1/7k, k) = 1/7k$, $M'(-1/7k, k) = 0$, $\lim_{x \rightarrow 0^-} M'(x, k) = \lim_{x \rightarrow 0^+} M'(x, k)$, $M'(1/7k, k) = 1$, $M''(-1/7k, k) = 0$, $\lim_{x \rightarrow 0^-} M''(x, k) = \lim_{x \rightarrow 0^+} M''(x, k)$, $M''(1/7k, k) = 0$. Therefore, $M(x, k)$ is 2^{nd} -order smooth about x
- (2) When $x \geq 1/7k$ and $x \leq -1/7k$, $M(x, k) \geq x_+$ holds true obviously $x \in (0, 1/7k)$; $M(x, k) - x_+ \geq M(1/7k, k) - x_+ = 0$; $x \in (-1/7k, 0)$, $M(x, k) - x_+$ shows incremental properties, $M(x, k) - x_+ \geq M(-1/7k, k) = 0$; therefore, $M(x, k) \geq x_+$
- (3) When $x \geq 1/7k$ and $x \leq -1/7k$, $M(x, k)^2 - x_+^2 \leq 1/821k^2$ holds true obviously $x \in (-1/7k, 0)$, $M(x, k)^2 - x_+^2 = M(x, k)^2 \leq M(0, k)^2 < 1/821k^2$; $x \in (0, 1/7k)$, let $f(x) = M(x, k)^2 - x_+^2 = M(x, k)^2 - x^2$, $t = 7kx \in (0, 1)$, by transformation, we are able to acquire $f(t) = -(1/49)t^3 + (1/28)t^2 + (1/14)t + (1/14)^2 - (1/49)t^2$; under $t \in (0, 1)$, the maximum value of $f(t)$ is $t = 0.1705$; plug that value into $f(t)$, we are able to acquire $f(0.1705) = 0.0012/k^2 < 1/821k^2$. Therefore, for any given $x \in R$, $M(x, k)^2 - x_+^2 \leq 1/821k^2$; the theorem is proved

The smooth function in this paper has a good degree of approximation under the same k value. The paper compared the result of functions from formula (5), (6), and (7) on the approximation effect over the nondifferentiable function. The comparison result is as in Figure 3. Figure 3 shows that the smooth function proposed by this paper has a better degree of approximation.

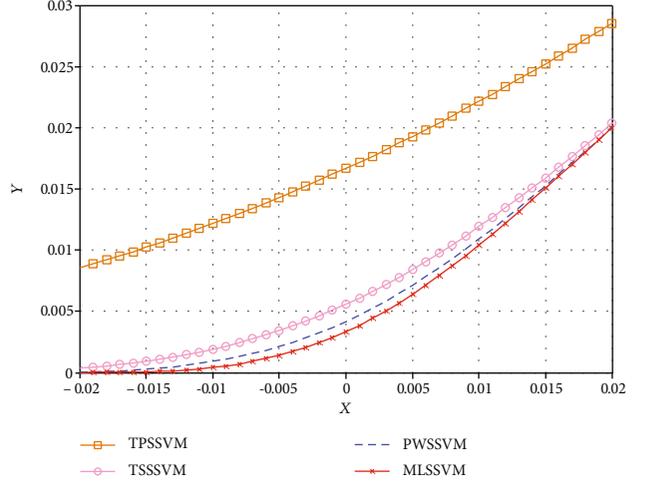


FIGURE 3: Approximation accuracy of four smooth functions.

In order to further improve the computing efficiency of the SSVM algorithm, the chaotic particle swarm algorithm with good optimization characteristics has been introduced for the optimization of parameters over penalty coefficient, insensitive parameters, and relaxation variable [22].

Chaotic characteristic itself is a pattern and possesses the property of pseudorandomness. The paper will take advantage of the two characteristics to track any state without repetition. The paper applies the logistic equation to build a chaotic optimization sequence, which is as formula (10) [23, 24].

$$x(t+1) = \mu x(t)(1-x(t)), \quad t = 0, 1, 2, \dots n. \quad (10)$$

In formula (10), μ is the overall control parameter.

When $0 < x(0) < 1$, $\mu = 4$, formula (10) is in a complete chaotic state [23, 24].

This paper applies two characteristics of chaos to initialize the position and velocity of particles in the system, which is pseudorandomness of chaos theory and its own law to enhance search capability for the swarm. Assume formulas hold true as follows:

$$\omega(i+1) = 4.0\omega(i) \cdot (1-\omega(i)), \quad (11)$$

$$\omega(i) = \alpha + (\beta - \alpha) \cdot \omega(i). \quad (12)$$

In formula (12), when $\alpha = 0.4$, $\beta = 0.9$, the system is in a perfect state of chaos. Two constants r_1, r_2 have been introduced to update the logistic mapping, which is as follows:

$$r_i(t+1) = 4.0r_i(t) \cdot (1-r_i(t)), \quad (13)$$

where $r_i(t) \in (0, 1)$, $i = 1, 2$.

Assume objective function is as follows:

$$\begin{aligned} & \min f(x_1, x_2, \dots, x_n) \\ & \text{s.t. } a_i \leq x_i \leq b_i, \quad i = 1, 2, \dots, n. \end{aligned} \quad (14)$$

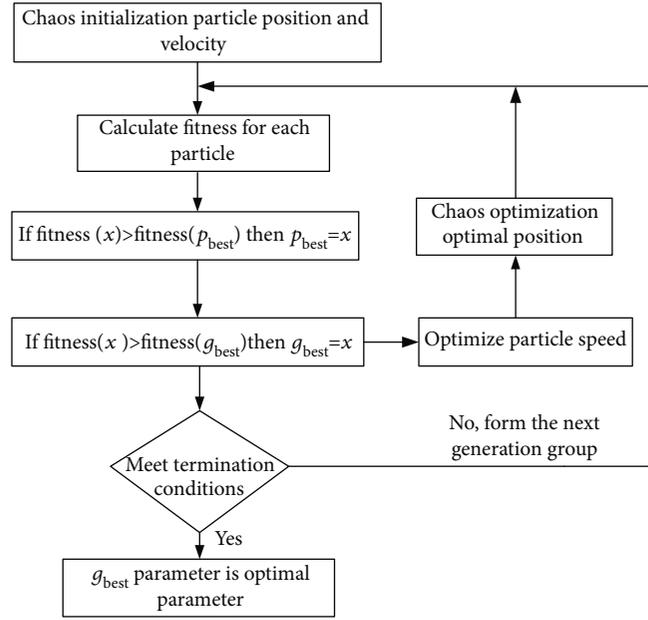


FIGURE 4: Diagram of the particle swarm algorithm.

The optimized process for the particle swarm algorithm is shown in Figure 4.

The specific procedures of the adaptive optimization algorithm are as follows [23, 24]:

- (1) Chaos initialization of corresponding parameters of a particle swarm algorithm
- (2) Comparison and optimization of the fitness level obtained from step (1)
- (3) Comparison between optimal fitness p_{best_t} within the swarm and g_{best_t} of all the particles and acquire the particle's optimal position and state
- (4) Update the particle's position and velocity
- (5) Chaotic optimization of the optimal position
- (6) In the original solution space, obtain a feasible solution $P_g^{(m)}$, calculate the level of fitness, obtain the optimal feasible solution p^* , and replace other particle positions
- (7) Through the operations of steps (1)-(7), and satisfying the set optimization conditions, the search is stopped, the optimal solution is given, and the best position is obtained, otherwise, return to step (2) and repeat the operation

3.4. Framework for the CPSO/SSVM-Based Urban Traffic Flow Prediction Model. The SVM prediction method is applied to predict multisource urban traffic flow data. It inherits the relevant ideas of machine learning. Through continuous training and learning of the prediction model, the

goal of effective prediction is finally achieved. The process mainly includes two parts, the training process and the testing process. Figure 5 shows the basic framework of the urban traffic flow prediction model.

In this paper, multisource urban traffic flow data is used as a model input, which needs to go through five stages: data collection, data preprocessing, data normalization processing, SSVM construction, and optimization problem solving. Among them, the preprocessed data consisted of training data and test data. The training data set is used to train the CPSO/SSVM model, and then, the test data set is used to test the performance of the established prediction model. The model's performance is improved by constant learning and adjusting, which will eventually lead to an automatic prediction of urban traffic flow. The execution steps of the urban traffic flow prediction model based on CPSO/SSVM are as follows.

- (1) Data collection stage: collect traffic flow data, weather condition data, and POI (points of interest) data from various sources
- (2) Data preprocessing stage: the collected multisource urban intersection traffic flow data went through data cleaning and preprocessing procedures, considering the universality of the algorithm application scenario. Firstly, preprocessed all 9,577,708 pieces of traffic information from 717 intersections on a Python 3.8 platform, all the invalid records have been removed, and a descriptive statistical analysis was carried out to filter out key urban intersections with higher average flow rate, ones with a higher average flow rate and a larger number of surrounding POIs is used as the model input

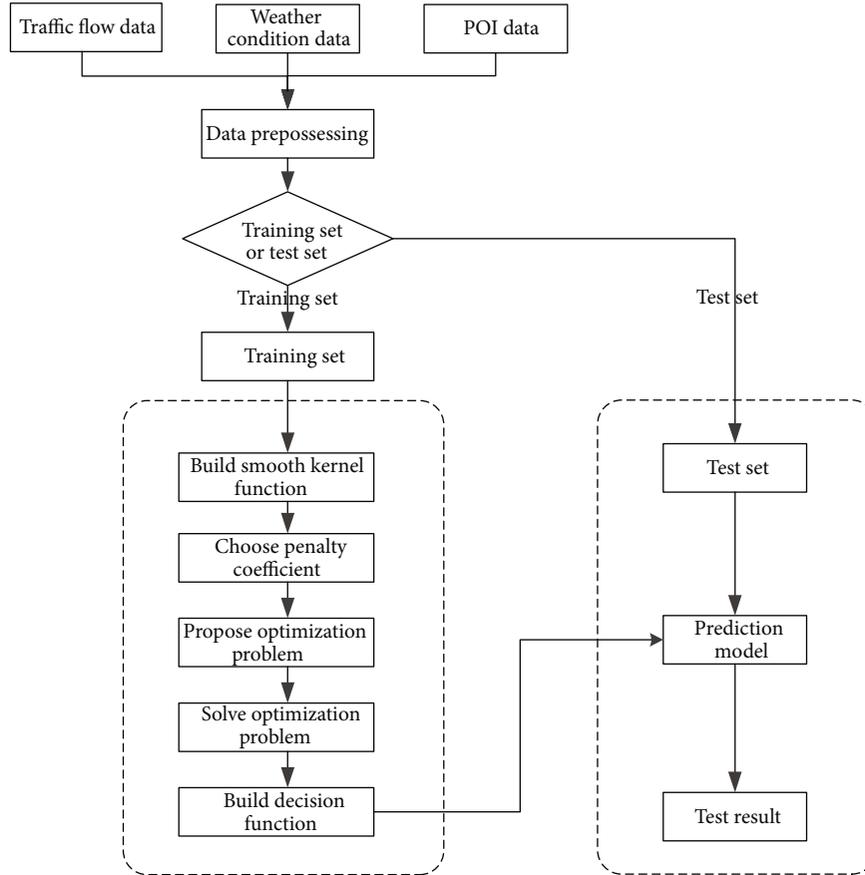


FIGURE 5: Framework for the CPSO/SSVM-based urban traffic flow prediction model.

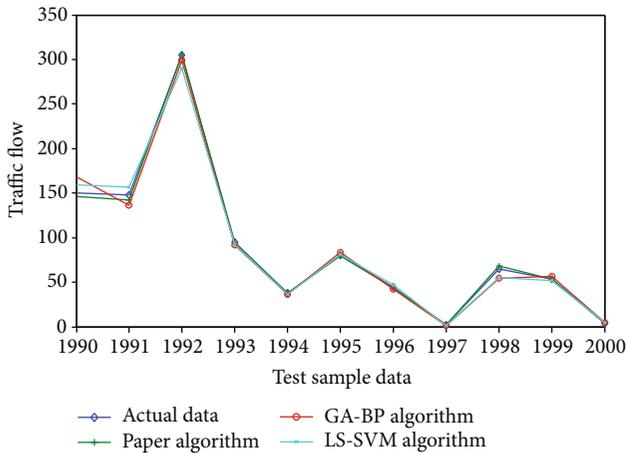


FIGURE 6: Analysis of prediction results.

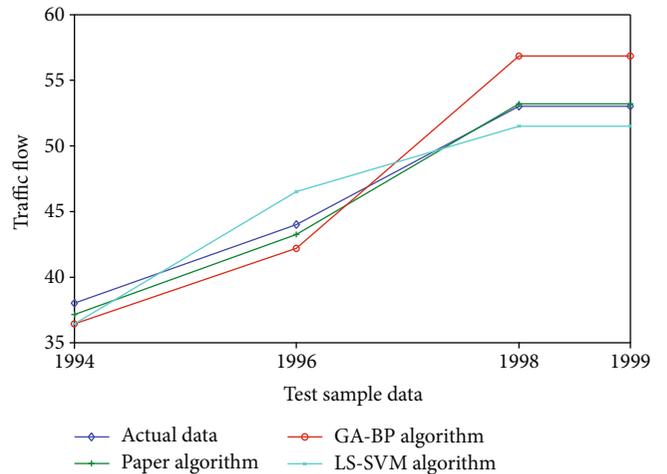


FIGURE 7: Comparison of sample data under the same dimension.

- (3) Data normalization processing: normalize the multi-source urban traffic flow data, including quantifying the collected POI information of the city intersection and the weather information of the day and apply the normalization algorithm to process all model variables to form a unified metric
- (4) Construct a smooth support vector machine: construct a smooth support vector machine algorithm model

- (5) Optimization problem solving: construct a second-order smooth kernel function and solve the optimization problem with the SSVM algorithm model to generated prediction results

4. Results and Discussion

Based on the Matlab_R2014a platform, this paper has built up a traffic flow prediction algorithm by applying the

TABLE 2: Relative error of prediction results of three algorithms.

Algorithm	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
GA-BP	11.91	7.97	1.92	3.95	4.11	4.06	4.14	24.50	16.75	7.28	5.25
LS-SVM	5.95	6.00	4.88	2.73	4.18	1.49	5.70	19.00	15.38	2.89	9.75
MLSSVM	2.59	3.95	0.22	0.95	2.32	1.11	1.77	2.00	4.85	0.32	0.25

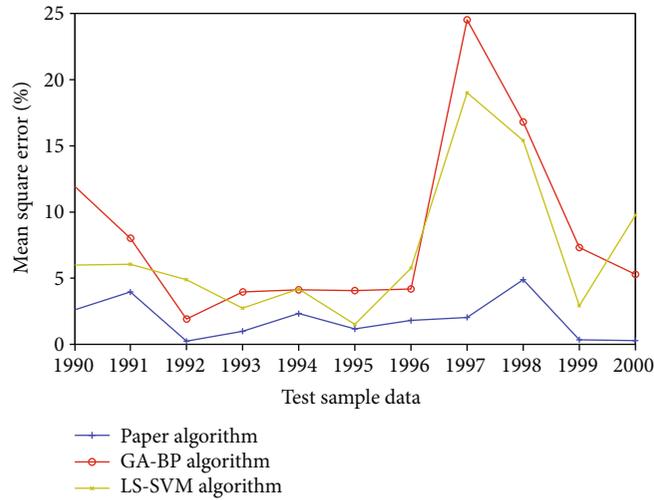


FIGURE 8: Analysis of relative error.

TABLE 3: Comparison of three algorithms time overhead.

Algorithm	No. of sample (500) time overhead (S)	No. of samples (1000) time overhead (S)	No. of samples (2000) time overhead (S)
GA-BP	0.0158	0.0276	0.0394
LS-SVM	0.0162	0.0287	0.0299
Paper algorithm	0.0147	0.0196	0.0255

optimized parameter results from part 3 and uses particle swarm optimization-smooth support vector regression to predict traffic flow. The experimental data set is the cross-section flow data of Guiyang City, Guizhou Province (5 min interval). 200 intersections with high average traffic flow were selected with 10 flow records per intersection, which is a total of 2000 flow records for model testing. Among them, 1989 records are the training set, and the last 11 records are the test set. In order to verify the prediction effect of the algorithm in this paper, this paper uses a genetic-BP neural network [25] and LS-SVM algorithm [26] for comparative analysis. The specific results are shown in Figure 6.

It can be concluded from Figure 6 that the proposed algorithm has higher prediction accuracy than other algorithms. Due to the large span of traffic flow in Figure 6, 1994th, 1996th, 1998th, and 1999th sample data have selected for comparison under the same dimension. The comparison result is as in Figure 7.

From Figure 7, it can be concluded that proposed algorithm is the best match with actual data. GA-BP algorithm have shown large deviation at 1998th and 1999th sample data, LS-SVM algorithm has shown large deviation at 1996th sam-

ple data. Therefore, the algorithm put forward by the paper has a better performance in prediction.

In order to facilitate the analysis, the relative error is introduced for analysis. The relative error data is shown in Table 2, its comparative analysis is shown in Figure 7.

As can be concluded from Figure 8 and Table 2, compared with the genetic-BP neural network and the minimum support vector machine regression algorithm, the relative error of this algorithm is lower, the prediction error is within 5%, and the accuracy and stability of the algorithm both meet the forecasting needs of actual traffic flow. Among three algorithms predicting fluctuating traffic flow, the genetic BP neural network still demonstrates the issue of overfitting, which leads to a higher prediction error. The minimum SVM performs well for the less fluctuated data but has a weak generalization ability when dealing with fluctuating traffic with high intensity. In comparison, CPSO/SSVM proposed in the paper has shown a stronger level of robustness and generalization ability.

In order to further analyze the characteristics of these three algorithms, the time cost of the three algorithms under different sample data in the prediction process is counted. The specific results are shown in Table 3.

It can be known from Table 3 that although all three algorithms can achieve traffic flow prediction under low time overhead, the algorithm in this paper has faster processing speed and higher adaptability. Therefore, based on the comprehensive analysis of the experimental results and the theoretical basis, the algorithm in this paper has a good prediction effect.

In conclusion, the SSVM algorithm put forward in this paper has better prediction accuracy in the area of traffic flow management and possesses better robustness and rapid adaptability. The algorithm can meet the requirements of low latency during the processing of heterogeneous data at the edge side, which can benefit prospective research that combines edge computing and big data analytics.

5. Conclusions

In this paper, a CPSO/SSVM model is constructed to predict traffic flow at the intersection of Guiyang City. The CPSO/SSVM model achieves better approximation and regression effects by constructing a new second-order smooth function, and at the same time, further improves the computational efficiency of the SSVM regression algorithm through particle swarm optimization. Based on experimental results, it is proved that CPSO/SSVM model is able to output more accurate result compared with the GA-BP algorithm and LS-SVM algorithm. The model has powerful information processing and prediction capabilities and can be applied to deal with complex nonlinear problems, especially the problem of traffic flow prediction at urban intersection, the location of which normally comes with complex scenes and various disturbance factors. The model provides an alternative solution for the research of data-driven urban traffic flow forecasting, and extends the application of SVM algorithm in the area of short-term urban traffic flow prediction at the same time. The output accuracy of the model is high and can be deployed in ITS to achieve short-term traffic flow prediction, which has a high application value for smart city development and real-time traffic management in edge computing scenarios.

Data Availability

The paper is based on traffic flow data from Guiyang City, with a spatial span of 717 intersections and a temporal span of 6 months. The experimental data set is the cross-section flow data of Guiyang City, Guizhou Province (5 min interval).

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The research is part of the author's employment to explore potential applications of big data analytics in smart city development and urban traffic planning. The employer is Zhejiang University.

References

- [1] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, pp. 1028–1031, Ningbo, China, September 2011.
- [2] Z. Khan, A. Anjum, K. Soomro, and M. A. Tahir, "Towards cloud based big data analytics for smart future cities," *Journal of Cloud Computing*, vol. 4, no. 1, p. 2, 2015.
- [3] F. Lin, Y. Zhou, X. An, I. You, and K.-K. R. Choo, "Fair resource allocation in an intrusion-detection system for edge computing: ensuring the security of internet of things devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 45–50, 2018.
- [4] H. Hui, C. Zhou, S. Xu, and F. Lin, "A novel secure data transmission scheme in industrial internet of things," *China Communications*, vol. 17, no. 1, pp. 73–88, 2020.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [6] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in the internet of things," *IEEE Internet of Things Journal*, p. 1, 2020.
- [7] L. Yang and B. Liu, "Temporal data fusion at the edge," in *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*, Shenyang, China, October 2019.
- [8] A. Ndikumana, N. H. Tran, T. M. Ho et al., "Joint communication, computation, caching, and control in big data multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1359–1374, 2020.
- [9] X. Xu, Q. Liu, Y. Luo et al., "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.
- [10] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: big data sharing and processing in collaborative edge environment," in *2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 20–25, Washington, DC, USA, October 2016.
- [11] B. P. L. Lau, S. H. Marakkalage, Y. Zhou et al., "A survey of data fusion in smart city applications," *Information Fusion*, vol. 52, pp. 357–374, 2019.
- [12] A. J. Jara, D. Genoud, and Y. Bocchi, "Short paper: sensors data fusion for smart cities with KNIME: a real experience in the smartsantander testbed," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, Seoul, South Korea, March 2014.
- [13] X. Jiang and H. Adeli, "Dynamic wavelet neural network model for traffic flow forecasting," *Journal of Transportation Engineering*, vol. 131, no. 10, pp. 771–779, 2005.
- [14] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.
- [15] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2020.
- [16] C. Yuan, D. Li, and Y. Xi, "Short term urban traffic mode prediction based on VOMM," in *2016 Chinese Control and Decision Conference (CCDC)*, Yinchuan, China, May 2016.

- [17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [18] Y.-J. Lee, W.-F. Hsieh, and C.-M. Huang, "Epsilon-SSVR: A smooth support vector machine for epsilon insensitive regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 678–685, 2005.
- [19] Y. B. Yuan, J. Yan, and C. X. Xu, "Polynomial smooth support vector machine," *Chinese Journal of Computers*, vol. 28, no. 1, pp. 9–17, 2005.
- [20] Q. Wu, "A new class of ε -piecewise smooth support vector regressions," *Journal of Information Science & Engineering*, vol. 31, no. 5, pp. 1813–1828, 2015.
- [21] W. U. Qing, L. Bo, W. Wan, and W. En, "Face recognition method based on multi-class classification of smooth support vector machine," *Journal of Computer Applications*, vol. 19, no. 5, pp. 688–696, 2015.
- [22] M. Clerc, *Particle swarm optimization*, John Wiley & Sons, New Jersey, 2010.
- [23] F.-P. An, "Image feature extraction algorithm based on bi-dimensional local mean decomposition," *Optical Review*, vol. 26, no. 1, pp. 43–64, 2019.
- [24] F.-P. An and Z.-W. Liu, "Bi-dimensional empirical mode decomposition (BEMD) algorithm based on particle swarm optimization-fractal interpolation," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 17239–17264, 2019.
- [25] G. D. Sun, L. A. Qin, Z. H. Hou et al., "Feasibility analysis for acquiring visibility based on lidar signal using genetic algorithm-optimized back propagation algorithm," *Chinese Physics B*, vol. 28, no. 2, article 024213, 2019.
- [26] J. Zhu, "Probability least squares support vector machine with L1 norm for remote sensing image retrieval," *International Journal of Information and Communication Technology*, vol. 13, no. 2, p. 208, 2018.