

A Short-Term Prediction Model Based on Support Vector Regression Optimized by Artificial Fish-Swarm Algorithm

GuiPing Wang^{ab}, ShuYu Chen^{c*}, Jun Liu^b and TianShu Wu^b

^aCollege of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, China

^bCollege of Computer Science, Chongqing University, Chongqing, China

^cCollege of Software Engineering, Chongqing University, Chongqing, China

w_guiiping@163.com, netmobilab@126.com, liujun_314@cqu.edu.cn, wutianshu@cqu.edu.cn

Abstract

In urban management, it is important to precisely forecast the short-term demand for necessary resources, including water, electric power, and gas. Although a variety of prediction models have been proposed in literature, the underlying defects and limitations confine the effectiveness and forecasting precision of these models. In this paper, the short-term prediction problem is modeled as a non-linear multivariate regression problem, which is solved by support vector regression (SVR). The parameters in SVR are optimized by artificial fish-swarm algorithm (AFSA). The proposed prediction model (termed SVR-AFSA) overcomes the defects of existing prediction models, thus promoting forecasting precision. In order to verify the effectiveness and prediction precision of SVR-AFSA, this paper conducts experiments on a real dataset of two-month hourly water consumption. It also compares SVR-AFSA with two commonly adopted models, i.e., traditional BP neural network, and SVR optimized by grid method (SVR-grid). The experiments results show that SVR-AFSA outperforms these two models in prediction precision in terms of mean squared error (MSE) and mean absolute percentage error (MAPE).

Keywords: Prediction, Regression Analysis, Support Vector Machine (SVM), Support Vector Regression (SVR); Parameter Optimization, Artificial Fish-Swarm Algorithm (AFSA)

1. Introduction

In urban management, an important task is to supply sufficient resources (e.g., water, electric power, gas) necessary for human life. Therefore, precisely forecasting the demand for these resources plays an important role to avoid resource shortage. Utilities need to know what the demand for today and tomorrow will be (i.e., short-term) to operate their infrastructure plants appropriately and meet these demands. Utilities also need to predict the demand 20–30 years in the future (i.e., long-term) to develop new sources and/or expand their existing infrastructure plants [1]. Short-term prediction and long-term prediction needs different information. This paper only focuses on short-term (e.g., daily or even hourly) forecasting.

In recent years, many prediction models, including multivariate regression [2], artificial neural network [2], [3], [4], Fuzzy model [5], and grey model [6], are proposed in literature. The application fields of these models differ, depending on the forecast variable, its periodicity, and the forecast horizon [1]. Donkor *et al.* [1] review the literature on urban water demand forecasting published from 2000 to 2010, which shows that artificial neural networks are more likely to be used for short-term forecasting.

Although the above prediction models have attracted much attention in literature, the underlying defects and limitations confine the effectiveness and prediction precision of

these models. For example, grey model can be constructed by only a few samples; however, it is only applicable to monotonously increasing or decreasing scenarios. Moreover, when the discrete degree (namely data grey) of data is great, the prediction precision is low. In the case of small samples, neural networks may result in such problems as low prediction precision, over-fitting, and local minima. In fuzzy model, the determination of memberships and fuzzy rules mainly depend on human experience and expertise.

Aiming at overcoming the above defects and therefore precisely forecasting urban short-term water consumption, this paper first models the short-term prediction problem as a non-linear multivariate regression problem. It then proposes a prediction model (termed SVR-AFSA) based on support vector regression (SVR) whose parameters are optimized by artificial fish-swarm algorithm (AFSA). This paper conducts experiments on a real dataset of two-month hourly water consumption collected from a water treatment plant which supplies water for a region in Chongqing municipality. It also compares the performance of SVR-AFSA, traditional BP neural network, and SVR optimized by grid method (SVR-grid). The experiments results show that SVR-AFSA outperforms the latter two models in prediction precision.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 introduces SVR and AFSA, and detailedly presents the proposed prediction model, *i.e.*, SVR-AFSA. Section 4 conducts experiments on a real dataset and presents experimental analyses. Finally, section 5 gives conclusions and looks into future work.

2. Related Work

This section summarizes research work related to regression analysis, SVM, and swarm intelligence.

2.1. Prediction and Regression

Let

$$T = \{ (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \} \quad (1)$$

be the *training dataset*, where a (\mathbf{x}_i, y_i) pair is a single *training sample*; $\mathbf{x}_i = \{ x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)} \} \in \Xi \subseteq R^n$ is the *n-dimensional input variable* which contains *n features*: $x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}, i = 1, 2, \dots, N$; $y_i \in \Psi \subseteq R$ is the *target variable* (or *output variable*); *N* is the number of training samples, *i.e.*, the *size* of the training dataset. Note that, the subscript represents the sequence number of a training sample in the dataset; while the superscript represents the sequence number of a feature in a training sample.

Prediction is a learning problem. Its task is to learn a model, such that given a test input, the model predicts corresponding output. Prediction can be classified into three categories: *classification*, *regression*, and *tagging*. When the target variable can take only a small number of discrete values, the learning problem is called a *classification* problem [7]. When the predicted target variable is continuous, the learning problem is called a *regression* problem [1]-[6]. When both the input variable and the output one are variable sequences, the learning problem is called a *tagging* problem [8].

Since the short-term water demand is a continuous variable, this paper only focuses on the continuous target variable scenario. Therefore, unless explicitly stated otherwise, the terms “*regression*” and “*prediction*” are interchanged in the remainder of this paper.

Although many prediction models have been proposed in literature, including multivariate regression [2], artificial neural network [2], [3], [4], Fuzzy model [5], and grey model [6]. Usually, these models are faced with problems including low prediction

precision, low convergence speed, over-fitting, and local minima. Therefore, novel and effective prediction models (*e.g.*, SVM) are still attracting researchers' attention.

2.2. Support Vector Machine (SVM)

The original SVM algorithm was invented by Vladimir N. Vapnik and the current standard incarnation (soft margin SVM) was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995 [9]. Currently, SVMs have been widely and successfully applied in many fields including text categorization, speech recognition, remote sensing image analysis, and time series forecasting [10].

In nature, SVM is a kind of supervised learning method. When it is used for classification, SVM is often referred to as SVC (Support Vector Classification). When it is used for regression, SVM is often referred to as SVR (Support Vector Regression). For linearly separable classification problems (the simplest scenario), SVC constructs two supporting hyperplanes with maximum geometric margin to separate training samples with different categories, as shown in Figure 1(a). The symbols, "+" and "o", represent vector points with different categories. The dash lines and the solid line represent the supporting hyperplanes and the classifying hyperplane, respectively. Support vector machine, as the name suggests, is divided into two sections: firstly, a support vector is a vector point on these two supporting planes, while a hyperplane right in the "middle" between these two supporting planes is the classifying hyperplane which separates two categories of vector points; secondly, the machine is an algorithm. In the field of machine learning, some algorithms are often referred to as machines.

For general linearly classification problems (not restricted to be linearly separable ones), SVC introduces slack variables to loose constraints, which derive the concept of "soft margin". Slack variables allow the existence of some outliers which cannot be separated by the classifying hyperplane, as shown in Figure 1(b).

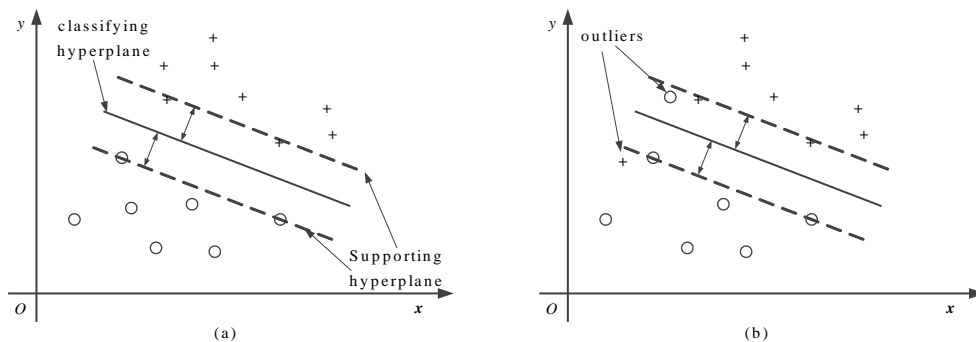


Figure 1. The Basic Idea of SVMs for Classification Problems (i.e., SVCs)

Given the training dataset (1), the goal of regression is to find a curve $y = g(\mathbf{x})$ which best fits the given data points, as shown in Figure 2(a) and 2(c), where "x" represents a data point. For linear regression problems, the curve is a linear function (*i.e.*, a hyperplane in high dimensional space): $y = (\mathbf{w} \cdot \mathbf{x}) + b$.

The linear support vector regression (SVR) is established by converting the linear regression problems to the linear classification ones, as shown in Figure 2(b). Each data point, x_i , is moved up and down with the distance of ϵ , and the points x_i^+ and x_i^- are obtained. x_i^+ and x_i^- are joined, thus obtaining the line segments $x_i^+ x_i^-$, just as shown by the dash lines in Figure 2(b). These line segments are open. Lastly, any line that passes through these open line segments is a hard ϵ -band hyperplane.

For non-linear regression, as shown in Figure 2(c), SVR maps the original input space (referred to as \mathbf{x}) into a high dimensional space (referred to as \mathbf{x}) by using a

kernel function so as to best fit the data points in the new space by a hyperplane, thus converting the non-linear problem into a linear one in new space, as shown in Figure 2(d).

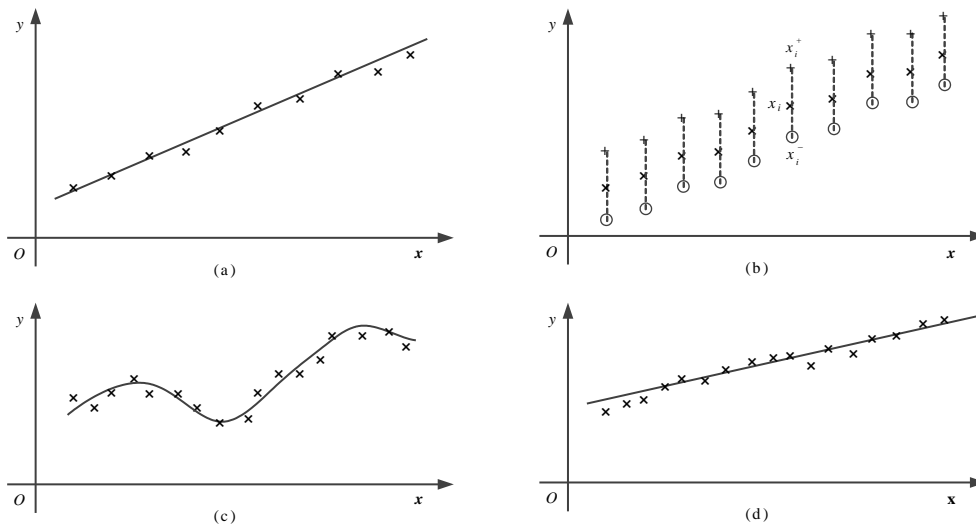


Figure 2. The Basic Idea of SVMs for Regression Problems (i.e., SVRs)

Compared with traditional neural networks, SVM has the following advantages:

(1) SVM is designed for problems with small sample size. SVM can obtain the optimal solution even in the case of limited samples.

(2) SVM is eventually transformed into a quadratic programming problem. Theoretically speaking, it can obtain the global optimal solution, thus avoiding the local optima problem of traditional neural networks.

(3) The topology of SVM is determined by the support vectors; while the traditional neural networks require repeated exploring to determine its topology.

(4) By using nonlinear mapping in kernel functions, SVM maps the original problem into a high dimensional feature space, and constructs a linear classification function in the high dimensional space, which not only ensures good generalization ability of the model, but also solves the curse of dimensionality.

LIBSVM ([11]) is a most popular library for SVMs. It is implemented by C++ and Java languages, and extended to Python, MATLAB, Perl, Ruby, *etc.* In this paper, the proposed prediction model is implemented and simulated in Matlab with LIBSVM.

2.3. Swarm Intelligence (SI) and Artificial Fish-Swarm Algorithm (AFSA)

The expression “swarm intelligence (SI)” was first used by Beni, Hackwood, and Wang [12], [13] in the context of cellular robotic systems. SI refers to any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies [14].

Artificial Fish-Swarm Algorithm (AFSA) is a novel method to search global optimum, which is a typical application of behaviorism in artificial intelligence. This algorithm is inspired by the collective movement of the fish and their various social behaviors. AFSA was first proposed in 2002 [15]. The basic idea of AFSA is to imitate the fish behaviors such as preying, swarming, and following with local search of fish individual for searching for the global optimum. AFSA is a random and parallel search algorithm. It is one of the best optimization methods among the swarm intelligence algorithms. A latest survey on AFSA can be found in [16].

Currently, AFSA is widely applied in combinatorial optimization [17], and parameter optimization of neural networks [18] and SVM [19]. In [18], a radial basis function neural network (RBFNN) is adopted to train data and forecast the stock indices, where AFSA is used to optimize the parameters in RBFNN. In [19], a SVC model with parameters optimized by AFSA is proposed to identify the critical features determining the pattern of a botnet.

In this paper, AFSA is used to optimize the parameters in a SVR model. Combining the advantages of SVR and AFSA, this paper proposes a regression model based on SVR and AFSA to predict short-term urban water consumption.

3. A Prediction Model Based on SVR Optimized by AFSA

For short-term urban water consumption, this section first constructs a prediction model (SVR-AFSA). Then, it details SVR and AFSA algorithms.

3.1. A Prediction Model for Forecasting Short-Term Urban Water Consumption

The short-term urban water consumption prediction problem can be modeled as a non-linear multivariate regression problem (as shown in Figure 3), which is solved by SVR.

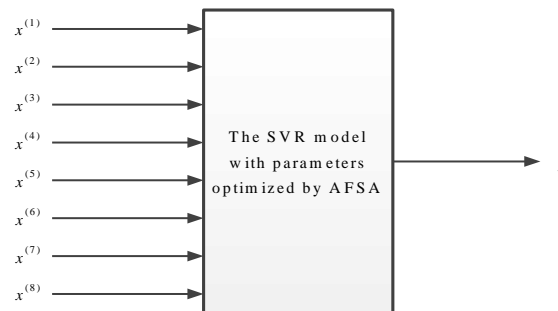


Figure 3. The SVR Model with Parameters Optimized by AFSA

Short-term urban water consumption is a periodic non-stationary stochastic process, which is affected by the type of day, weather condition (maximum and minimum temperature). Day type contains workday, weekend and major holidays. Short-term water consumption forecasting mainly needs information including historical consumption data, and weather condition. Therefore, in order to precisely forecast the short-term demand, the input variable, $\mathbf{x} = \{x^{(1)}, x^{(2)}, \dots, x^{(8)}\}$, consists of 8 features, which are listed as follows:

$x^{(1)}$: The actual hourly water consumption of the i -th hour in the same weekday (in the week before last) as the forecasting one.

$x^{(2)}$: The actual hourly water consumption of the i -th hour in the same weekday (in the last week) as the forecasting one.

$x^{(3)}$: The actual hourly water consumption of the i -th hour in the day before last.

$x^{(4)}$: The actual hourly water consumption of the i -th hour in the last day.

$x^{(5)}$: The actual hourly water consumption of the $(i-2)$ -th hour in the forecasting weekday.

$x^{(6)}$: The actual hourly water consumption of the $(i-1)$ -th hour in the forecasting weekday.

$x^{(7)}$: The difference in maximum temperature between the forecasting weekday and the day before last.

$x^{(8)}$: The difference in maximum temperature between the forecasting weekday and the last day.

The target variable, y , is the predicted hourly water consumption of the i -th hour in the currently forecasting weekday.

3.2. ε -SVR

The above non-linear multivariate regression problem is solved by the ε -SVR algorithm list as follows:

(1) Input the training dataset, $T = \{ (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N) \}$, where $\mathbf{x}_i \in \mathbf{X} \subseteq R^n$, $y_i \in \Psi = R$, $i = 1, 2, \dots, N$;

(2) Choose an appropriate kernel $K(\mathbf{x}, \mathbf{x}')$, an appropriate accuracy $\varepsilon > 0$, and the penalty parameter $C > 0$;

(3) Construct and solve the convex quadratic programming problem:

$$\min_{\alpha, \alpha^* \in R^N} \frac{1}{2} \sum_{i,j=1}^N (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon \sum_{i=1}^N (\alpha_i^* + \alpha_i) - \sum_{i=1}^N y_i (\alpha_i^* - \alpha_i), \quad (2)$$

$$\text{s.t.} \quad \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0, \quad (3)$$

$$0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C, i = 1, 2, \dots, N, \quad (4)$$

thus obtaining a solution $\bar{\alpha}^{(*)} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \dots, \bar{\alpha}_N, \bar{\alpha}_N^*)^T$;

(4) Compute \bar{b} : Choose a component of $\bar{\alpha}^{(*)}$, $\bar{\alpha}_j \in (0, C)$ or $\bar{\alpha}_k^* \in (0, C)$. If $\bar{\alpha}_j$ is chosen, compute:

$$\bar{b} = y_j - \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) K(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon; \quad (5)$$

If $\bar{\alpha}_k^*$ is chosen, compute:

$$\bar{b} = y_k - \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) K(\mathbf{x}_i, \mathbf{x}_k) - \varepsilon; \quad (6)$$

(5) Construct the decision function:

$$y = g(\mathbf{x}) = \sum_{i=1}^N (\bar{\alpha}_i^* - \bar{\alpha}_i) K(\mathbf{x}_i, \mathbf{x}) + \bar{b}. \quad (7)$$

The commonly adopted kernel function is Gaussian radial basis function kernel with a parameter σ , which is defined by:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / \sigma^2). \quad (8)$$

The most important parameters in ε -SVR are C and σ , which are optimized by AFSA in this paper, as illustrated in the next sub-section.

After constructing the decision function $y = g(\mathbf{x})$, the water demand at a given hour can be predicted given the test input \mathbf{x} .

3.3. Parameter Optimization by AFSA

When optimizing parameters by AFSA, each artificial fish (denoted as X_i) represents a solution to the optimization problem. In this paper, X_i consists of the parameters in ε -SVR (*i.e.*, C and σ). The corresponding objective function value Y is the Mean Squared Error (MSE) of the SVR by using the parameters contained in X_i .

The whole artificial fish swarm is represented by:

$$X = \{X_1, X_2, \dots, X_i, \dots, X_n\}, \quad (9)$$

where n is the size of the swarm.

The environment of an artificial fish mainly contains the solution space of an optimization problem and the state of other artificial fish (referred to as companions). In the next moment, X_i 's behavior depends on its own state and the current state of the environment (including quality of the current solution, and the state of companions).

Conversely, through its activities, X_i also affects the environment and the state of companions. X_i perceives the outside world through its vision. For simplicity, the model of artificial fish adopts the method illustrated in Figure 4 to represent and implement vision.

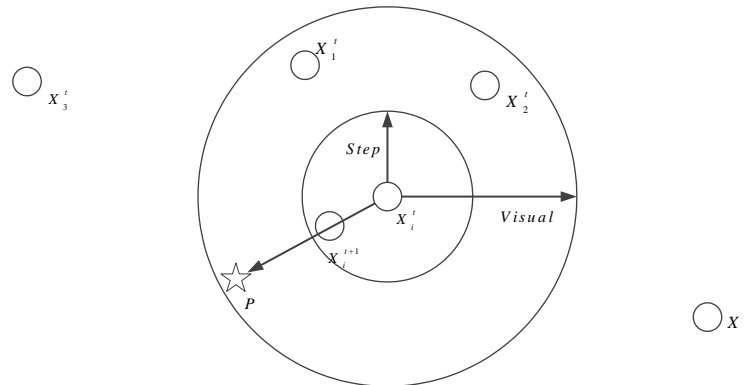


Figure 4. The Vision of an Artificial Fish, X_i

Figure 4 illustrates the vision of an artificial fish (AF). The AF is denoted as X_i . X_i^t is the current (*i.e.*, the t moment) state of X_i . *Visual* is its visual distance, and *step* is the length of a move. There are a certain number of companions (X_1^t and X_2^t in Fig. 4) in X_i 's vision. P is the visual position at some moment. If the state at P is better than the current state (X_i^t), X_i goes forward a step in this direction, and arrives the X_i^{t+1} state; otherwise, X_i continues an inspecting tour in the vision. The greater number of inspecting tour X_i does, the more knowledge about overall states of the vision the AF obtains ([16]). Certainly, it does not need to travel throughout complex or infinite states, which is helpful to find the global optimum by allowing certain local optimum with some uncertainty ([16]).

For an artificial fish, AFSA defines the following four basic behaviors.

(1) Preying behavior

Let X_i^t be the current state of X_i . When preying, X_i randomly selects a state, X_j , in its vision:

$$X_j = X_i + \text{Visual} \cdot \text{Rand} () , \quad (10)$$

where $\text{rand}()$ generates a random real number between 0 and 1. Y is the food concentration, which corresponds to the MSE (the smaller, the better) obtain by the SVR model. If $Y_j < Y_i$, X_i goes forward a step in this direction by:

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \cdot \text{Step} \cdot \text{Rand} () . \quad (11)$$

Otherwise, X_i selects a state X_j randomly again and judge whether it satisfies the forward condition (*i.e.*, $Y_j < Y_i$). If the condition cannot be satisfied after *Try_number* times, X_i moves a step randomly by:

$$X_i^{t+1} = X_i^t + \text{Visual} \cdot \text{Rand} () . \quad (12)$$

The preying behavior can be illustrated by the flow diagram in Figure 5.

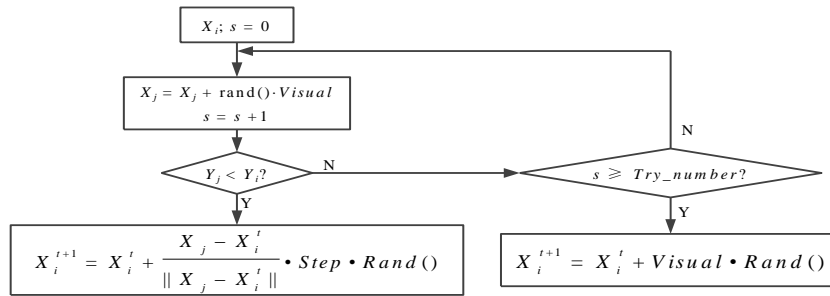


Figure 5. The Flow Diagram of Preying Behavior of X_i ,

(2) Swarming behavior

Let n_f be the number of companions in the current neighborhood of state of X_i (i.e., the distance between X_i and X_j , $d_{ij} < Visual$), and X_c be the center position of these companions. If $Y_c / n_f > \delta Y_i$ (where δ is the crowd factor, $0 < \delta < 1$), which means that the center X_c has more food (higher fitness function value) and is not very crowded, then X_i goes forward a step towards X_c by:

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{\|X_c - X_i^t\|} \cdot Step \cdot Rand() . \quad (13)$$

Otherwise, X_i executes the preying behavior.

The swarming behavior can be illustrated by the flow diagram in Figure 6.

(3) Following behavior

When a single fish or several ones find food, the neighbor companions will follow and reach the food quickly. X_i explores the companion X_j in the neighborhood ($d_{ij} < Visual$), which has the greatest Y_j . If $Y_j / n_f > \delta Y_i$, which means that the state of companion X_j has a higher food concentration (higher fitness function value) and the surrounding is not very crowded, X_i goes forward a step towards the companion X_j by:

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\|X_j - X_i^t\|} \cdot Step \cdot Rand() . \quad (14)$$

Otherwise, X_i executes the preying behavior.

The following behavior can be illustrated by the flow diagram in Figure 7.

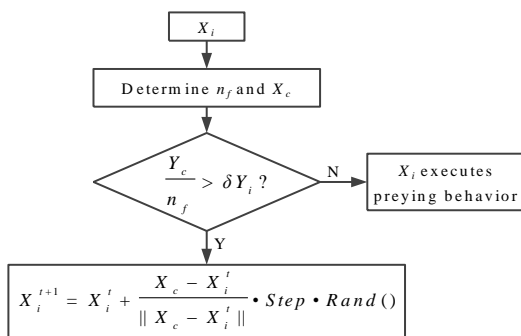


Figure 6. The Flow Diagram of Swarming Behavior of X_i

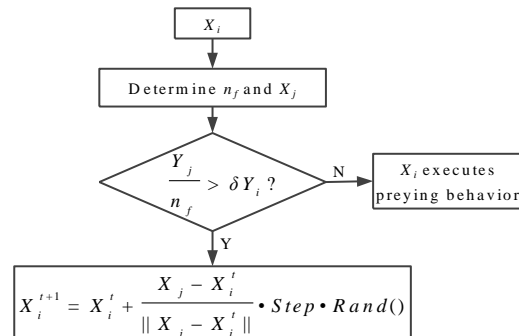


Figure 7. The Flow Diagram of Following Behavior of X_i

(4) Randomly moving behavior

X_i chooses a random state in the vision, and then it moves towards this state by the equation (12). In fact, randomly moving behavior is a default behavior of preying, as illustrated in Figure 5.

The whole flow diagram of AFSA is illustrated in Figure 8.

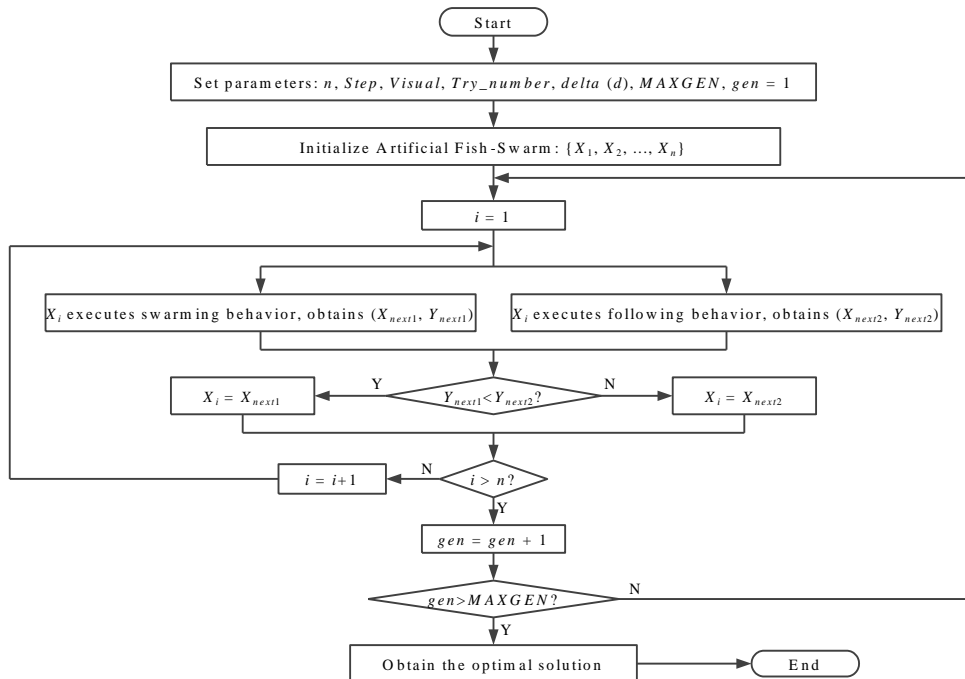


Figure 8. The Flow Diagram of AFSA

The notations of ASFSA are summarized as follows:

$X = \{X_1, \dots, X_i, \dots, X_n\}$: the whole artificial fish swarm, where X_i is a single artificial fish.

n : the size of the swarm.

$step$: the length of a move.

$Visual$: the visual distance of an artificial fish, X_i .

Try_number : the number of trial in preying behavior.

δ (δ): the crowd factor, $0 < \delta < 1$

$MAXGEN$: the maximum generation in AFSA.

4. Experiments and Analyses

This section conducts experiments on a real dataset, and compares SVR-AFSA with traditional BP, and SVR optimized by grid method (SVR-grid).

4.1. Dataset Description

The real dataset is two-month (Jan., and Feb. in 2014) hourly water consumption data collected from a water treatment plant which supplies water for a region in Chongqing municipality. Totally, there are $(31+28)*24 = 1416$ samples. Since the proposed prediction model (shown in Figure 3) needs the consumption data two weeks ago to predict hourly water consumption in the forecasting weekday, the data of the former 2 weeks in Jan. are neglected. Therefore, the number of remaining samples is 1080. Table 1 shows the former 10 samples, where $\{x^{(1)}, \dots, x^{(8)}\}$ is the input variable as shown in Figure 3, and y is actual hourly water consumption of the i -th hour in the

currently forecasting weekday, which can be used to compute mean squared error (MSE) and mean absolute percentage error (MAPE) of a prediction model.

Table 1. The Remaining Samples of the Real Dataset

hour	$x^{(1)}$ (ton/h)	$x^{(2)}$ (ton/h)	$x^{(3)}$ (ton/h)	$x^{(4)}$ (ton/h)	$x^{(5)}$ (ton/h)	$x^{(6)}$ (ton/h)	$x^{(7)}$ (°C)	$x^{(8)}$ (°C)	y (ton/h)
0	5010	4891	5303	5599	5039	4815	-2	-3	4987
1	5303	5235	5731	5941	4815	4987	-2	-3	5381
2	6268	6211	6679	6923	4987	5381	-2	-3	6369
3	7507	7360	7851	8051	5381	6369	-2	-3	7428
4	8070	7910	8392	8601	6369	7428	-2	-3	7964
5	8712	8544	9091	9213	7428	7964	-2	-3	8591
6	9525	9419	9893	10101	7964	8591	-2	-3	9527
7	10323	10231	10679	10927	8591	9527	-2	-3	10354
8	10651	10534	11083	11323	9527	10354	-2	-3	10631
9	10994	10892	11371	11605	10354	10631	-2	-3	11005
...

During experiments, 500 samples are randomly selected as training samples, and the rest 580 samples are selected as testing samples.

4.2. Parameters Setting

This paper implements SVR-AFSA in Matlab with LIBSVM ([11]). The parameters in AFSA are set as follows: $n = 20$; $step = 1.25$; $Visual = 10$; $Try_number = 5$; $delta (\delta) = 0.3$; $MAXGEN = 20$.

This paper also compares SVR-AFSA with two commonly adopted models, *i.e.*, traditional BP neural network, and SVR optimized by grid method (SVR-grid).

The parameters in BP are set as follows: the size of N-1 hidden layers, 10; learning rate, 0.1; the number of epochs, 1000; the goal of mean square error, $1e-3$.

The parameters in SVR-grid are set as follows: $minC = -10$, $maxC = 10$, $minS = 10$, $maxS = 10$ (the minimum and maximum values of C and σ), $step = 0.5$ (the enumeration step of C and σ).

4.3. Experimental Results and Analyses

After obtaining the learning model by using the above three algorithms, this paper takes 580 testing samples as input, and predicts the corresponding water consumption. It compares the performance of these three algorithms. The most important evaluation metric is mean squared error (MSE), which is computed by:

$$MSE = \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2 . \quad (15)$$

where $f(x_i)$ is the predicted value for each testing sample, x_i ; while y_i is the actual value.

The experimental results are listed in Table 2. Since hourly water consumption is a large value, MSE is a much larger value. The results show that MSE in SVR-grid is slightly smaller than that in BP; while MSE in SVR-AFSA is evidently smaller than that in BP and SVR-grid.

Table 2. The Experimental Results (MSE) on All Testing Samples

	SVR-AFSA	BP	SVR-grid
MSE	58822.38	151928.59	126287.96

Particularly, this paper compares relative error (RE) and mean absolute percentage error (MAPE) of these three algorithms on 24 samples on Feb. 18. RE is computed by:

$$RE_i = \frac{f(x_i) - y_i}{y_i} \times 100 \% . \quad (16)$$

MAPE is computed by:

$$MAPE = \frac{1}{N} \sum_{i=1}^N abs(ER_i) . \quad (17)$$

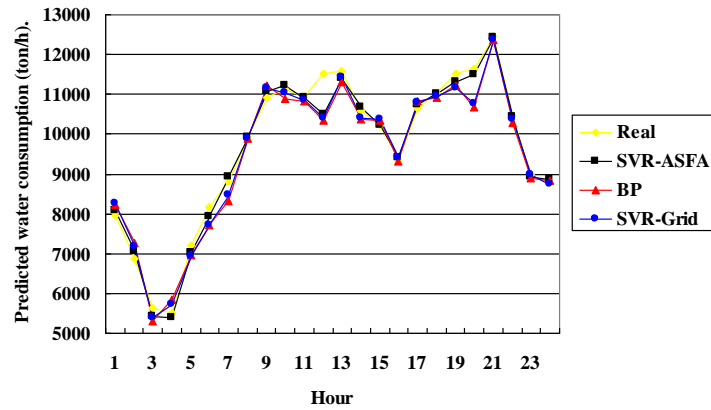
where $abs()$ is a function returns the absolute value. MAPE reflects the prediction precision of a prediction algorithm.

The experimental results are listed in Table 3. For each hour on Feb. 18, Table 3 lists the actual value of water consumption. Moreover, for each algorithm, Table 3 lists the predicted value $f(x_i)$ and RE of each testing sample, and the ultimate MAPE. The experimental results are also pictorialized in Figure 9, where Figure 9(a) shows the hourly predicted value, while Figure 9(b) shows the hourly RE value.

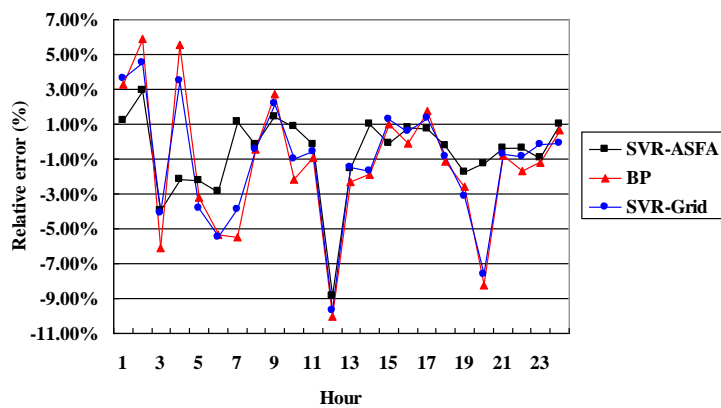
The results in Table 3 and Figure 9 both show that SVR-grid slightly outperforms BP; while SVR-AFSA evidently outperforms BP and SVR-grid.

Table 3. The Experimental Results on Samples of Feb. 18

hour	y (ton/h)	SVR-AFSA			BP			SVR-grid		
		f(x) (ton/h)	RE (%)	MAPE (%)	f(x) (ton/h)	RE (%)	MAPE (%)	f(x) (ton/h)	RE (%)	MAPE (%)
0	7972	8067.4	1.20%		8232.3	3.27%		8262.1	3.64%	
1	6859	7060.2	2.93%		7262	5.88%		7170.4	4.54%	
2	5632	5409.7	-3.95%		5289.9	-6.07%		5399.7	-4.13%	
3	5527	5407.4	-2.16%		5833.6	5.55%		5718.4	3.46%	
4	7197	7034.9	-2.25%		6968.4	-3.18%		6921.1	-3.83%	
5	8162	7931.2	-2.83%		7723.4	-5.37%		7715.3	-5.47%	
6	8814	8911.6	1.11%		8328	-5.51%		8467.9	-3.93%	
7	9925	9906.8	-0.18%		9877.3	-0.48%		9877.9	-0.47%	
8	10908	11060.1	1.39%		11206.2	2.73%		11147.4	2.19%	
9	11139	11233.2	0.85%		10897	-2.17%		11030.3	-0.98%	
10	10938	10915.6	-0.20%		10839.4	-0.90%		10871.6	-0.61%	
11	11506	10487.5	-8.85%		10353.5	-10.02%		10390.2	-9.70%	
12	11584	11407.3	-1.53%		11313.8	-2.33%		11415.9	-1.45%	
13	10571	10677.8	1.01%		10373.3	-1.87%		10391.4	-1.70%	
14	10234	10221.7	-0.12%		10336	1.00%		10364.5	1.28%	
15	9338	9412.3	0.80%		9326.6	-0.12%		9394.4	0.60%	
16	10648	10725.1	0.72%		10834.4	1.75%		10788.6	1.32%	
17	11035	11008.6	-0.24%		10908.5	-1.15%		10942	-0.84%	
18	11522	11315.4	-1.79%		11223.9	-2.59%		11157.4	-3.16%	
19	11646	11495.5	-1.29%		10688	-8.23%		10760.3	-7.61%	
20	12467	12415.8	-0.41%		12364	-0.83%		12380.5	-0.69%	
21	10466	10424.9	-0.39%		10291.4	-1.67%		10373.9	-0.88%	
22	8998	8914.3	-0.93%		8888.5	-1.22%		8980.5	-0.19%	
23	8767	8854.1	0.99%		8825.6	0.67%		8757.3	-0.11%	
		1.59%			3.11%			2.62%		



(a) Hourly Predicted Water Consumption



(b) Hourly Relative Error

Figure 9. The Pictorialization of Experimental Results on Feb. 18

From the experimental results shown in Table 2 and Table 3, it is concluded that SVR-ASFA exhibits better generalization performance on testing samples. The underlying reasons are summarized as follow:

(1) Multivariate regression problems are usually non-linear. Through the introduction of kernel functions, SVM can map the original input space into a high dimensional space so that it make possible to find a hyperplane that best fits the data points in the new space.

(2) SVR converts the linear regression problems to the linear classification problems. For linear classification problems, SVC maximizes the geometric margin between two supporting hyperplane. By introducing the penalty parameter C , SVM seeks a better trade-off between two conflicting goals: maximizing the margin and minimizing the training error.

As for SVR-grid, the grid method is an enumeration method, which exhaustively tests each combination of the values of C and σ . If the enumeration step is set as a small value, SVR-grid may find the more optimum parameters with smaller MSE and MAPE, but it consumes much computation time. If the enumeration step is set as a large value, SVR-grid may miss the optimum parameters. It is hard to seek a trade-off between computation time and optimum parameters.

5. Conclusion and Future Work

For short-term continuous variable prediction, this paper proposes a support vector regression model whose parameters are optimized by AFSA. The model is termed SVR-AFSA. The experimental results show that, compared with SVR-grid and traditional BP neural network, SVR-AFSA exhibits better generalization performance for testing samples.

The future work of this paper will focus on the performance of other kernel functions (*e.g.*, sigmoid, polynomial) in non-linear mapping. In addition, some new and novel behaviors of artificial fish swarm are explored in literature. These behaviors are expected to promote the performance of AFSA. In future, this paper will conduct experiments on these extended ASFAs to verify their performance in parameter optimization.

Acknowledgments

We are grateful to the editors and anonymous reviewers for their valuable comments and feedback on this paper.

The work of this paper is partially supported by National Natural Science Foundation of China (Grant No. 61272399) and Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110191110038).

References

- [1] E. Donkor, T. Mazzuchi, R. Soyer and J. A. Roberson, "Urban Water Demand Forecasting: Review of Methods and Models", *Journal of Water Resources Planning and Management*, vol.140, no.2, (2014), pp.146-159.
- [2] J. Adamowski and C. Karapataki, "Comparison of multivariate regression and artificial neural networks for peak urban Water-Demand Forecasting: Evaluation of Different ANN Learning Algorithms", *Journal of Hydrologic Engineering*, vol.15, no.10, (2010), pp.729-743.
- [3] G.-C. Liao and T.-P. Tsao, "Application of fuzzy neural networks and artificial intelligence for load forecasting", *Electric Power Systems Research*, vol.70, no.3, (2004), pp.237-244.
- [4] M. Beccali, M. Cellura, V. Lo Brano and A. Marvuglia, "Forecasting daily urban electric load profiles using artificial neural networks", *Energy Conversion and Management*, vol.45, no.18, (2004), pp.2879-2900.
- [5] H. B. Liu, T. G. Deng and H. W. Zhang, "Research on forecasting method of urban water demand based on fuzzy theory", In: *Proceedings of the 6th international conference on Fuzzy systems and knowledge discovery (FSKD)*, (2009).
- [6] C.-C. Hsu and C.-Y. Chen, "Applications of improved grey prediction model for power demand forecasting", *Energy Conversion and Management*, vol.44, no.14, (2003), pp.2241-2249.
- [7] E. Avci, "Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm - support vector machines: HGASVM", *Expert Systems with Applications*, vol.36, no.2, (2009), pp.1391-1402.
- [8] P. Heymann, D. Ramage and H. Garcia-molina, "Social tag prediction", In: *Proceedings of International Conference on Research and Development in Information Retrieval (ACM-SIGIR)*, (2008).
- [9] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol.20, no.3, (1995), pp.273-279.
- [10] N. Y. Deng, Y. J. Tian and C. H. Zhang, "Support Vector Machines - Optimization based Theory, Algorithms, and Extensions", CRC Press, (2013).
- [11] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, vol.2, no.3, (2011), Article 27.
- [12] G. Beni, "The Concept of Cellular Robotic System", In: *Proceedings of IEEE International Symposium on Intelligent Control*, (1988).
- [13] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems", In: *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, (1989); Italy.
- [14] E. Bonabeau, M. Dorigo and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, USA, (1999).
- [15] X. L. Li, Z. J. Shao and J. X. Qian, "An optimizing method based on autonomous animats: fish-swarm algorithm", *System Engineering Theory and Practice*, vol.22, no.11, (2002), pp.32-38.
- [16] M. Neshat, G. Sepidnam, M. Sargolzaei and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications", *Artificial Intelligence Review*, published online, (2012).

- [17] Md. A. K. Azada, A. Maria A.C. Rocha and Edite M.G.P. Fernandes, "A simplified binary artificial fish swarm algorithm for 0-1 quadratic knapsack problems", *Journal of Computational and Applied Mathematics*, vol.259, (2014), pp.897-904.
- [18] W. Shen, X. P. Guo, C. Wu and D. S. Wu, "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm", *Knowledge-Based Systems*, vol.24, no.3, (2011), pp.378-385.
- [19] K.-C. Lin, S.-Y. Chen and J. C. Hung, "Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm", *Journal of Applied Mathematics*, (2014).

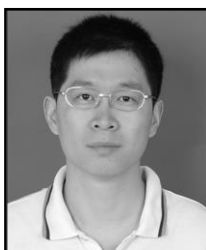
Authors



GuiPing Wang, he received his B.S. degree and M.S. degree in Chongqing University, P. R. China, at 2000 and 2003, respectively. Since July 2003, he acts as a lecturer in Department of Computer Science, Zhejiang University of Finance and Economy. Currently he is a full-time Ph.D. candidate in College of Computer Science, Chongqing University. His research interests include machine learning, dependability analysis and design of distributed systems, and cloud computing. As the first author, he has published nearly 20 papers in related research areas during recent years at journals including IPL, and ESWA.



ShuYu Chen, he received his Ph.D. degree in Chongqing University, P. R. China, at 2001. Currently, he is a professor of College of Software Engineering at Chongqing University. His research interests include distributed systems, cloud computing, and embedded Linux system. He has published over 120 journal and conference papers in related research areas during recent years.



Jun Liu, he received his B.S. degree in Southwest University, P. R. China, at 2001, and M.S. degree in Chongqing University, P. R. China, at 2008. Since 2001, he has been engaged in embedded system R & D work for over 10 years. Currently he is a Ph.D. candidate in College of Computer Science, at Chongqing University. His current interests include distributed systems, machine learning, and large-scale data mining.



TianShu Wu, he received his B.S. degree in Chongqing University of Posts and Telecommunications, P. R. China, at 2011. Currently he is a full-time Ph.D. candidate in College of Computer Science, at Chongqing University. His current interests include machine learning, and large-scale data mining.