sensors

MDPI

Article

Data Prediction of Mobile Network Traffic in Public Scenes by SOS-vSVR Method

Xiaoliang Zheng [1,2], Wenhao Lai [2,*], Hualiang Chen [2] and Shen Fang [3]

[1] State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines, Anhui University of Science and Technology, Huainan 232000, China; zhengxl@aust.edu.cn

[2] School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232000, China; hualiang_chen520@163.com

[3] Huainan Branch of China Mobile Group Anhui Company Limited, Huainan 232000, China; fangshen@ah.chinamobile.com

* Correspondence: whlai9@163.com

Received: 23 December 2019; Accepted: 20 January 2020; Published: 22 January 2020

check for updates

Abstract: Accurate base station traffic data in a public place with large changes in the amount of people could help predict the occurrence of network congestion, which would allow us to effectively allocate network resources. This is of great significance for festival network support, routine maintenance, and resource scheduling. However, there are a few related reports on base station traffic prediction, especially base station traffic prediction in public scenes with fluctuations in people flow. This study proposes a public scene traffic data prediction method, which is based on a v Support Vector Regression (vSVR) algorithm. To achieve optimal prediction of traffic, a symbiotic organisms search (SOS) was adopted to optimize the vSVR parameters. Meanwhile, the optimal input time step was determined through a large number of experiments. Experimental data was obtained at the base station of Huainan Wanda Plaza, in the Anhui province of China, for three months, with the granularity being one hour. To verify the predictive performance of vSVR, the classic regression algorithm extreme learning machine (ELM) and variational Bayesian Linear Regression (vBLR) were used. Their optimal prediction results were compared with vSVR predictions. Experimental results show that the prediction results from SOS-vSVR were the best. Outcomes of this study could provide guidance for preventing network congestion and improving the user experience.

Keywords: public scene; mobile network traffic data prediction; vSVR; symbiotic organisms search

1. Introduction

Mobile Internet not only enriches people's entertainment life, but also brings great convenience to daily life. The relationship between our life and mobile network has become closer. By the end of March 2019, the number of 4G users in China was 1.204 billion. The per capita traffic was 7.27 GB per month [1]. Also, the use number and per capita traffic maintain a growing trend. Rapidly growing network size and user demands present many challenges to current network infrastructure. For some multifarious personnel flow and densely populated scenes (high-speed rail stations, tourist attractions, commercial centers, playgrounds, sports competitions, concert venues, etc.), the rapidly growing traffic puts tremendous pressure on its network architecture. Burst traffic data of a wireless network may cause the failure of the normal function of the base station equipment. When that problem is not severe, users' internet access speed just becomes slow; however, in extreme cases, users cannot access the internet and the mobile phone connection rate drops sharply. People often try to connect to network under such failure circumstances, which will bring secondary shocks to wireless networks and core networks, e.g., at around 11:08 on 29 May 2019, there was a brief interruption in 4G internet access of batch users in municipalities of China. Then some users re-registered, causing instantaneously high load work to seven sets of switches. Some mobile phones under switches could not access the internet. The number of users affected by this network accident exceeded one million.

In daily life, traffic-change prediction based on hourly granularity and guidance for on-demand allocation of network resources could reduce network operation costs. In festivals or large-scale activities, the user size and mobile traffic of public scenes could be predicted quickly and accurately based on people flow and the tidal effect of traffic. This prediction could help operators master upcoming congestion and make network expansion, adjustment, and optimization in advance, and use limited wireless carrier resources to meet network peaks. Emergency plan for a network schedule could developed to give access to users that does not degrade in case of sudden increased traffic. Under the impact of large-scale network congestion and blocking, traffic and routing could be scheduled in time to ensure the network access rate and network connection rate of essential areas and to retain connected customers. Therefore, the prediction of traffic data in a base station for multifarious personnel flow and densely populated public scenes is of considerable significance in network security.

At present, there are related reports on base station traffic prediction [2–6]. However, research on traffic prediction for 4G base stations in public scenes has seldom been reported. For example, the authors of [2] extracted and modeled traffic patterns of 9000 cellular towers deployed in a metropolis, and evaluated a big data-driven method for predicting mobile network traffic data. However, this method was only for traffic prediction of large-scale cellular networks. In [3], an end-to-end deep Traffic Predictor (TP) based on deep learning was proposed. This realized traffic demand prediction based on data in Shanghai and predicted base station traffic data from third-generation communication technology (3G). A deep TP requires massive training data, has higher requirements for hardware. Besides, and deep models are more difficult to train. Qiu et al. [4] proposed a wavelet-based stacked denoising autoencoder (Wavelet-SDA) deep learning framework to predict the number of users of mobile base stations; however, it was not used to predict traffic data. Therefore, this paper proposes a traffic data prediction model for 4G networks based on machine learning in public places, which is used for traffic prediction of public scenes with large changes in people flow. This study aims to provide guidance for operators' network security and maintenance.

Traffic prediction is a typical time series prediction problem, also known as a regression problem in machine learning [7]. Commonly used regression algorithms are the autoregressive integrated moving average model (ARIMA) [8], Bayesian linear regression (BLR) [9], extreme learning machine (ELM) [10], support vector regression (SVR) [11], artificial neural network (ANN) [12–15], et al. SVR was developed by Vapnik's theory of the support vector machine [16]. From a theoretical point of view, the principle of structural risk minimization (SRM) in SVM makes it have better generalization power than the principle of empirical risk minimization (ERM) used in neural networks [17]. SVR has great potential and excellent performance in practical applications. For example, the wind speed prediction of SVR wind farms is reported in [18]. Results showed that comparing with the least-squares method for autoregressive model adjustment, SVR has a higher accuracy for wind energy technology (WET) prediction. Kaytez et al. [19] comprehensively compared neural networks with the least squares support vector machine in predicting electricity consumption in Turkey. Analysis results showed that the Least Squares-Support Vector Machines (LS-SVM) model can be effectively used for long-term net power consumption predictions in Turkey. Chen et al. [20] used seasonal index adjustment SVR for vacation forecasting of daily tourist traffic. Results showed that SVR had a higher accuracy and less errors than the Back Propagation Neural Network (BPNN) model. In addition, SVR was successfully applied to a toxicity assessment [21], battery life prediction [22,23], chemical production [24,25], financial projection [26,27], and agricultural production [28,29]. Based on excellent regression performance and a wide range of successful application cases of SVR, it was selected to predict the traffic data of the 4G network base stations in public scenes in this study.

Symbiotic organisms search (SOS), a new metaheuristic optimization algorithm proposed by Cheng et al. [30], which was inspired by symbiotic relationship between organisms in the ecosystem.

A significant advantage of SOS is that, it does not have any specific adjusting parameters, avoiding risk of incorrect parameter adjustments which may affect result accuracy. The algorithm implementation is simple and stable. SOS optimization has been tested on selected benchmark functions. Results were compared with many recently developed effective optimization techniques, such as genetic algorithm (GA), bees algorithm (BA), particle swarm optimization (PSO), and differential evolution (DE). Results showed that SOS had a better performance than other metaheuristic technologies [31]. As a result, the SOS algorithm has received extensive attention in the optimization research community and other related fields [32]. Ezugwu et al. [33] used SOS for the optimization of travelling salesmen's problems and the discrete search performance of SOS was further verified. Cheng et al. [34] allocated multiple resources in the process of SOS optimization of multiple project-planning problems (DSOS-MRLMP). Results indicated that the SOS model could obtain more reliable and effective results. Verma et al. [35] used SOS for congestion management in a deregulated environment and Çelik et al. [36] used SOS for the design of the Proportional Integral Derivative (PID) controller.

Although SVR has excellent optimization performance, it is as sensitive to parameter settings as other machine learning algorithms. Manual determination of optimal parameters of SVR is time-consuming and laborious. Based on the excellent optimization performance of SOS, we used the symbiotic organisms search optimization algorithm for parameter optimization of SVR, which was used for traffic data prediction of 4G network base stations in public scenes. In this study, we not only used ELM and vocational Bayesian Linear Regression (vBLR) for traffic prediction, but also adopted moth-flame optimization (MFO) [37] and PSO for optimization. Finally, the results were used for optimization algorithm comparison and performance verification of SOS-SVR.

2. Materials and Methods

2.1. Methods

2.1.1. v Support Vector Regression Algorithm

Support vector regression is an application of a support vector machine in a regression task. Support vector machine (SVM) [16,38] is a classical algorithm that is driven from results of statistical learning theory. It was originally developed for classification tasks in the field of pattern recognition and represented decision boundaries of a typically small subset of all training examples called support vectors. The core task of SVM is to find the hyperplane with the largest interval, then divide the samples into different categories.

SVM and SVR function is described as: $f(x) = w \cdot \Phi(x) + b$, where b is the bias and w is weight vector. Φ is nonlinear mapping, x is mapped into a feature space in which a linear estimate function is defined. $x \in R^N$, R^N is represents input space.

Given a data (x_i, y_i), x_i and $y_i \in R$, the constrained form is:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 \\ &s.t.\ y_i \cdot \left(W^T x_i + b\right) \geq 1,\ \forall i \end{aligned} \tag{1}$$

where, W is an element in the high-dimensional spaces, b is the threshold, $\|w\|^2$ is the description function.

The core of SVR is to find a hyperplane to minimize structural risks. To reduce the effect of noise, a slack variable ξ is introduced. Thus, the constraint becomes:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\left(\xi_i + \hat{\xi}_i\right) \\ &s.t.\ \left\{ \begin{array}{l} y_i - \left(W^T x_i + b\right) \leq \varepsilon + \xi_i \\ \left(W^T x_i + b\right) - y_i \geq \varepsilon + \hat{\xi}_i \end{array} \right. ,\ \xi_i, \hat{\xi}_i \geq 0, w \in W \end{aligned} \tag{2}$$

where, C is a constant, ε is insensitive loss function.

Equation (2) is a constrained optimization problem. When a Lagrange function is introduced, and a dual form is obtained, the maximization function is as follows:

$$\begin{aligned}&W(\alpha,\hat{\alpha}) = \sum_i^m(\alpha_i-\hat{\alpha}_i)y_i - \varepsilon\sum_i^m(\alpha_i+\hat{\alpha}_i) - \tfrac{1}{2}\sum_i^m(\alpha_i-\hat{\alpha}_i)\left(\alpha_j-\hat{\alpha}_j\right)\left(x_i\cdot x_j\right)\\ &s.t.\ \begin{cases}\sum\limits_i^m(\alpha_i-\hat{\alpha}_i)=0\\ 0\le\alpha_i,\ \alpha_i^*\le C\end{cases}\end{aligned} \tag{3}$$

where, α is a Lagrangian operator.

The regression estimate obtained by Equation (3) is linear. In order to adapt the support vector regression to a nonlinear problem, an input vector is mapped into a high-dimensional feature space (in Equation (4)), then is re-performed in this space. Linear regression is used to obtain effect of nonlinear regression in original space.

$$x_i\cdot x_j\to\Phi(x_i)\cdot\Phi\left(x_j\right) \tag{4}$$

where, $\Phi(\cdot)$ is a mapping function that maps x from low-dimensional space to high-dimensional space, which is also called a kernel function.

When solving ε Support Vector Regression (εSVR), the parameter ε plays a decisive role in fitting performance of objective function. However, it is difficult to determine the reasonable value. In order to determine the reasonable value of ε more conveniently, a new parameter v denoted as vSVR is introduced to control number of support vectors and training error. Therefore, its constraints change accordingly:

$$\begin{aligned}&\min\tfrac{1}{2}\|w\|^2+C\left(v\varepsilon+\sum_{i=1}^m\left(\xi_i+\hat{\xi}_i\right)\right)\\ &s.t.\begin{cases}y_i-\left(W^Tx_i+b\right)\le\varepsilon+\xi_i\\ \left(W^Tx_i+b\right)-y_i\ge\varepsilon+\hat{\xi}_i\end{cases},\ \xi_i,\hat{\xi}_i\ge0\end{aligned} \tag{5}$$

where, the value range of v is [0, 1]. The dual problem of Lagrange function of vSVR constraint is shown in Equation (6):

$$\begin{aligned}&W(\alpha,\hat{\alpha})=\sum_i^m(\alpha_i-\hat{\alpha}_i)y_i-\tfrac{1}{2}\sum_i^m(\alpha_i-\hat{\alpha}_i)\left(\alpha_j-\hat{\alpha}_j\right)k\left(x_i,x_j\right)\\ &s.t.\ \begin{cases}\sum\limits_i^m(\alpha_i-\hat{\alpha}_i)=0,\ \sum\limits_i^m(\alpha_i+\hat{\alpha}_i)\le Cv\\ 0\le\alpha_i,\ \alpha_i^*\le C\end{cases}\end{aligned} \tag{6}$$

where, $k\left(x_i,x_j\right)$ is the mapping of input vectors to high-dimensional space. The kernel function is a Gaussian kernel function. By solving above optimization problem, we could finally get the output function of support vector machine:

$$f(x)=\sum_i^m(\alpha_i-\hat{\alpha}_i)\cdot k\left(x_i,x_j\right)+b \tag{7}$$

2.1.1.1. Symbiotic Organisms Search

Symbiotic organisms search [30] is a new meta-heuristic swarm intelligence optimization algorithm proposed by Cheng et al. and was inspired from the symbiotic interactions observed between two organisms in the ecosystem.

(1). Symbiosis

In 1869, German mycologist de Bary first used this word, symbiosis, to define the relationship between two different species of organisms that are interdependent. Symbiotic relationships are broadly divided into two types, such as obligate and facultative. In an obligate relationship, both organisms entirely depend on each other for their survival, whereas in a facultative relationship, the organisms may depend on each other, but it is not mandatory. Three types of symbiotic relationships are found in nature. These are mutualism, commensalism, and parasitism. Mutualism refers to the relationship between two different species of organisms where both individuals are benefited. Commensalism describes the symbiotic relationship between two organisms in which one benefits and the other is, not significantly, affected. Parasitism is the kind of symbiotic relationship where one organism is benefited and the other is, effectively, harmed. Living organisms undergo symbiotic relationships in order to adapt themselves in the environment and, hence, they improve their fitness to survive in the ecosystem over the long term.

(2). Features

SOS algorithms commence with an initial population of organisms which is called the ecosystem. Almost all metaheuristic algorithms apply a succession of operations to solutions in each iteration in order to generate new solutions for the next iteration. In SOS, new solution generation is governed by imitating the biological interaction between two organisms in the ecosystem. Each organism of the ecosystem is considered as a candidate solution to the corresponding problem and is correlated to a certain fitness value which imitates the degree of adaptation to the desired objective. The new solutions are generated by simulating the symbiotic interactions between two organisms in the ecosystem, which includes the mutualism, commensalism and parasitism phases. Each organism in the ecosystem randomly interacts with the other through all these three phases and this process of interaction is repeated until the termination criterion is fulfilled.

(3). Mutualism phase

In this phase, the relationship established by the mutually beneficial symbiotic order can benefit both sides of the organism. A similar example is a relationship between bees and flowers; the organism renewal formula is as shown in Equation (8):

$$\begin{cases} x_{inew} = x_i + r_1 \cdot (x_{best} - \varphi \cdot BF_1) \\ x_{jnew} = x_j + r_2 \cdot (x_{best} - \varphi \cdot BF_2) \\ \varphi = 0.5\big(x_i + x_j\big) \end{cases} \tag{8}$$

where, r_1 and r_2 are rand number in [0, 1], x_i and x_j are two organisms that have a symbiotic relationship, φ is the relationship characteristics between the two organisms. BF represents the level of benefit of the two organisms, recorded as the benefit factor. In some symbiotic relationships, the benefits obtained by the two organisms are different.

1. Commensalism Phase

In this phase, the relationship established during the commensal phase benefits only one organism, while the other is harmless or has little interest. A similar example is the relationship between crucians and a shark. The organism at this stage is shown:

$$x_{inew} = x_i + r_3 \cdot (x_{best} - x_j) \tag{9}$$

where, r_3 is a random number in [0, 1], x_i is the party that benefits more from the symbiotic relationship and x_j is the party that is harmless in this relationship.

2. Parasitism Phase

In this phase, one organism gets benefits and the other organism is damaged. A similar example is the relationship between malarial mosquitoes and humans. In the SOS algorithm, this stage forms an individual called "Parasite_Vector" by duplicating the organism x_i, then modifying the randomly selected dimensions using a random number. The Organism x_j is selected randomly from the ecosystem and serves as a host to parasite vector. Parasite_Vector tries to replace x_j in the ecosystem. Both organisms are then evaluated to measure their fitness. If Parasite_Vector has a better fitness value, it will kill the organism x_j and assume its position in the ecosystem. If the fitness value of x_j is better, x_j will have immunity from the parasite. Parasite_Vector will no longer be able to live in that ecosystem. The optimization process of SOS is as follows:

Step 1: Define and initialize the ecosystem population, and initialize all variables or parameters;
Step 2: Find the best individual x_{best} in the population;
Step 3: x_i and x_j are selected randomly. Calculate the modified organism according to Equation (8), calculate the fitness value, and update the individual information;
Step 4: x_j is selected randomly. Calculate the modified organism according to Equation (9), calculate the fitness value, and update the individual information;
Step 5: x_i and x_j are selected randomly. Modify the values of certain dimensions of organism i, creating a Parasite_Vector. Calculate the fitness value, compare it with the organism j, and choose to update or retain;
Step 6: If the current creature is not the last creature in the ecosystem, go back to step 3;
Step 7: Identify the end condition of the search. If no, go back to step 2; If yes, output the optimal results.

2.1.1.2. SOS-vSVR Algorithm

If the parameters can be set reasonably, v-SVR will have excellent performance in regression tasks. These parameters include the regularization coefficient C in Equation (6) and number parameter v for controlling support vector. In addition, the kernel function used in this study is Gaussian kernel function (in Equation (10)), and the parameters g also need reasonable settings.

$$k\left(x_i, x_j\right) = \exp(-g\left\|x_i, x_j\right\|^2) \tag{10}$$

Manually setting of parameters c, g, and v requires lots of experimentation, which is time-consuming and difficult to find the optimal value. Therefore, SOS is used for the optimization of vSVR parameters as follows:

In Figure 1, *Num_org*, *c_Iter*, *num_fit*, *max_Iter*, and *maxnum_fit* are the number of creatures, the current number of iterations, the number of fitness calculations in optimization process, the maximum number of iterations, and the maximum number of calculations in optimization process, respectively. Root mean square error (RMSE) was used to calculate the fitness value, as shown in Equation (11):

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}\left|y_i - \hat{y}\right|^2} \tag{11}$$

where, y_i and $\hat{y}_i$ are observations and predictions, respectively.

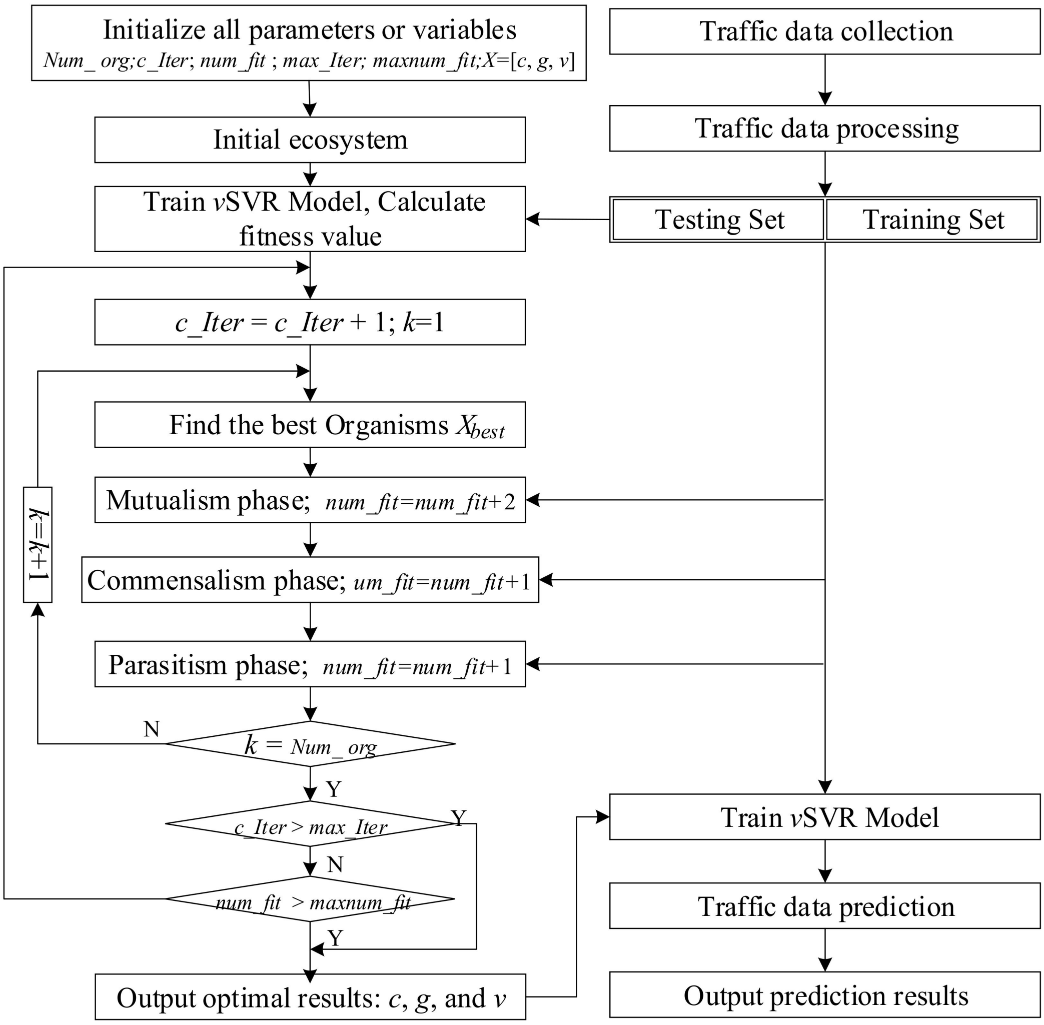

Initialize all parameters or variables
Num_ org;c_Iter; num_fit ; max_Iter; maxnum_fit;X=[c, g, v]
Initial ecosystem
Train *v*SVR Model, Calculate fitness value
c_Iter = c_Iter + 1; k=1
Find the best Organisms X_{best}
Mutualism phase; *num_fit=num_fit+2*
Commensalism phase; *um_fit=num_fit+1*
Parasitism phase; *num_fit=num_fit+1*
k=k+1
N
k = Num_ org
Y
c_Iter > max_Iter
Y
N
num_fit > maxnum_fit
Y
Output optimal results: *c*, *g*, and *v*
Traffic data collection
Traffic data processing
Testing Set
Training Set
Train *v*SVR Model
Traffic data prediction
Output prediction results

Figure 1. The flowchart of symbiotic organisms search (SOS) optimized *μ* support vector regression (*v*SVR).

2.1.2. Materials

2.1.2.1. Experimental Data Collection

In this paper, the selected public scene was Huainan Wanda Plaza. The period for obtaining traffic data was 14 April–13 July 2019, including International Labor Day (May 1) and China's traditional festival Dragon Boat Festival (June 7). The obtained traffic data is shown in Figure 2.

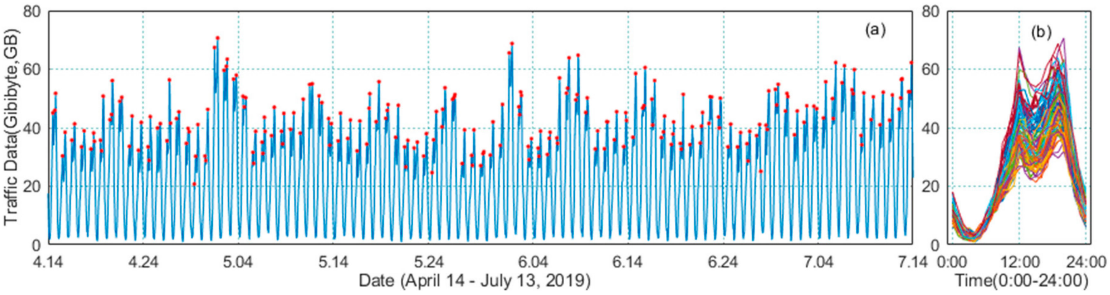


Figure 2. Base station data of Wanda Plaza. (**a**) Data curve for April 14–July 13. (**b**) Daily traffic comparison.

In Figure 2, the data collection granularity is 1 h, that is, 24 times per day. It could be seen from the figure that the daily traffic flow trend is similar (the peak value is usually the data collected at 12:00 or 20:00). However, the daily peak values vary greatly, and the festivals' traffic peaks were significantly higher than the non-holiday traffic peaks. Among the obtained traffic data, the maximum daily traffic peak was on May 1 (holiday), which was 70.67 GB, while the minimum daily peak value was on May 22 (Wednesday, weekday), which was 35.11 GB.

2.1.2.2. Data Processing

Base station data prediction is a typical time series prediction problem. Data processing for time series prediction usually has two parts, one is numerical processing (such as data normalization), and the other is sequence processing (processed into equal length input data and its corresponding output data).

$$x' = \frac{x}{\max(X)}, \; x \in X \tag{12}$$

where x' is the processed traffic data.

If we use a normalized method to process the data, we cannot directly use the prediction results to evaluate the model because there is a zero value in the normalized data y_i. In this paper, we used Equation (12) to process traffic data in numerical values. Compared to the normalized processing method, a significant advantage of Equation (12) is that it allows us to directly use prediction results to evaluate the prediction performance of the model without inverse transformation.

For sequence processing, the data is sliced into sequences of equal length, and each sequence is continuous. Equal-length sequences are used as inputs to the prediction model. The next value of each sequence is output as the training model target. The way of sequence processing of traffic data is as shown in Figure 3.

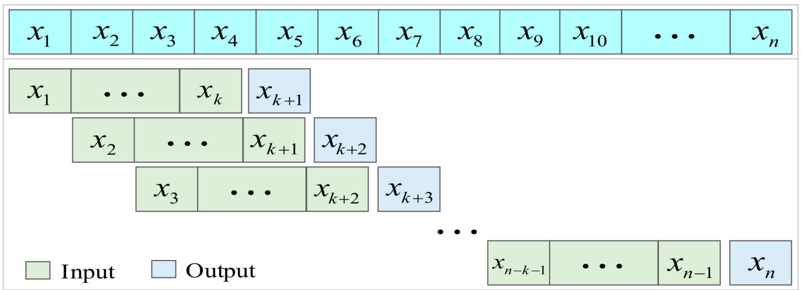

x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} . . . x_n
x_1 . . . x_k x_{k+1}
x_2 . . . x_{k+1} x_{k+2}
x_3 . . . x_{k+2} x_{k+3}
. . .
x_{n-k-1} . . . x_{n-1} x_n
Input
Output

Figure 3. Sequence processing of traffic data.

In Figure 3, n and k are the length of time series and the length of flow data of theinput prediction model, respectively. The model output length is 1. After sequence processing, the input data has n-k samples, that is, the input data dimension is (n−k) *k. In this paper, the data collected were from 14 April to 13 July 2019.24 data were collected each day, so the value of n is 2184.

The length of input data affects the vSVR different performance (k takes different values). In this paper, to get the best prediction, the optimization of k was one of our main tasks.

3. The Prediction of Traffic Data

3.1. Experimental Preparation

3.1.1. Experiments Setup

In machine learning, when the data is not extremely large, the rate of training set to the test set is generally 4:1. Therefore, the rate of training set and test set in this paper is 4:1. The traffic data from May 2 to July 13 was used for the training of the predicted model, while the remaining data was used for the model test. To verify the performance of SOS optimized vSVR, we also used the optimization algorithms MFO and PSO for parameters optimization of vSVR. Besides, to test the performance of vSVR on traffic data prediction, εSVR, ELM, and vBLR were also used for prediction. The above prediction results are used for comparison. In the experiment, the parameter settings of each optimization algorithm are shown in Table 1.

Table 1. Initial parameters of the SOS, particle swarm optimization (PSO), and moth-flame optimization (MFO).

Algorithm	Parameter	Value
SOS	*max_Iter;*	30
	maxnum_fit	100
	Number of organisms	40
PSO	Acceleration constants	[1.7; 1.5]
	Inertia w	[0.9, 0.75]
	Generations	30
	Number of particles	40
MFO	*b*	1
	Iterations	30
	Number of search agents	40
SOS/PSO /MFO-vSVR	lb_{cgv}	[0.1; 0.001; 0.01]
	ub_{cgv}	[100; 10; 1]
SOS/PSO /MFO-εSVR	lb_{cg}	[0.1; 0.001]
	ub_{cg}	[100; 10]

In Table 1, the variables ub_{cgv} and lb_{cgv} are the upper and lower boundary vectors of the parameters c, g, and v of the vSVR algorithm, respectively. The variables ub_{cg} and lb_{cg} are the boundary vectors of parameters c and g of the εSVR algorithm, respectively.

Each experiment in this paper was run many times, independently. All experiments in this work were conducted on a personal computer with Intel(R) Core (TM) i7-9700K processor, 3.60 GHz, 16GB RAM, using Windows 10 as the operating system. The LibSVM-3.23 implementation was used for the SVR regression. We also used Matlab implementation environment in our experiment.

3.1.2. Prediction Model Evaluation Method

In the parameter optimization process of prediction algorithm, the fitness function of optimization algorithm used RMSE. To more objectively compare the performance of different prediction algorithms, mean absolute percent error (MAPE) was also used to evaluate the model. The evaluation mechanism of MAPE is shown in Equation (13).

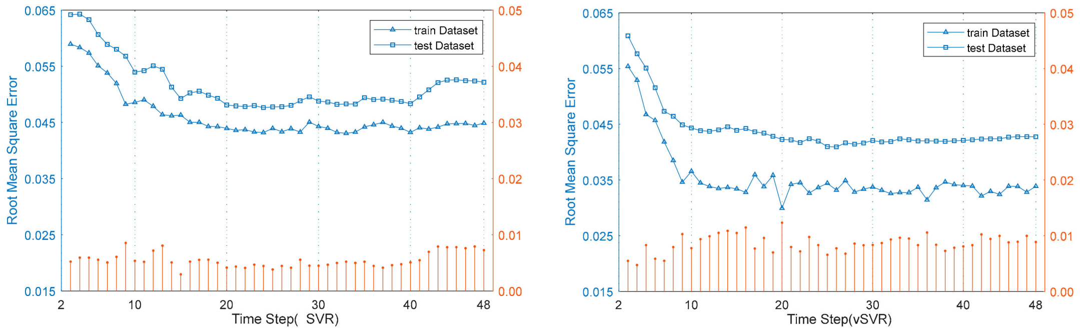
$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \quad (13)$$

where, y_i and $\hat{y}_i$ are observations and predictions, respectively.

3.2. Application of SOS-vSVR in Traffic Data Prediction

In this part of experiment, we used SOS to search for the optimal parameters of vSVR, and then used the obtained optimal parameters for traffic data prediction. To verify the optimization performance of SOS and the performance of vSVR, we not only used MFO and PSO for parameter optimization of vSVR, but also used εSVR for traffic data prediction.

In the time series prediction problem, the sequence length of model input affects the prediction performance. The optimal sequence length corresponding to different prediction models may also be different. To better obtain the best performance of SVR on traffic data prediction in a public scene, we first needed to get the optimal sequence length, that is, the optimal k value. For different values of k, the minimum RMSE of vSVR and εSVR searched by SOS are shown in Figure 4.



(**a**) (**b**)

Figure 4. Root mean square error (RMSE) of vSVR and εSVR when k gets different values. (**a**) The RMSE of vSVR; (**b**) The RMSE of εSVR.

In the experiment, each k value was repeated ten times, independently. The minimum RMSE was selected. In Figure 3, the left axis is RMSE of training set and test set, while the right axis is the subtraction of RMSE of test set and training set.

In an excellent classification or regression model, the error in training set and test set both should be small. Likewise, in an excellent classification or regression model, the test error is as small as a training error, but it is usually higher than the training error. Therefore, we were more concerned with the performance of the prediction model on the test set. It can be seen from Figure 4 that when the k values are the same, the RMSE of the vSVR is less than the RMSE of εSVR, whether it is the training error or test error. Besides, the minimum RMSE of the vSVR and εSVR on the test set decreased first, before increasing with the rise of k. The minimum RMSE of vSVR and εSVR on the test set are 0.0409 and 0.0476, respectively, and the corresponding k values are 26 and 24, respectively. Therefore, in the next part of experiment, the length of vSVR and εSVR input sequences were 26 and 24. The results of SOS, MFO, and PSO optimization of vSVR and εSVR are shown in Table 2.

Table 2. Prediction results of vSVR and εSV.

Algorithm	Time Steps (k)	Train		Test		c	g	v	Optimization Time (s)
		RMSE	MAPE	RMSE	MAPE				
SOS-vSVR	26	0.0332	9.25%	0.0409	12.36%	3.28	0.2602	0.837	298.25
MFO-vSVR	26	0.0333	9.43%	0.0408	12.19%	5.84	0.1969	0.776	1433.11
PSO-vSVR	26	0.0346	9.81%	0.0415	12.37%	57.12	0.0557	0.891	1189.03
SOS-εSVR	24	0.0432	17.38%	0.0476	20.61%	63.73	0.0692	—	2.94
MFO-εSVR	24	0.0432	17.43%	0.0475	20.54%	100.00	0.0568	—	30.37
PSO-εSVR	24	0.0432	17.30%	0.0476	20.51%	64.83	0.0675	—	23.16

Abbreviation: MAPE, mean absolute percent error.

In Table 2, each model was repeatedly optimized ten times, and the optimal results were selected. c, g, and v are the parameter values corresponding to optimal predictions of vSVR and εSVR searched by the optimization algorithm. 'Optimization time' is the average time of 10 repeated experiments.

It can be seen from Table 2 that the support vector predicted base station traffic data in the public scene. The prediction accuracy of vSVR was significantly better than εSVR, whether on the training set or the test set. SOS, MFO, and PSO were used to optimize vSVR parameters, to predict base station traffic data for public scenarios. The prediction results of SOS-vSVR on the training set were optimal, whether it was RMSE or MAPE, and the prediction result of MFO-vSVR on the test set was optimal.

In the experiment, different optimization algorithms were used to search the parameters of vSVR, the searched optimal settings were used to predict the base station traffic data of the public scene. The optimal prediction results differed in different data sets and evaluation methods. However, compared to the difference between the predicted results of εSVR and vSVR, the difference between the optimal results of the optimization algorithm search was very small. For example, the maximum difference in the MAPE of the vSVR searched by SOS, MFO, and PSO on the test set was 0.18%, and the maximum difference in MAPE between εSVR and vSVR was 8.32%.

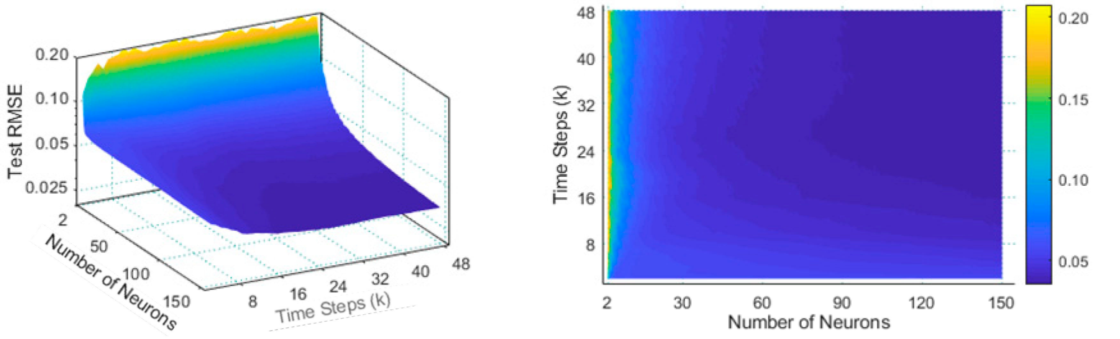
SOS, MFO, and PSO optimize vSVR to predict traffic data. In terms of accuracy, MFO and SOS had the same excellent performance. However, SOS had an absolute advantage in optimizing time-consumption, whether it was optimizing εSVR or vSVR. For example, the average time of SOS optimization εSVR was less than one-tenth of MFO, which means that SOS optimization efficiency is very high in traffic prediction.

3.3. Compared With Non-Optimal Regression Algorithms

In the previous section, we used PSO and MFO for support vector regression optimization and compared the results with SOS optimization results to verify the efficiency of SOS-vSVR in traffic prediction. In this section, extreme learning machine (ELM) and variational Bayesian linear regression (vBLR) are adopted to predict mobile traffic, which in turn validates the advancement of vSVR's predicted traffic data.

We optimized the vSVR when it predicted traffic data. For fairness, the best results of ELM and vBLR for traffic prediction should also be obtained before comparison. In this part of experiment, we first determined the optimal time step k of ELM and vBLR. Since the regression performance of ELM is closely related to the number of hidden layer nodes. It is also necessary to determine the optimal number of hidden layer nodes of ELM.

In Figure 5, the value of k is an integer in [2, 48], and the number of hidden layer nodes is an integer of [2, 150]. Since some of the ELM parameters are random, and these random parameters also affect their performance. When we looked for the optimal parameters of ELM, our number of experiments was as high as 47*149*100 times (the combination of different k and hidden layer neuron numbers was repeated 100 times independently).



(**a**) (**b**)

Figure 5. RMSE of vSVR and εSVR when k takes different values. (**a**) The RMSE of vSVR; (**b**) The RMSE of εSVR.

As can be seen from Figure 5b, when the time step was greater than 20 and the number of hidden layer nodes was greater than 90, the error of ELM on the test set no longer decreased with the increase in number of nodes. Considering computer resources and efficiency, we set the number of hidden layer nodes of the ELM to 90. Also, it can be seen from Figure 5, the ELM predicted the traffic data of public scene base station. The influence of the number of hidden layer nodes on prediction performance was greater than the time step k.

Figure 6a shows the training and test error curves for different time steps when the number of hidden nodes in the ELM was 90, and Figure 6b is the error curve for vBLR. As can be seen from Figure 6, the value of k corresponding to the minimum RMSE of ELM on the test set is 27, while the value of k corresponding to the minimum RMSE of vBLR is 27. The minimum RMSE of the ELM and vBLR on the test set are 0.0426 and 0.0472, respectively.

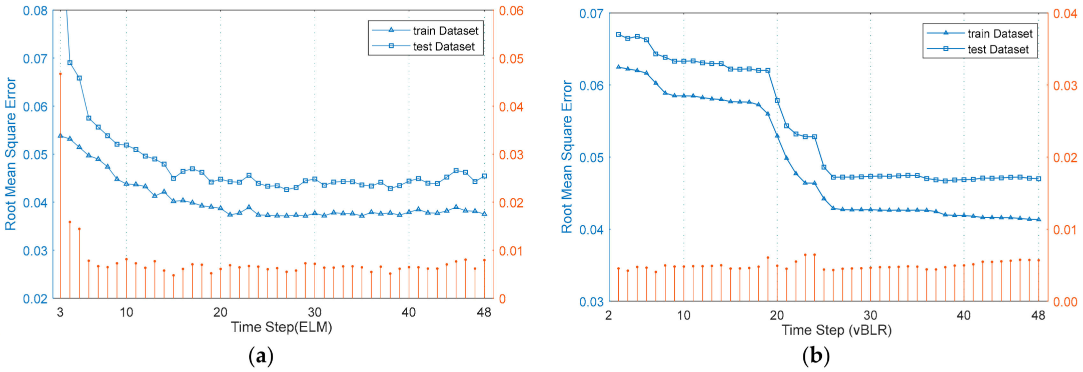


Figure 6. RMSE of ELM and vBLR when k takes different values. (**a**) The RMSE of ELM; (**b**) The RMSE of υSVR.

The RMSE and MAPE comparisons of ELM, vBLR, and SOS-SVR are shown in Table 3. Figure 7 is a comparison of predicted and actual values for each regression model on the test set.

In Table 3, "εSVR" and "SOS-εSVR" represent the settings of default parameters of εSVR and search optimal values, respectively. The "υSVR" parameter setting is a default value. The minimum MAPE of εSVR and υSVR with default parameters on the test set was 43.41% and 14.23%, qfter parameter optimization, the MAPE was 20.61% and 12.36%, respectively. This means that after the parameter optimization, the prediction accuracy of support vector machine for traffic data was improved. Especially the εSVR regression, the error rate of traffic prediction was reduced by more than 20% after SOS parameter optimization. ELM and vBLR predicted traffic data. The MAPE on the test set was 14.35% and 15.98%, respectively, which were higher than the MAPE of SOS-υSVR. Compared with the ELM and vBLR regression algorithms, whether it is RMSE or MAPE, the prediction results of SOS-υSVR for public scene base station data were optimal.

In Figure 7, the left coordinate is the comparison between prediction result of the regression algorithm and actual value. While the right coordinate axis is the subtraction of the predicted value and observed value. From Figure 6, we can more intuitively see that the SOS-υSVR prediction results were optimal.

Table 3. Prediction results of SVR, vBLR, and ELM.

Algorithm	**Optimal Time Step (k)**	**Train**		**Test**	
		RMSE	**MAPE**	**RMSE**	**MAPE**
vBLR	26	0.0429	13.79%	0.0472	15.98%
ELM	27	0.0371	12.42%	0.0426	14.35%
εSVR	24	0.0554	39.05%	0.0624	43.41%
SOS-εSVR	24	0.0432	17.38%	0.0476	20.61%
υSVR	26	0.0402	12.25%	0.0457	14.23%
SOS-υSVR	26	0.0332	9.25%	0.0409	12.36%

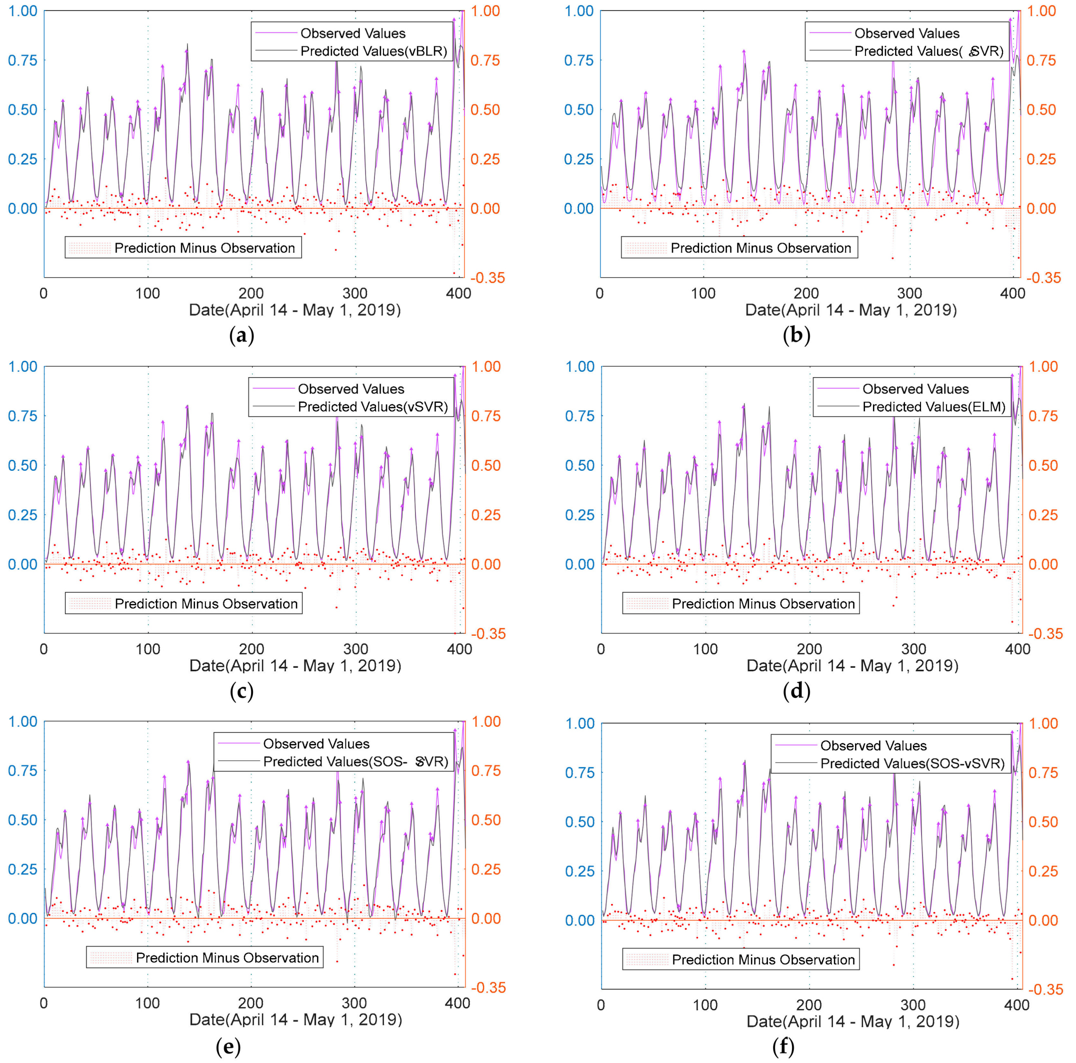


Figure 7. Comparison of predicted and observed values of each regression model on the test set. (**a**) vBLR. (**b**) εSVR (**c**) *υ*SVR. (**d**) ELM. (**e**)SOS-εSVR. (**f**) SOS-*υ*SVR.

4. Conclusions

Traffic prediction of the base station in a public scene is of great significance for network emergency support during festivals and large-scale activities. This paper takes Huainan Wanda Plaza as the experimental object and established a *υ*SVR traffic prediction model. To achieve optimal traffic prediction, we not only used SOS for *υ*SVR parameter optimization, but also optimized the time step. In the experiment, when the time step was 26, the RMSE and MAPE of *υ*SVR optimal traffic prediction on test set were 0.0409 and 12.36%, respectively, and the corresponding parameters c, g, and v were 3.28, 0.2602, and 0.837, respectively.

To verify the superiority of SOS-*υ*SVR in traffic prediction, first, we also used MFO, PSO, and εSVR in experiments for comparison. Results showed that SOS optimizes *υ*SVR parameters with the highest efficiency. Second, ELM and vBLR were adopted to predict traffic data. For fairness, a large number of experiments were used to obtain optimal predictions of traffic data from ELM and vBLR. Results showed that SOS-*υ*SVR has the least error in traffic prediction, whether it is a test set or a training set.

Author Contributions: Data curation, S.F.; methodology, W.L.; writing—original draft, H.C.; writing—review and editing, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National 135 Program "National Key Research Program", No. 2018YFF0301000.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Economic Operation of Communications Industry in January–March 2019. Available online: http://www.miit.gov.cn/n1146285/n1146352/n3054355/n3057511/n3057518/c6802477/content.html (accessed on 23 July 2019).
2. Xu, F.L.; Lin, Y.Y.; Huang, J.X.; Shi, H.Z.; Song, J.; Li, Y. Big Data Driven Mobile Traffic Understanding and Forecasting: A Time Series Approach. *IEEE Trans. Serv. Comput.* **2016**, *9*, 796–805. [CrossRef]
3. Feng, J.; Chen, X.L.; Gao, R.; Zeng, M.; Li, Y. DeepTP: An End-to-End Neural Network for Mobile Cellular Traffic Prediction. *IEEE Netw.* **2018**, *32*, 108–115. [CrossRef]
4. Qiu, C.; Zhang, Y.; Feng, Z.; Zhang, P.; Cui, S. Spatio-Temporal Wireless Traffic Prediction with Recurrent Neural Network. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 554–557. [CrossRef]
5. Li, J.; Wang, J.; Xiong, Z. Wavelet-Based Stacked Denoising Autoencoders for Cell Phone Base Station User Number Prediction. In Proceedings of the IEEE iThings and IEEE- GreenCom and IEEE- CPSCom and IEEE-SmartData, Chengdu, China, 15–18 December 2016; pp. 833–838.
6. Ni, F.; Zang, Y.; Feng, Z. A study on cellular wireless traffic modeling and prediction using Elman Neural Networks. In Proceedings of the 2015 4th ICCSNT, Harbin, China, 19–20 December 2015; pp. 490–494.
7. Dietterich, T.G. *Machine Learning for Sequential Data: A Review*; Springer: Berlin/Heidelberg, Germany, 2002.
8. Kinney, R.W., Jr. ARIMA and Regression in Analytical Review: An Empirical Test. *Account. Rev.* **1978**, *53*, 48–60.
9. Mitchell, T.J.; Beauchamp, J.J. Bayesian variable selection in linear regression. *J. Am. Stat. Assoc.* **1988**, *83*, 1023–1032. [CrossRef]
10. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489. [CrossRef]
11. Schölkopf, B.; Smola, A.J.; Robert, C.; Williamson, R.C.; Peter, L.B. New Support Vector Algorithms. *Neural Comput.* **2000**, *12*, 1207–1245. [CrossRef]
12. Paliwal, M.; Kumar, A.U. Neural networks and statistical techniques: A review of applications. *Expert Syst. Appl.* **2009**, *36*, 2–17. [CrossRef]
13. Mandic, D.P.; Chambers, J. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*; John Wiley & Sons, Inc.: New York, NY, USA, 2001. [CrossRef]
14. Greff, K.; Srivastava, K.R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef]
15. Harpham, C.; Dawson, C.W.; Brown, M.R. A review of genetic algorithms applied to training radial basis function networks. *Neural Comput. Appl.* **2004**, *13*, 193. [CrossRef]
16. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NA, USA, 2013; p. 314. [CrossRef]
17. Wu, C.H.; Ho, J.M.; Lee, D.T. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 276–281. [CrossRef]
18. Santamaría-Bonfil, G.; Reyes-Ballesteros, A.; Gershenson, C. Wind speed forecasting for wind farms: A method based on support vector regression. *Renew. Energy* **2016**, *85*, 790–809. [CrossRef]
19. Kaytez, F.; Taplamacioglu, M.C.; Cam, E.; Hardalac, F. Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *Int. J. Electr. Power Energy Syst.* **2015**, *67*, 431–438. [CrossRef]
20. Chen, R.; Liang, C.Y.; Hong, W.C.; Gu, D.X. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Appl. Soft. Comput.* **2015**, *26*, 435–443. [CrossRef]
21. Qiang, S.; Lu, W.; Du, D.S.; Chen, F.X.; Niu, B.; Chou, K.C. Prediction of the aquatic toxicity of aromatic compounds to tetrahymena pyriformis through support vector regression. *Oncotarget* **2017**, *8*, 49359–49369.
22. Wang, F.K.; Mamo, T. A hybrid model based on support vector regression and differential evolution for remaining useful lifetime prediction of lithium-ion batteries. *J. Power Sources* **2018**, *40*, 49–54. [CrossRef]
23. Wei, J.; Dong, G.; Chen, Z. Remaining Useful Life Prediction and State of Health Diagnosis for Lithium-Ion Batteries Using Particle Filter and Support Vector Regression. *IEEE Trans. Ind. Electron.* **2018**, *65*, 5634–5643. [CrossRef]

24. Golkarnarenji, G.; Naebe, M.; Badii, K.; Milani, A.S.; Jazar, R.N.; Khayyam, H. Support vector regression modelling and optimization of energy consumption in carbon fiber production line Computers. *Chem. Eng.* **2018**, *109*, 276–288.
25. Sivaramakrishnan, K.; Nie, J.J.; Klerk, A. Least squares-support vector regression for determining product concentrations in acid-catalyzed propylene oligomerization. *Ind. Eng. Chem. Res.* **2018**, *57*, 13156–13176. [CrossRef]
26. Jiang, H.; He, W.W. Grey relational grade in local support vector regression for financial time series prediction. *Expert Syst. Appl.* **2012**, *39*, 2256–2262. [CrossRef]
27. Peng, Y.; Albuquerque, P.H.M. The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Syst. Appl.* **2017**, *97*, 177–192. [CrossRef]
28. Ghorbani, M.A.; Shamshirband, S.; Haghi, D.Z.; Azani, A.; Bonakdari, H.; Ebtehaj, I. Application of firefly algorithm-based support vector machines for prediction of field capacity and permanent wilting point. *Soil Tillage Res.* **2017**, *172*, 32–38. [CrossRef]
29. Felipe, A.; Marco, M.; Miguel, O.; Alex, Z.; Claudio, F. A Method to Construct Fruit Maturity Color Scales based on Support Vector Machines for Regression: Application to Olives and Grape Seeds. *J. Food Eng.* **2015**, *162*, 9–17.
30. Cheng, M.Y.; Prayogo, D. Symbiotic organisms search: A new metaheuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [CrossRef]
31. Das, S.; Bhattacharya, A. Symbiotic organisms search algorithm for short-term hydrothermal scheduling. *Ain Shams Eng. J.* **2018**, *9*, 499–516. [CrossRef]
32. Ezugwu, A.E.; Prayogo, D. Symbiotic organisms search algorithm: Theory, recent advances and applications. *Expert Syst. Appl.* **2018**, *119*, 184–209. [CrossRef]
33. Ezugwu, A.E.S.; Adewumi, A.O.; Frîncu, M.E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. *Expert Syst. Appl.* **2017**, *77*, 189–210. [CrossRef]
34. Cheng, M.Y.; Prayogo, D.; Tran, D.H. Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search. *J. Comput. Civ. Eng.* **2015**, *30*, 04015036. [CrossRef]
35. Verma, S.; Saha, S.; Mukherjee, V. A novel symbiotic organisms search algorithm for congestion management in deregulated environment. *J. Exp. Theor. Artif. Intell.* **2015**, *29*, 59–79. [CrossRef]
36. Çelik, E.; Öztürk, N. A hybrid symbiotic organisms search and simulated annealing technique applied to efficient design of PID controller for automatic voltage regulator. *Soft Comput.* **2018**, *22*, 8011–8024. [CrossRef]
37. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
38. Zhang, L.; Zhou, W.D.; Jiao, L.C. Wavelet support vector machine. *IEEE Trans. Syst.* **2004**, *34*, 34–39. [CrossRef] [PubMed]