Article

A Study of Cellular Traffic Data Prediction by Kernel ELM with Parameter Optimization

Xiaoliang Zheng ^{1,2}, Wenhao Lai ^{2,*} , Hualiang Chen ², Shen Fang ³ and Ziqiao Li ²

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 (H.C.); 13644578773@163.com (Z.L.)

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

* Correspondence: whlai9@163.com

Received: 23 April 2020; Accepted: 14 May 2020; Published: 19 May 2020

Abstract: Accurate and efficient prediction of mobile network traffic in a public setting with changing flow of people can not only ensure a stable network but also help operators make resource scheduling decisions before reasonably allocating resources. Therefore, this paper proposes a method based on kernel extreme learning machine (kELM) for traffic data prediction. Particle swarm optimization (PSO), multiverse optimizer (MVO), and moth–flame optimization (MFO) were adopted to optimize kELM parameters for finding the best solution. To verify the predictive performance of the kernel ELM model, backpropagation (BP) neural network, *v*-support vector regression (*v*SVR), and ELM were also applied to traffic prediction, and the results were compared with kELM. Experimental results showed that the smallest mean absolute percentage error in the test (11.150%) was achieved when kELM was optimized by MFO with Gaussian as the kernel function, that is, the prediction result of MFO-kELM was the best. This study can provide significant guidance for network stability and resource conservation.

Keywords: traffic data prediction; public setting; kernel extreme learning machine; parameter optimization

1. Introduction

With the rapid development of communication technology, mobile internet has been closely linked to our daily lives. There will be 5.5 billion mobile phone users in 2021 [1], and mobile data traffic will increase sevenfold. In China, by the end of March 2019, the number of 4G network users had reached 1.204 billion [2] with a per capita monthly flow of 7.27 GB, and it is still maintaining a strong growth trend. Rapidly growing user numbers and cellular traffic data have put a lot of pressure on existing network architectures and devices. However, mobile internet plays an important role in people's life, and its stable operation and security has become very important.

There are many reasons for communication network accidents, such as the weather, malicious attacks, system failures, etc. System failure accounted for more than 60% of communication network accidents in EU countries in 2017. One of the main reasons for mobile internet system failure is that the main control card and baseband card in the base station are overloaded by a sudden increase in traffic demand. As a result, the light network is congested, which leads to a decrease in the internet access speed of mobile phones and affects user experience. In serious cases, it will lead to network equipment failure, which will reduce the connection rate of mobile phones and cause users to be unable to access the internet and make calls. When cell phones are not connected to each other, people often try to

connect to the network again and again, which has a huge secondary shock on wireless networks and core networks. For example, when a concert is being held in a gym, the network traffic suddenly increases due to early failure in trying to support communication, resulting in overload of the main control card and baseband card in the base station equipment, causing card congestion failure. Due to this failure, the speed of internet access sharply reduces, and users cannot access the internet and make calls.

The demand for mobile network traffic data presents a remarkable spatiotemporal characteristic. A sudden increase in traffic data demand often occurs in settings with changing flow of people, such as commercial streets, stadiums, and railway stations. The prediction of cellular traffic data in settings with changing traffic flow allows operators to understand network traffic demand in advance and achieve on-demand distribution through resource scheduling. This could effectively alleviate network pressure and reduce the impact and damage caused by a sudden increase in traffic demand on the network as well as improve user experience. The radio system has been optimized for maximum load, which results in excessive energy waste in case of low traffic [3]. The information and communication technology (ICT) industry consumed 3–4% of the world's electricity in 2008, and this use has been growing rapidly by doubling every decade [4]. Prediction of cellular traffic could help to allocate network resources on-demand, selectively shut down and rest base stations, and also reduce the power consumption of networked devices. Therefore, mobile network traffic prediction is not only of great significance in network security but also in energy conservation.

There have been some relevant reports on prediction of cellular traffic data [5–9]. Wu et al. [5] described the regularized orthogonal matching tracking (BT-ROMP) cellular flow prediction method based on threshold control. The theoretical basis of this method is compressed sensing technology; the principle is complex, and the calculation is time-consuming. In addition, the rational setting of threshold requires mass experimentation. He and Li [6] reported a mobile communication base station traffic prediction method based on the vector autoregressive model. The autoregressive model is simple and easy to implement, but the learning ability is weak. Meanwhile, the prediction accuracy needs to be improved. Loumiotis et al. [7] investigated the backhaul resource allocation problem on one side of the base station. Flow prediction model was established using an artificial neural network. Compared with the autoregressive model, the prediction performance was improved. However, artificial neural networks are difficult to train, and they can easily fall into the local optimum. In order to achieve higher performance, Qiu et al. [8] employed multiple recurrent neural network (RNN) learning models to construct traffic prediction models by exploring spatiotemporal correlations between base stations. The structure of RNN neurons is more complex than traditional artificial neural networks. This method constructs multiple RNNs, which makes the prediction model more complicated. In addition, the model's training and testing only take 15 days, and its performance regarding flow prediction needs to be further verified. Therefore, a novel and efficient flow forecasting method is proposed in this paper.

Extreme learning machine (ELM) [10], a typical single-hidden layer feedforward neural network (SLFN), learns extremely fast and does not fall into the local optimum compared with backpropagation (BP) algorithms. In order to further improve the performance of ELM, Huang et al. [11] proposed a kernel ELM (kELM) by referring to the idea of feature mapping of a support vector machine (SVM), which retains the high efficiency of ELM and inherits the excellent learning ability of SVM. The technique has been successfully used for classification [12–16] and regression tasks [17–21]. Therefore, in this study, we employed kELM to predict cellular flow data. There are four kernel functions in the classical SVM. In order to achieve optimal prediction of traffic data by kELM, four classical kernel functions were used for flow prediction. The meta-inspired optimization algorithm [22] was used for parameter optimization, and the kernel function with the optimal result was selected. Wolpert and Macready [23] as well as other authors have proven that none of the optimization algorithms can search for the optimal results of all optimized problems. Therefore, we chose the classic particle swarm optimization method (PSO) [24] as well as two heuristic optimization algorithms with strong searchability, namely,

multiverse optimizer (MVO) [25] and moth–flame optimization (MFO) [26], for parameter optimization of ELM with different kernel functions.

The rest of the paper is structured as follows. In Section 2, network accidents caused by traffic are described, and the proposed method is introduced. Section 3 outlines the experimental preparation for validation of the proposed method. In Section 4, the experimental results are presented and analyzed. Section 5 concludes the paper.

2. Accident Description and Prediction Method

2.1. Accident Description

When network resources are limited, the overload of base stations caused by a sudden increase in cellular traffic demand could result in network accidents. These accidents often occur in public settings with changing traffic. Table 1 shows some network accidents caused by an increase in traffic demand in different settings and events in the Huainan area in the past two years.

Table 1. Network accidents caused by increased traffic.

Setting	Time	Reasons	A Network Failure (Caused by Congestion)	Consequences (Number)
Shopping mall	2019.09.19 18:00–20:00	Annual Celebration Concert	Short-term passenger flow increased too much and demand for network traffic increased, resulting in 4GTD-LTE/FDD-LTE base station equipment overload.	About 1500 people surfed the internet at reduced speeds. About 130 people did not have normal internet access and calls.
Commercial Street	2018.5.1 10:00–22:00	Labor Day	The steep increase and gathering of people during the holidays resulted in limited resources of mobile 4G network equipment and overload of base stations.	Internet download rate was limited for about 1700 people, and user experience was reduced.
Train station	2018.10.01 2018.10.07	National Day	The rapid increase in the flow of people in specific areas led to the limitation of mobile 4G wireless resources and the overload of base stations.	About 1200 people surfed the internet at a reduced speed.
Bagong Mountain Scenic Area	2019.04.16 00:00–21:00	Bagong Hill Temple Fair	During the temple fair, the flow of people increased sharply, resulting in limited mobile 4G wireless resources and overloaded base stations.	About 2300 people surfed the internet at a reduced speed, and call success rate dropped for 120 people.

Table 1 only shows one network accident caused by increase in cellular traffic demand in shopping malls, commercial streets, railway stations, and tourist attractions. However, in the network operation, the reduction in user experience caused by an increase of flow in each setting is much more than once. Traffic prediction of public settings helps us understand the upcoming traffic peak in advance. Good network protection can effectively reduce accidents. When we know that the network is about to be congested, we will increase our capacity by carrying out steps to alleviate network load.

1. Based on user-focused areas and distribution of existing network base stations, check the load situation of surrounding sites and find the community occupied by user terminal frequency.
2. Inquire whether hardware configuration of the corresponding high-load community meets the requirements for capacity expansion. If so, carry out remote capacity expansion in the background; generally, the corresponding setting is added to the software. If not, on-site hardware expansion is required.

3. After the main occupation cell is expanded to full-load configuration standard of hardware and software, it is necessary to pay attention to occupation of the base station in real time and adjust the load balance in real time.
4. If it is still under high load after full matching and balancing, the interoperation parameters between base stations needs to be changed to balance users to surrounding sites. The base stations with a light load in surrounding areas will bear the traffic load.

After the evaluation, if existing network equipment cannot meet the scheduling needs, it is necessary to use network emergency support vehicle. At this stage, the network emergency communication vehicle uses three 4G cellular networks, each with three frequency point configurations: FDD1800M, TDD's D1 frequency point, and TDD's D2 frequency point. The direction carrier bandwidth of each cellular capacity is 60M, which can meet the normal internet access and traffic needs of 1200 people.

2.2. Prediction Methods

2.2.1. Kernel ELM

The ELM is a new SLFN that overcomes the disadvantages of traditional neural networks, such as slow training speed, easily falling into a local minimum point, and sensitive learning rate selection. The connection weight between the input layer, the hidden layer, and threshold values of neurons in the hidden layer of ELM are all random. There is no need for adjustment in the training process. The weights are obtained through a matrix operation, which means fast learning speed and strong generalization ability.

Although ELM has excellent learning speed and generalization ability, its learning ability needs to be improved. Huang et al. borrowed the idea of feature mapping of SVM to improve ELM, called kELM, which not only inherits the advantages of ELM but also has the same learning performance as SVM.

Let $T = \{(x_1, y_1), \dots, (x_l, y_l)\}$, $x \in R^n$, $y \in R^m$. The ELM model with l hidden nodes and $g(\cdot)$ as the activation function is as follows:

$$f(x_j) = \sum_{i=1}^N \beta_i g(w_i \cdot x_j + b_i) \quad (1)$$

where $w_i = [w_{i1}, \dots, w_{in}]$ is the weight of the i -th input neuron and hidden neurons, $\beta_i = [\beta_{i1}, \dots, \beta_{im}]$ is the weight between the i -th hidden layer neuron and the output layer, and b is the bias.

Let

$$G = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_l \cdot x_1 + b_l) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_n + b_1) & \cdots & g(w_l \cdot x_n + b_l) \end{bmatrix}_{n \times l}, \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_l^T \end{bmatrix}_{l \times m}, F = \begin{bmatrix} f_1^T \\ \vdots \\ f_l^T \end{bmatrix}_{n \times m}$$

Equation (1) can be written as follows:

$$G\beta = F \quad (2)$$

In the BP neural network, Equation (2) is solved by multiple iterations using the error backpropagation method. However, Equation (2) is solved by matrix operations in the ELM as shown in Equation (3):

$$\beta = G^+ F \quad (3)$$

where $G^+ = (G^T G)^{-1} G^T$. The mechanism tells us that the core factor of fast ELM learning is that the assignment of bias b and weight w between the input layer and hidden layer is a random number. Meanwhile, it is not modified during the model learning process, and the weight β is obtained through matrix operations.

The solution obtained in Equation (3) is just one solution, but it is impossible to avoid errors in practical situations. In applications, the solution of ELM model can also be written as follows:

$$\|G\beta - F\| = \min_{\beta} \|G\hat{\beta} - F\| \quad (4)$$

In Equation (4), the obtained β is an approximate solution of the weight matrix, which, to a certain extent, improves model generalization. However, its learning ability still needs to be improved. Huang et al. improved the ELM through feature mapping, called kELM, and its mathematical model is as follows:

$$\begin{aligned} \text{Minimize : } L_{kELM} &= 0.5\|\beta\|^2 + 0.5C\sum_{i=1}^n \|\varepsilon_i\|^2 \\ \text{s.t. } g(x_i)\beta &= f_i - \varepsilon_i \end{aligned} \quad (5)$$

where $\varepsilon_i = [\varepsilon_{i1}, \dots, \varepsilon_{im}]$ is the training error vector, and C is a parameter set by the user. According to Karush–Kuhn–Tucker (KKT) conditions, Equation (5) can be transformed into its dual problem as follows:

$$L'_{kELM} = 0.5\|\beta\|^2 + 0.5C\sum_{i=1}^n \|\varepsilon_i\|^2 - \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}(g(x_i)\beta_j - f_{i,j} + \varepsilon_{i,j}) \quad (6)$$

where $\alpha_{i,j}$ is the Lagrangian operator. The solution of Equation (6) is as follows:

$$\beta = G^T \left(\frac{I}{C} + GG^T \right)^{-1} F \quad (7)$$

Therefore, the output of kELM model is as follows:

$$\hat{y}(x) = g(x)\beta = g(x)G^T \left(\frac{I}{C} + GG^T \right)^{-1} F \quad (8)$$

where, $\hat{y}$ is the model output.

The SVM algorithm maps features to high-dimensional spaces through kernel functions to improve its learning performance, and the kELM model takes the above methods into consideration to improve its own performance.

$$\hat{y}(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_n) \end{bmatrix} \left(\frac{I}{C} + K(X, X) \right)^{-1} F \quad (9)$$

where $K(\cdot)$ is a kernel function.

From Equation (9), it can be seen that the weight and offset of the input layer to the hidden layer of kELM are obtained through the mapping operation of the kernel function, eliminating the influence of random weights on ELM performance. The number of neurons and bias in the hidden layer need not be set and optimized.

The normal kernel functions are Gaussian, polynomial, linear, and sigmoid. They can be written as follows:

$$K(x, y) = \exp(-\|x - y\|^2 / k_1) \quad (10)$$

$$K(x, y) = (x^T \cdot y + k_1)^{k_2} \quad (11)$$

$$K(x, y) = x^T \cdot y \quad (12)$$

$$K(x, y) = \tanh(k_1 x^T \cdot y + k_2) \quad (13)$$

where k is the parameter of the kernel function, which is set by the user. Whether its value is reasonable or not closely affects the performance of kELM. In order to realize the optimal prediction of cellular traffic data by kELM, all four kernel functions were used in this study for feature mapping of kELM.

Then, the optimal mapping method was selected from them. In addition, the optimization algorithm was used to search for optimal parameter values of the kernel function.

2.2.2. kELM Parameter Optimization

For the kernel ELM algorithm, it is important to select appropriate parameters. Manual selection of the optimal parameters requires a large number of experiments, and the process is tedious. The optimal parameters may not be found. In this study, the metaheuristic optimization algorithm was used for kELM parameter optimization. Wolpert and Macready (1997) pointed out that none of the optimization algorithms can solve all optimization problems. There are four kernel functions that are commonly used. In order to find the optimal parameters of kELM with different kernel functions, we employed three optimization algorithms to optimize the parameters of kELM: MFO, MVO, and PSO.

PSO was first proposed by Eberhart and Kennedy in 1995, and its basic concept was derived from the study of the foraging behavior of birds, with particles used to simulate individual birds. Each particle can be regarded as a searching individual in the n -dimensional search space. The current position of a particle is a candidate solution of the corresponding optimization problem. The flight process of a particle is the individual search process. The flight speed of a particle can be dynamically adjusted according to the historical optimal position of the particle and the historical optimal position of the population.

MVO is an optimization algorithm proposed by Mirjalili et al. and inspired by the multiverse theory. In the multiverse theory, there are multiple parallel universes in the world, and these universes exchange matter through white holes, black holes, and wormholes. The MVO algorithm takes expansion rate as fitness function and uses the material exchange capacity of white holes, black holes, and wormholes to improve the expansion rate of the universe so as to search for the best location of the universe.

MFO is a new group intelligence optimization algorithm proposed by Mirjalili in 2015. It is derived from the navigation method of transverse positioning of the night moth, which maintains its flight by maintaining its own angle of light relative to the moon. The near-parallel light of surface is used to keep its flight straight. The main advantages of the algorithm are simple structure, few parameters, and better optimization ability.

The three optimization algorithms were used for optimal parameter searching of kELM, as shown in Figure 1. The pseudocode is shown in Table 2.

Table 2. The pseudocode of multiverse optimizer (MVO), particle swarm optimization (PSO), and moth–flame optimization (MFO) to optimize the parameters of kELM.

Algorithm PSO, MFO, and MFO to optimize the parameters of kELM
Input: Max_Iterations, Boundaries, X, Population Initialization all parameters while Iteration <= Max_Iterations for each individual indexed by j in Population Evaluate fitness of each individual: $\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i / y_i \times 100\%$ end for Search for the optimal fitness of the current iteration and update for each individual indexed by j in Population Update individual location end for Iteration $\leftarrow$ Iteration + 1 Output: C, k

Figure 1. Algorithm steps to optimize the kernel extreme learning machine (kELM) prediction model.

3. Experimental Preparation

3.1. Data Acquisition and Processing

The experimental setting in this study was a commercial street in Huainan that has a large flow of people and is largely influenced by festivals. In other words, the experimental data was the cellular flow data of the base station covering the setting. All experimental data in the study were provided by the Huainan Branch of Anhui Mobile Co., Ltd (Huainan, China). The data acquisition time period was from 15 April 2019 to 24 September 2019. Our data was evenly collected 24 times a day, i.e., the acquisition interval was 1 h. The obtained flow data is shown in Figure 2.

Figure 2. Cellular flow data.

In Figure 2, it can be seen that the overall cellular traffic during July and August were higher than those for other months, and the peak of traffic was also relatively high in the days around 1 May and 1 June. This is because July and August are summer holidays, 1 May is International Labor Day, and 1 June is Children's Day. During this period, most people generally choose to go out for dinner and shopping, which also causes the traffic flow of commercial streets to be significantly higher than during the working period. From Figure 2, it can be seen that the maximum daily peak value of cellular traffic data was more than twice the minimum peak value. A sudden increase in traffic demand will

bring great pressure to the existing network architecture. Predicting traffic is of great significance for network security and resource scheduling as it allows us to do the scheduling work in advance to deal with the upcoming congestion.

The raw data obtained was a sequence with a length of 3912 (163×24), which needed to be processed into several equal-length sequences as the input to the prediction model and the corresponding output before the experiment. The processing of the data is shown in Figure 3.

Figure 3. Sequence Processing.

In Figure 3, s is the time step. In this study, we set the time step to 24. n is the total length of collected sequence ($n = 3912$ in this study), and $X = \{X_1, X_2, X_3\}$, and $Y = \{Y_1, Y_2, Y_3\}$ are the inputs and outputs of the prediction model, respectively. In the experiment, we used data from 4.15 to 8.24 as the training set and the remaining data as the test set for a well-trained model test.

3.2. Experimental Setup

The kernel limit learning machine has different flow prediction performance depending on the kernel function selected. Even if it is the same kernel function with different parameters, the results of flow prediction will be different. Therefore, it is necessary to optimize the kernel function and its parameter when the kernel limit learning machine is used for flow prediction. In this study, in order to maximize the optimal flow prediction of the kernel limit learning machine, we applied the metaheuristic optimization algorithm for parameter optimization of kELM.

The parameter settings of the metaheuristic optimization algorithm in the experiment are shown in Table 3. In kernel function parameter optimization, for the same kernel function, the variable boundary settings of all optimization algorithms are the same, as shown in Table 4.

Table 3. Initial parameters of MVO, PSO, and MFO.

Algorithm	Parameter	Value
MVO	WEP	[0.2 1.0]
	Iterations	50
	Number of universes	12
	Acceleration constants	[1.5 1.9]
PSO	Inertia w	[0.75, 0.75]
	Generations	30
	Number of particles	12
	b	1
MFO	Iterations	12
	Number of search agents	50

Table 4. Variable boundaries.

Kernel Function	Upper	Lower
Gaussian [C k ₁]	[1000, 1000]	[0.001, 0.001]
Linear [C]	1000	0.001
Polynomial [C k ₁ k ₂]	[1000, 10, 6]	[0.001, 0, 0]
Sigmoid [C k ₁ k ₂]	[1000, 10, 10]	[0.001, −10, −10]

When the metaheuristic optimization algorithm searches for the parameters of kELM, the performance evaluation mechanism of kELM is the mean absolute percentage error (MAPE), as shown in Equation (14). To prevent overlearning, we used MAPE of the test set as the fitness function of the optimization algorithm.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (14)$$

where y_i is the actual cellular traffic value, and $\hat{y}_i$ is the predicted value.

4. Experiment Results and Analysis

4.1. Performance Analysis of ELM with Different Kernel Functions

In this section, we investigate the predictive performance of kELM for cellular flow under different kernel functions. As the prediction performance of kELM is different when the parameters of different kernel functions take different values, we used the optimal prediction of flow of different kernel functions for comparison. Three metaheuristic optimization algorithms with excellent performance were used for parameter optimization of kELM. In the process of parameter optimization, each experiment was independently repeated 10 times. Finally, the optimal results were determined. The curves of MAPE of the training set the test sets of kELM with different kernel functions are shown in Figures 4 and 5.

Figure 4. The mean absolute percentage error (MAPE) curve of the training set. (a) Gaussian, (b) linear, (c) polynomial, (d) sigmoid.

Figure 5. The MAPE curve of the test set. (a) Gaussian, (b) linear, (c) polynomial, (d) sigmoid.

In Figure 4, it can be seen that when the kernel function of ELM was Gaussian and polynomial, the value of MAPE appeared to increase with the number of iterations of the optimization algorithm. However, as shown in Figure 5, all kernel functions decreased with the number of iterations of the optimization algorithm. This is because we used the minimum MAPE of the test set as a function of the fitness of the optimization algorithm to prevent model overlearning.

From Figures 4 and 5, it can be intuitively seen that, in the initial stage of parameter optimization, the value of MAPE was largest when the kernel function of kELM was sigmoid. The parameters for the three optimization algorithm searches led to its MAPE being greater than 60%. Then, its MAPE decreased rapidly as parameter optimization was carried out, which meant that the sigmoid kernel function predicted the cellular flow more sensitively to the parameters in the given variable interval. In addition, the linear kernel function had relatively minimal variation of the optimal MAPE for the search of the three optimization algorithms, but its optimal MAPE was the largest for both the training and test sets. The optimal results of the four kernel functions of MVO, MFO, and PSO for optimized kELM are shown in Table 5.

Table 5 shows in detail the optimal results of kELM under multiple optimizations of the four kernel functions by the three metaheuristic algorithms. From Table 4, we can see that when the kernel function was Gaussian, the MAPE of the MFO search was the smallest at 11.150%. When the kernel function was polynomial, the MAPE of the MVO search was the smallest at 11.495%, and the result of the MFO search was relatively large at 11.611%. This indicates that the ability of the same metaheuristic optimization algorithm to search parameters of different kELM kernel functions is different, which is the main reason we chose multiple search algorithms for parameter optimization in this study.

Table 5. Results of optimizing the kELM by different search algorithms.

Kernel	MFO			MVO			PSO		
	Time (s)	Train	Test	Time (s)	Train	Test	Time (s)	Train	Test
Gaussian	149.49	9.411%	11.150%	146.18	9.156%	11.236%	148.91	9.767%	11.193%
Linear	97.69	15.449%	15.234%	96.86	15.448%	15.236%	96.43	15.477%	15.293%
Polynomial	227.67	9.938%	11.611%	235.40	9.909%	11.495%	255.35	9.739%	11.496%
Sigmoid	223.80	11.928%	11.715%	224.17	11.379%	11.650%	195.61	11.692%	12.123%

kELM predicts cellular traffic in the public setting. For the test set, the Gaussian kernel function had the minimum MAPE at 11.150%, while the polynomial kernel function had the second smallest MAPE at 11.495%. The MAPE was the largest when the kernel function was linear, both for the test and training sets. In order to prevent overlearning, the fitness function of each optimization algorithm should have the minimum MAPE in the test set. Therefore, the kernel function of the minimum MAPE of the test set and the optimization algorithm were combined for cellular flow prediction of the base station in the public setting. In other words, we used the results of Gaussian kernel function of MFO-kELM optimization for the following experimental comparison.

The “time” in Table 5 is the time consumed in the parameter optimization process. From this table, we can also see that no matter which optimization algorithm was used, when the kernel function of kELM was linear, the optimization time was the lowest, followed by the Gaussian kernel function. At the same time, the parameter optimization time of the polynomial and sigmoid kernel functions were relatively high. As can be seen from Table 5, when the kernel function of kELM was linear, only one parameter needed to be optimized, whereas two parameters needed to be optimized for the Gaussian kernel function and three parameters needed to be optimized for the polynomial and sigmoid kernel functions. This indicates that the kELM parameter optimization time is closely related to the number of parameters optimized. Furthermore, although the kELM parameter optimization process took up to 100 s, the time was mainly consumed in the process of optimal algorithm optimization for selecting the best parameters.

4.2. Study of the Prediction Using Other Regression Algorithms

As noted in the previous section, we investigated the performance of kELM’s predicted cellular traffic data under different kernel functions. As a second step, we used v -support vector regression (v SVR), backpropagation neural networks, and basic ELM for predicting cellular traffic data in public settings and compared the results. To study the predictive traffic data of kELM, we optimized its parameters. To make the comparison accurate, we first searched for the optimal prediction of traffic data by v SVR, BP, and ELM.

To optimize the parameters of v SVR, we chose MFO as the search algorithm. All the variable settings were kept the same for optimization of MFO by v SVR as those for optimization of kELM parameters except for the variable boundary settings. The upper and lower boundary settings of parameters c , g , and v to be optimized were [1500, 10,001] and [0.01, 0.01], respectively. To prevent overlearning, the MAPE for the MFO test set was used as the fitness function. The MAPE curve in the optimization process is shown in Figure 6.

In pattern recognition or machine learning, the performance of BP neural network and ELM is closely related to the nodes in the hidden layer. A small number of nodes will lead to weak model learning, increase the model training time, and even lead to overlearning, which will degrade the model performance. Therefore, the BP neural network and ELM should select the appropriate number of hidden layer nodes. In our experiment, the implementation of BP was done in a MATLAB neural network toolbox. The learning rate was set to 0.2, the maximum fail was set to 30, the hidden layer and output layer activation function selection was {‘logsig’, ‘tansig’}, and other parameters were set to default.

As the initial part of ELM weights and all the weights of the BP neural network are random, these will affect its performance, especially the BP neural network. When the initial weights are not set

properly, the model will be difficult to train and may not even converge. Therefore, we independently repeated 100 trials under different neural nodes (which means that the number of our experiments was as high as $100 \times 150 + 100 \times 60$). Except for the nonconvergence, the results took the mean. The mean absolute error (MAE) curves of the training and test sets for BP and ELM at different hidden layer nodes are shown in Figures 7 and 8.

Figure 6. MAPE curve for optimization of *v*-support vector regression (*v*SVR) parameters.

Figure 7. MAPE curves of ELM with different hidden layer nodes.

Figure 8. MAPE curves of backpropagation (BP) at different hidden layer nodes.

In Figure 5, although the fitness function of the MFO-optimized *v*SVR was the MAPE for the test set, its training set MAPE decreased with the increase in iteration times. At the end of optimization, the values of parameters *c*, *g*, and *v* were 1371.092, 0.024, and 0.891, respectively.

In Figure 6, it can be seen that the average error of ELM of the training set decreased with the increase in hidden layer nodes. However, the error of the test set barely decreased when the number of

hidden layer nodes was greater than 100. Therefore, the hidden layer node value of ELM was 100. In Figure 7, the average error of the BP training set decreased with the increase in hidden layer nodes. When the number of nodes was greater than 30, the error of the test set increased with the increase in nodes, showing that the model had a learning phenomenon. In conclusion, when predicting cellular flow data, the hidden layer node of BP should be set to 30.

From Figures 7 and 8, we can also see that the consumption time of ELM had an approximately linear relationship with the number of hidden nodes. In contrast, for BP, there was an exponential relationship, and the training time of BP was much longer than that of ELM.

4.3. Comparison with Other Regression Algorithms

We studied the predictive cellular flow data of kELM and other regression algorithms and compared the prediction results of other regression algorithms with those of kELM, as shown in Table 6.

Table 6. kELM compared with other regression algorithms.

Algorithm	MAPE		Standard Deviation of MAPE		Parameters	Time (s)
	Train	Test	Train	Test		
MFO-kELM (Gaussian)	9.411%	11.150%	0	0	[622.078, 1.770]	149.49
MFO-vSVR	10.265%	11.082%	0	0	[1371.092, 0.024, 0.891]	11,405.70
ELM	13.448%	13.367%	0.374%	0.608%	100	—
BP neural network	13.066%	12.128%	0.881%	0.340%	30	—

In Table 6, the parameters of MFO-ELM (Gaussian) and MFO-vSVR are [C ag] and [c, g, v], while the parameters for ELM and BP are the number of hidden layer nodes. “Time” is the parameter optimization time.

From Table 6, it can be seen that MFO-vSVR had the smallest MAPE in the test set at 11.082%, while the MFO-kELM had the smallest MAPE in the training set at 9.411%. However, kELM was more efficient, and its optimization time was 149.49 s, which was much less than the optimization time of 11,405.70 s for vSVR. The worst performance for cellular flow prediction was ELM, which had the largest MAPE in both the test and training sets. Differences between the results predicted by each regression algorithm and actual value of cellular traffic is shown in Figure 9. The standard deviation of kELM and vSVR was 0, while that of BP and ELM was not 0, i.e., the training results of kELM and vSVR were more stable.

Figure 9. The predicted values of each regression algorithm minus the actual values. (a) kELM, (b) vSVR, (c) ELM, (d) BP.

In addition, from Table 6, we can also see that the introduction of kernel functions in ELM to map features to high-dimensional space not only significantly improved the prediction accuracy of ELM but also eliminated the uncertainty brought by the random initial weight to model prediction performance.

5. Conclusions

In mobile network operations, network failures caused by a sudden increase in cellular traffic data demand often occur. Therefore, it is of great practical significance to predict the flow of mobile networks in settings with changing traffic flow for stable network operation and resource scheduling. In order to realize accurate prediction of cellular flow data, this study analyzed the performance of kELM in predicting cellular traffic data with different kernel functions. In the experiment, to determine optimal parameters of kernel function, three metaheuristic optimization algorithms (PSO, MVO, and MFO) were adopted.

The results showed that kELM optimized by MFO with Gaussian as kernel function had the smallest test set MAPE (11.150%). Moreover, we used ELM, BP, and SVR for flow prediction to verify the performance of kELM. The optimal prediction results of ELM, BP, and SVR for cellular flow data were obtained through a large number of experiments. kELM had a significant advantage in prediction accuracy compared with ELM and BP. Although the prediction accuracy of SVR was as good as that of kELM, the optimization time for SVR was very long, and its prediction efficiency was low.

We studied an efficient prediction model of mobile traffic based on the kELM algorithm. A commercial street in Huainan was selected to verify the effectiveness of the model through experiments. The proposed kELM-based traffic forecasting method will allow operators to prepare for coping with upcoming congestion and improve service quality. Meanwhile, it could also guide network operators to rationally allocate network resources, effectively saving energy and reducing operating costs.

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

Funding: This work was supported in part by the National 135 Program “National Key Research Program” under Grant 2018YFF0301000.

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

References

1. 2017–2022 White Paper, Cisco Visual Networking Index: Forecast and Trends. Available online: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html> (accessed on 24 October 2019).
2. Operation Monitoring and Coordination Bureau, *Economic Operation of Communications Industry in January–March 2019*; MIIT: Beijing, China, 2019. Available online: <http://www.miit.gov.cn/n1146312/n1146904/n1648372/c6802908/content.html> (accessed on 2 November 2019).
3. Blume, O.; Eckhardt, H.; Klein, S.; Kuehn, E.; Wajda, W.M. Energy savings in mobile networks based on adaptation to traffic statistics. *Bell Labs Tech. J.* **2010**, *15*, 77–94. [CrossRef]
4. Pickavet, M.; Vereecken, W.; Demeyer, S.; Audenaert, P.; Vermeulen, B.; Delders, C. Worldwide Energy Needs for ICT: The Rise of Power-Aware Networking. In Proceedings of the 2008 2nd International Symposium on Advanced Networks and Telecommunication Systems, Mumbai, India, 15–17 December 2008.
5. Jia, W.; Yun, Z.; Li, J.Z.; Wen, S. Prediction of cellular traffic based on space-time compression sensing. *Comput. Mod.* **2018**, *280*, 15–19.
6. He, Y.; Li, Y.T. Forecasting the traffic flow of base station based on vector auto-regression. *Ind. Eng. Manag.* **2013**, *22*, 79–84.
7. Loumiotis, I.; Adamopoulou, E.; Demestichas, K.; Kosmides, P.; Theologou, M. *Artificial Neural Networks for Traffic Prediction in 4G Networks*; International Wireless Internet Conference: Lisbon, Portugal, 2014; pp. 141–146.

8. Qiu, C.; Zhang, Y.Y.; Feng, Z.Y.; Zhang, P.; Cui, S.G. Spatio-temporal wireless traffic prediction with recurrent neural network. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 554–557. [\[CrossRef\]](#)
9. Ni, F.X. Cellular wireless traffic prediction based on improved wavelet-Elman neural network. *Electron. Des. Eng.* **2017**, *25*, 171–175.
10. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [\[CrossRef\]](#)
11. Huang, G.; Zhou, H.; Ding, X.; Zhang, R. Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst.* **2011**, *42*, 513–529.
12. Cao, J.W.; Xiong, L.L. Protein sequence classification with improved extreme learning machine algorithms. *Biomed. Res. Int.* **2014**, *2014*, 12. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Shang, W.; Wu, Z.B.; Xu, Y.; Zhang, Y. Hyperspectral supervised classification using mean filtering based kernel extreme learning machine. In Proceedings of the 2018 Fifth International Workshop on (EORSA), Xi'an, China, 17–20 June 2018.
14. Chen, Z.; Cao, J.; Lin, D.; Wang, J.; Huang, X. Vibration source classification and propagation distance estimation system based on spectrogram and KELM. *Cogn. Comput. Syst.* **2019**, *1*, 26–33. [\[CrossRef\]](#)
15. Zeng, Y.J.; Xu, X.; Shen, D.Y.; Fang, Y. Traffic sign recognition using kernel extreme learning machines with deep perceptual features. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1647–1653. [\[CrossRef\]](#)
16. Iosifidis, A.; Tefas, A.; Pitas, I. Approximate kernel extreme learning machine for large scale data classification. *Neurocomputing* **2017**, *219*, 210–220. [\[CrossRef\]](#)
17. Shamshirband, S.; Mohammadi, K.; Chen, H.L.; Samy, G.N.; Petković, D.; Ma, C. Daily global solar radiation prediction from air temperatures using kernel extreme learning machine: A case study for Iran. *J. Atmos. Sol.-Terr. Phys.* **2015**, *134*, 109–117. [\[CrossRef\]](#)
18. Yaseen, Z.M.; Deo, R.C.; Hilal, A.; Abd, A.M.; Bueno, L.C.; Salcedo-Sanz, S.; Nehdi, M.L. Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. *Adv. Eng. Softw.* **2018**, *115*, 112–125. [\[CrossRef\]](#)
19. Chen, Y.H.; Kloft, M.; Yang, Y.; Li, C.; Li, L. Mixed kernel based extreme learning machine for electric load forecasting. *Neurocomputing* **2018**, *312*, 90–106. [\[CrossRef\]](#)
20. Tang, Q.; Zhang, J.H.; Xie, Z.Y. Short-term micro-grid load forecast method based on EMD-KELM-EKF. In Proceedings of the 2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG), Taipei, Taiwan, 15–17 December 2018.
21. Parida, N.; Mishra, D.; Das, K.; Rout, N. Development and performance evaluation of hybrid KELM models for forecasting of agro-commodity price. *Evol. Intel.* **2019**. [\[CrossRef\]](#)
22. Fister, I., Jr.; Yang, X.; Fister, I.; Brest, J.; Fister, D. A brief review of nature-inspired algorithms for optimization. *arXiv* **2013**, arXiv:1307.4186.
23. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1996**, *1*, 67–82. [\[CrossRef\]](#)
24. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory, MHS'95. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995.
25. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [\[CrossRef\]](#)
26. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [\[CrossRef\]](#)