International Journal of u- and e- Service, Science and Technology
Vol.9, No. 6 (2016), pp.311-322
http://dx.doi.org/10.14257/ijunesst.2016.9.6.28

# A Short-term Prediction for QoS of Cloud Manufacturing Based on Improved Support Vector Regression Algorithm

Shumei Liu

*Hengshui University Center of Modern Educational Technology*
*Hengshui, Hebei 053000 China*
*lanxin0188@163.com*

***Abstract***

*Cloud manufacturing combined with information manufacturing, the Internet of things(IoT), cloud computing, and semantic web technology. Through extending and changing the service technology and network manufacturing, it makes the manufacturing resources and manufacturing capabilities virtualization and servicization. It can make centralized and unified intelligent management for manufacturing. Therefore, the establishment of appropriate cloud manufacturing QoS(quality of service) model is the basis and prerequisite for the development of related research of cloud manufacturing. The existing QoS model ignored the effect of time on QoS. This directly leads to the lack of accuracy of modeling, and further affects the implementation effect of the follow-up study on the prediction, service selection and so on. With cloud manufacturing evaluation criteria, this paper proposes a five dimensional QoS evaluation criteria and the corresponding calculation formula, it is consistent with the cloud manufacturing background. Secondly, we establish the QoS prediction model based on the support vector machine with arbitrary penalty band. This model can cover the shortage for the previous studies, and effectively solve the problem of the dimension of QoS prediction in cloud manufacturing. Thirdly, this paper predicts 5 important QoS indexes. The experimental data from a cloud manufacturing provider's historical data, and the experimental use a large number of historical QoS value to train the support vector machine with arbitrary penalty band. Network training is terminated when the coefficient is greater than 0.995. Then, the QoS value of the seven nodes in the future is predicted by the QoS value of the sample data set. Finally, the experiment shows that the support vector regression machine with arbitrary penalty has a good prediction effect on QoS of cloud manufacturing. But the artificial neural network and the gray forecast model has poor effect on it.*

***Keywords****: cloud manufacturing, Qos, forecasting, support vector regression*

## 1. Introduction

Cloud manufacturing combined with information manufacturing, the Internet of things [1-2], cloud computing [3-4], and semantic web technology [5]. Through extending and changing the service technology and network manufacturing, it makes the manufacturing resources and manufacturing capabilities virtualization and servicization. Therefore, the prediction of cloud manufacturing QoS has become an important research. At the same time, with the increase in the amount of cloud manufacturing services, how to predict the cloud manufacturing services based on historical transaction data becomes very important.

At present, most of the QoS related work is used the fixed value to model the various dimensions of the QoS [6-8]. According to Service running history, Jorge Cardoso [9] and T. Yu [10] uses the maximum, minimum and average value to describe the QoS attributes. E.M. Maximilien [11] describes the execution time, cost, credit, reliability and availability of the service by using the QoS vector, and each dimension of the vector is fixed value. Liangzhao Zeng [12] detailed defines three QoS properties, such as time, cost, and

ISSN: 2005-4246 IJUNESST

reliability. The response time is subdivided into processing time and delay time. The cost is subdivided into execution cost and control cost. The reliability is measured from two aspects, one is the system failure rate, the other is probability of service execution end to the error state. In addition, the standard continuous probability distribution can be used to model QoS. Literature [13] uses the standard probability distribution to describe the cloud manufacturing QoS, such as exponential distribution, normal distribution, Weibull distribution, *etc.*. Literature [14] points out that QoS negotiation can use the probability distribution function to describe the QoS protocol between service providers and service users. The author using T location scale distribution to fit the monitoring data of QoS service. Literature [15] modeling the respond time and cost of the member service as independent random variable with Beta distribution, so as to reduce the response time and cost of service combination scheme in the sense of probability.

To sum up, firstly, this paper proposes a five dimensional QoS evaluation criteria and the corresponding calculation formula. Then, we establish the QoS prediction model based on the support vector machine with arbitrary penalty band. Next, we predict 5 important QoS indexes. The experimental data from a cloud manufacturing provider's historical data, and the experimental use a large number of historical QoS value to train the support vector machine with arbitrary penalty band. Network training is terminated when the coefficient is greater than 0.995. Then, the QoS value of the seven nodes in the future is predicted by the QoS value of the sample data set.

## 2. Basic Knowledge of Cloud Manufacturing Services and Qos

With cloud manufacturing evaluation criteria, this paper proposes a five dimensional QoS evaluation criteria and the corresponding calculation formula, it is consistent with the cloud manufacturing background. Its definition and quantification are as follows:

(1) Time( $T$ ):

Time is the time consuming from the user submits the request to the service execution complete and return the result. It is influenced by the three parts, which are the online execution time $T_1$, the offline execution time $T_2$ and the time error coefficient $\delta_T$.

$$T = (T_1 + T_2) * \delta_T \tag{1}$$

Online execution time $T_1$ is the sum of the time spent on system response the request and the execution of the computer program in the virtual cloud environment.

$$T_1 = T_{pro} + T_{tra} \tag{2}$$

Where, $T_{pro}$ represents the system running time, $T_{tra}$ represents the network transmission time.

Offline execution time $T_2$ is the sum of the time consumed by the cloud service complete the manufacturing task in offline physical environment.

$$T_2 = T_{mag} + T_{wt} + T_{\log} + T_{exc} \tag{3}$$

Where, $T_{mag}$ represents the time of the service management process, the waiting time for the execution is $T_{wt}$, $T_{\log}$ represents the logistics transportation time, and $T_{exc}$ represents the task execution time.

Time error coefficient $\delta_T$ is the change trend of the actual execution time and the expected execution time of cloud services, which means that the execution time of cloud service is fluctuating.

 

$$\delta_T = \prod_{i=1}^{n}(1+\frac{T_{act}-T_{\exp}}{T_{\exp}}) \tag{4}$$

Where, $T_{act}$ represents the actual execution time, $T_{\exp}$ represents the expected execution time, and $n$ represents the total number of cloud services running.

(2) Cost( $C$ ):

Cost is the all expenses from the user submits the request to the service execution complete and return the result. It is composed of two parts, one is online expense $C_1$, and the other is the offline expense $C_2$.

$$C = C_1 + C_2 \tag{5}$$

Online expense $C_1$ is the service charge that the user needs to pay in use of cloud manufacturing system. Offline expense $C_2$ is the sum of expenses by the cloud service complete the manufacturing task in offline physical environment.

$$C_2 = C_{mag} + C_{\log} + C_{exc} \tag{6}$$

Where, $C_{mag}$ represents the spent on the service management, $C_{\log}$ represents the spent on the logistics transportation, and $C_{exc}$ represents the spent on the task execution.

(3) Availability( $Av$ ):

Availability is the time of a cloud service can provide a particular service. It is the proportion of time of successful access to the cloud service and the total time.

$$Av = \frac{t_{succ}}{t} \tag{7}$$

Where, $t_{succ}$ represents the time of successful access to cloud services.

(4) Reliability( $Rel$ ):

Reliability is the ability to perform successfully in a given time and condition. In the calculation of the reliability, we need to consider the successful execution rate $R_s$ and the average failure rate $R_f$ of cloud services.

$$R = \partial_s R_s + \frac{\partial_f}{R_f} \tag{8}$$

According to the actual situation, users set the value of $\partial_s$ and $\partial_f$. Obviously, the reliability is $0 < Rel < 1$.

(5) Reputation( $Rep$ ):

Reputation can measure the trustworthy degree of a cloud service , and it is mainly based on the experience of user using the cloud services. Different users may have different evaluation for the same cloud service.

$$Rep = \sum_{i=1}^{n}\frac{R_i}{n} \tag{9}$$

Where, $R_i$ represents the evaluation of cloud services by the $i$ th user.

In conclusion, the Qos model of each manufacturing cloud service can be expressed as:

$$Q_{CS} = (T(CS), C(CS), Av(CS), Rel(CS), Rep(CS)) \tag{10}$$

Any cloud service execution path can be composed of four basic structures, they are sequential model, parallel model, selection model and cyclic model. Therefore, we can derive Qos formula of the cloud service execution path.

 

Assuming each model contains $n$ manufacturing cloud services which are $\{CS_1, CS_2, \cdots, CS_n\}$.

(1) The sequential model.

The Qos formula is as follows:

$$\begin{cases} T_{seq} = \sum_{i=1}^{n} T(CS_i) \\ C_{seq} = \sum_{i=1}^{n} C(CS_i) \\ Av_{seq} = \prod_{i=1}^{n} Av(CS_i) \\ Rel_{seq} = \prod_{i=1}^{n} Rel(CS_i) \\ Rep_{seq} = \sum_{i=1}^{n} \frac{S(CS_i)}{n} \end{cases} \tag{11}$$

(2) The parallel model.

The $n$ programs are executed in parallel, and the Qos calculation formula is as follows:

$$\begin{cases} T_{par} = \max(T(CS_i)) \quad i \in [1, n] \\ C_{par} = \sum_{i=1}^{n} C(CS_i) \\ Av_{par} = \prod_{i=1}^{n} Av(CS_i) \\ Rel_{par} = \prod_{i=1}^{n} Rel(CS_i) \\ Rep_{par} = \sum_{i=1}^{n} \frac{S(CS_i)}{n} \end{cases} \tag{12}$$

(3) The selection model.

The probability of each program is selected is $\lambda_i$ ($\sum_{i=1}^{n} \lambda_i = 1$), and its Qos calculation formula is as follows:

$$\begin{cases} T_{sel} = \sum_{i=1}^{n} (T(CS_i) * \lambda_i) \\ C_{sel} = \sum_{i=1}^{n} (C(CS_i) * \lambda_i) \\ Av_{sel} = \sum_{i=1}^{n} (Av(CS_i) * \lambda_i) \\ Rel_{sel} = \sum_{i=1}^{n} (Rel(CS_i) * \lambda_i) \\ Rep_{sel} = \sum_{i=1}^{n} S(CS_i) * \lambda_i \end{cases} \tag{13}$$

(4) The circular model.

This model is executed $\theta$ times, and the Qos calculation formula is as follows:

$$\begin{cases} T_{cir} = \theta * \sum_{i=1}^{n} T(CS_i) \\ C_{cir} = \theta * \sum_{i=1}^{n} C(CS_i) \\ Av_{cir} = \prod_{i=1}^{n} Av(CS_i) \\ Rel_{cir} = \sum_{i=1}^{n} Rel(CS_i) \\ Rep_{sel} = \sum_{i=1}^{n} \frac{Rep(CS_i)}{n} \end{cases} \tag{14}$$

According to the above four equations, we can deduce the expression of the execution path of arbitrary cloud services.

# 3. Support Vector Machine

## 3.1. Basic Knowledge

When the feature space $H$ is Hilbert space, the $K(x,y)$ is defined as a binary function in the input space $R^n$, and the orthonormal basis of $\phi_1(x), \phi_2(x), ..., \phi_n(x)$ is Y. If $K(x,y) = \sum_{k=1}^{\infty} a_k{}^2 (\phi_k(x), \phi_k(y)), \{a_k\} \in l^2$, the $\phi(x) = \sum_{k=1}^{\infty} a_k \phi_k(x)$ is the nonlinear embedding map. Because the domain of the kernel function $K(x,y)$ is the original input space rather than the high dimensional feature space. Therefore, the method of support vector machine can cleverly avoid the operation of high dimensional inner product （$\phi(x), \phi(y))$, so as to save the computational cost. In the actual calculation, we just choose a $K(x,y)$, and do not need to reconstruct the embedding map $\phi(x) = \sum_{k=1}^{\infty} a_k \phi_k(x)$. So it is the main task to find the symmetric and non negative kernel function $K(x,y)$. There are a lot of kernel functions to meet the above conditions.

In general, the kernel function is commonly used as the linear kernel function, polynomial kernel function, the radial basis kernel function, and sigmoid kernel function. The functions are as follows:

(1) Linear kernel function

$$K(x, x_i) = x * x_i$$

(2) Polynomial kernel function

$$K(x, x_i) = [(x * x_i) + 1]^d$$

Where the $d$ is the order of the polynomial

(3) Radial basis kernel function

$$K(x, x_i) = \exp(-\|x - x_i\|^2 / 2\sigma^2)$$

Where the $\sigma$ is width of the kernel function

(4) Sigmoid kernel function

$$K(x, x_i) = \tanh(\gamma(x * x_i) + c)$$

Commonly used kernel function can be divided into two categories: one category is the global kernel function, the other is local kernel functions: The linear kernel function, polynomial kernel function, and Sigmoid kernel function is a global common kernel function

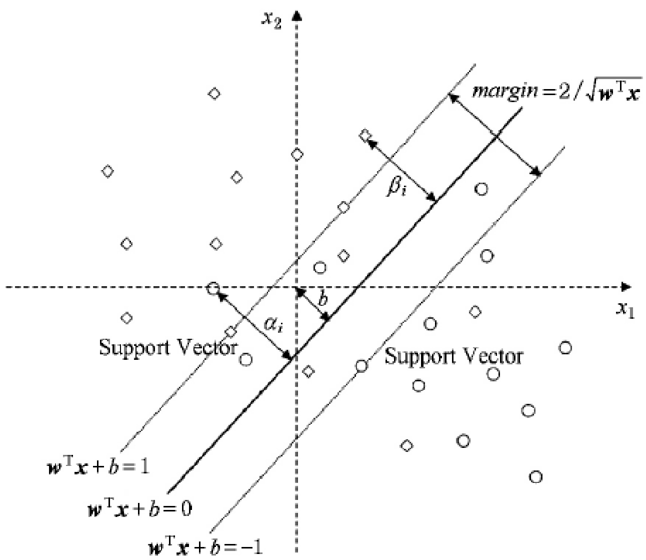


**Figure 1. The Structure of Support Vector Machine**

### 3.2. Support Vector Machine with Arbitrary Penalty Band(APB-SVM)

The traditional support vector regression algorithm transforms the objective function into a convex quadratic programming problem. In this paper, we propose to use the module $L_1$ to replace the $L_2$ module, so as to replace the convex quadratic programming by improving the linear programming. In this way, we can use the mature linear programming technique to solve the objective function of support vector regression. According to the optimization theory, there is $\omega = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i$ . So, we replace the module of original objective function $\|\omega\|_2$ by using the module $\|\alpha^{(*)}\| = \sum_{i=1}^{l}(|\alpha_i| + |\alpha_i^*|)$ . Then, the original module can be rewritten as $\|\alpha^{(*)}\| = \sum_{i=1}^{l}(\alpha_i + \alpha_i^*)$ . In the original objective function, using $\|\alpha^{(*)}\|$ instead of $\|\omega\|^2$, and put the $\omega$ into the original constraint condition. Finally, adding constraint conditions $\alpha_i, \alpha_i^* \geq 0, i = 1, 2, \cdots l$ .
Then,

$$\begin{cases} \min\limits_{\alpha^{(*)},\xi^{(*)},b} & \frac{1}{l}\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \frac{C}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ s.t. & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(x_i \cdot x_j) + b - y_i \leq \varepsilon + \xi_i \\ & y_i - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(x_i \cdot x_j) - b \leq \varepsilon + \xi_i^* \\ & \alpha_i^{(*)}, \xi_i^{(*)} \geq 0, \qquad i = 1, 2, \cdots, l \end{cases} \tag{15}$$

 

In view of the particularity of the actual problem, we can choose the appropriate penalty function. By defining the $\varepsilon$ -generalized insensitivity loss function, we have established support vector machine with arbitrary penalty band.

$$c(x,y,f(x)) = \begin{cases} y-f(x)-\varepsilon\varsigma(x), & y-f(x) > \varepsilon\varsigma(x); \\ 0, & \varepsilon\varsigma(x) \ge y-f(x) \ge -\varepsilon\varsigma^*(x); \\ |y-f(x)|-\varepsilon\varsigma^*(x), & y-f(x) < -\varepsilon\varsigma^*(x); \end{cases} \tag{16}$$

The method can transform the original optimization problem into the following form:

$$\begin{cases} \min\limits_{\alpha^{(*)},\xi^{(*)},b} & \frac{1}{l}\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) + C\cdot\left[\nu\sum_{i=1}^{l}\xi_i + \nu^*\sum_{i=1}^{l}\xi_i^* + \frac{1}{l}\sum_{i=1}^{l}(\xi_i+\xi_i^*)\right] \\ s.t. & \omega\cdot x_i + b - y_i \le \varepsilon_i\varsigma(x_i) + \xi_i \\ & y_i - \omega\cdot x_i - b \le \varepsilon_i^*\varsigma^*(x_i) + \xi_i^* \\ & \varepsilon_i^{(*)}, \xi_i^{(*)} \ge 0, \qquad i = 1,2,\cdots,l \end{cases} \tag{17}$$

## 4. The Simulation and Result Analysis

This paper predicts 5 important QoS indexes. The experimental data from a cloud manufacturing provider's historical data, the experimental use a large number of historical QoS value to train the support vector machine with arbitrary penalty band. Network training is terminated when the coefficient is greater than 0.995. Next, the QoS value of the seven nodes in the future is predicted by the QoS value of the sample data set. The 3000 historical data as our sample data set that contain time, cost, availability, reliability, and reputation. At the same time, the output values are given. It include online time and execution status. The transaction state is used to indicate whether the cloud service is successful or not. The 1 represents success, and the 0 represents failure. Table 1 shows the training sample data set, which is the data of the original data after the normalized processing. Table 2 is the training result of support vector machine.

**Table 1. The Training Sample Data Set**

| | predict-input | | | | | predict-output | |
|---|---|---|---|---|---|---|---|
| | *T* | *C* | *Av* | *Rel* | *Rep* | $T_1$ | TranS |
| 1 | 0.82 | 0.55 | 0.92 | 0.67 | 0.41 | 0.39 | 1 |
| 2 | 0.67 | 0.66 | 0.56 | 0.78 | 0.52 | 0.29 | 1 |
| 3 | 0.81 | 0.52 | 0.55 | 0.74 | 0.44 | 0.47 | 1 |
| 4 | 0.46 | 0.67 | 0.42 | 0.51 | 0.95 | 0.31 | 1 |
| 5 | 0.58 | 0.78 | 0.39 | 0.67 | 0.82 | 0.48 | 1 |
| 6 | 0.69 | 0.74 | 0.69 | 0.46 | 0.61 | 0.38 | 1 |
| 7 | 0.77 | 0.51 | 0.57 | 0.72 | 0.74 | 0.69 | 0 |
| 8 | 0.43 | 0.67 | 0.81 | 0.24 | 0.55 | 0.33 | 1 |
| 9 | 0.93 | 0.68 | 0.88 | 0.39 | 0.56 | 0.69 | 0 |
| 10 | 0.81 | 0.45 | 0.38 | 0.69 | 0.87 | 0.57 | 1 |
| 11 | 0.61 | 0.69 | 0.46 | 0.57 | 0.48 | 0.39 | 1 |
| 12 | 0.71 | 0.78 | 0.75 | 0.81 | 0.72 | 0.68 | 0 |
| 13 | 0.55 | 0.53 | 0.64 | 0.88 | 0.71 | 0.37 | 1 |
| 14 | 0.82 | 0.67 | 0.97 | 0.38 | 0.57 | 0.51 | 1 |
| 15 | 0.49 | 0.83 | 0.22 | 0.46 | 0.52 | 0.28 | 1 |
| … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| … | … | … | … | … | … | … | … |
| 2990 | 0.73 | 0.41 | 0.69 | 0.39 | 0.38 | 0.39 | 1 |
| 2991 | 0.82 | 0.51 | 0.67 | 0.69 | 0.45 | 0.71 | 0 |
| 2992 | 0.65 | 0.89 | 0.81 | 0.57 | 0.73 | 0.52 | 1 |
| 2993 | 0.58 | 0.48 | 0.83 | 0.81 | 0.49 | 0.41 | 1 |
| 2994 | 0.74 | 0.67 | 0.38 | 0.39 | 0.56 | 0.58 | 1 |
| 2995 | 0.72 | 0.36 | 0.46 | 0.69 | 0.45 | 0.48 | 1 |
| 2996 | 0.47 | 0.76 | 0.72 | 0.57 | 0.72 | 0.26 | 1 |
| 2997 | 0.65 | 0.73 | 0.24 | 0.81 | 0.41 | 0.39 | 1 |
| 2998 | 0.38 | 0.49 | 0.39 | 0.88 | 0.74 | 0.29 | 1 |
| 2999 | 0.32 | 0.76 | 0.52 | 0.38 | 0.78 | 0.17 | 1 |
| 3000 | 0.82 | 0.63 | 0.52 | 0.46 | 0.75 | 0.71 | 0 |

**Table 2. SVR Regression Analysis**

| | | |
|---|---|---|
| Kernel function type | RBF | |
| Parameter setting | | |
| Degree | 3 | |
| Gamma | 0.5 | |
| Coef() | 0.001 | |
| Eps | 0.001 | |
| C | 1 | |
| nu | 0.5 | |
| Shrinking | 1 | |
| p | 0.01 | |
| probability | 1 | |
| coefficient | | |
| project | rho | Prob. |
| Const | -0.4281 | 0.1295 |

| $\alpha$ | SV1 | SV2 | SV3 | SV4 | SV5 |
|---|---|---|---|---|---|
| -1 | -1 | -0.9902 | -1 | -0.9531 | -0.8401 |
| -1 | -0.885 | -1 | -0.8848 | -1 | -0.1221 |
| … | … | … | … | … | … |
| … | … | … | … | … | … |
| 0.5101 | 1 | 1 | 1 | 1 | -0.2878 |

| **Number** | **Actual value** | **Fitted value** |
|---|---|---|
| 1 | 0.39 | 0.382 |
| 2 | 0.29 | 0.292 |
| 3 | 0.47 | 0.477 |
| 4 | 0.31 | 0.319 |
| 5 | 0.48 | 0.489 |
| 6 | 0.38 | 0.389 |
| 7 | 0.69 | 0.681 |
| 8 | 0.33 | 0.327 |
| 9 | 0.69 | 0.695 |
| 10 | 0.57 | 0.571 |

| | | |
|---|---|---|
| 11 | 0.39 | 0.399 |
| 12 | 0.68 | 0.678 |
| 13 | 0.37 | 0.375 |
| 14 | 0.51 | 0.518 |
| 15 | 0.28 | 0.288 |
| … | … | … |
| … | … | … |
| … | … | … |
| 2990 | 0.39 | 0.379 |
| 2991 | 0.71 | 0.702 |
| 2992 | 0.52 | 0.532 |
| 2993 | 0.41 | 0.416 |
| 2994 | 0.58 | 0.588 |
| 2995 | 0.48 | 0.468 |
| 2996 | 0.26 | 0.266 |
| 2997 | 0.39 | 0.379 |
| 2998 | 0.29 | 0.294 |
| 2999 | 0.17 | 0.174 |
| 3000 | 0.71 | 0.716 |
| correlation index | 0.99769 | |
| coefficient of determination | 0.99538 | |

In the results of support vector machine training, Rho is equivalent to the constant term in the regression equation, $\alpha$ is equivalent to the regression coefficient of the regression equation, and SV is the support vector. From the results of SVR regression analysis, the coefficient of determination is 0.99538, the result is higher than the linear regression coefficient. Therefore, the fitting effect is better.

In this way, the training of support vector machine with arbitrary penalty band support vector regression is terminated. Next, we will predict the online time and execution status of the next seven nodes based on the sample data set. Table 3 gives a comparison between the predicted value and the actual value. It can be seen that the model of this article can be used to predict the service quality well.

**Table 3. Comparison between Predicted Value and Actual Value**

| | $T_1$ | | TranS | |
|---|---|---|---|---|
| | **predicted value** | **actual value** | **predicted value** | **actual value** |
| 3001 | 0.702 | 0.696 | 0.001 | 0 |
| 3002 | 0.517 | 0.516 | 0.998 | 1 |
| 3003 | 0.698 | 0.701 | 0.002 | 0 |
| 3004 | 0.309 | 0.311 | 0.997 | 1 |
| 3005 | 0.706 | 0.702 | 0.005 | 0 |
| 3006 | 0.459 | 0.469 | 0.997 | 1 |
| 3007 | 0.419 | 0.425 | 0.993 | 1 |

Next, we compare the predicted results of online time $T_1$ of this article's method with several other classical forecasting models, as shown in Figure 2.

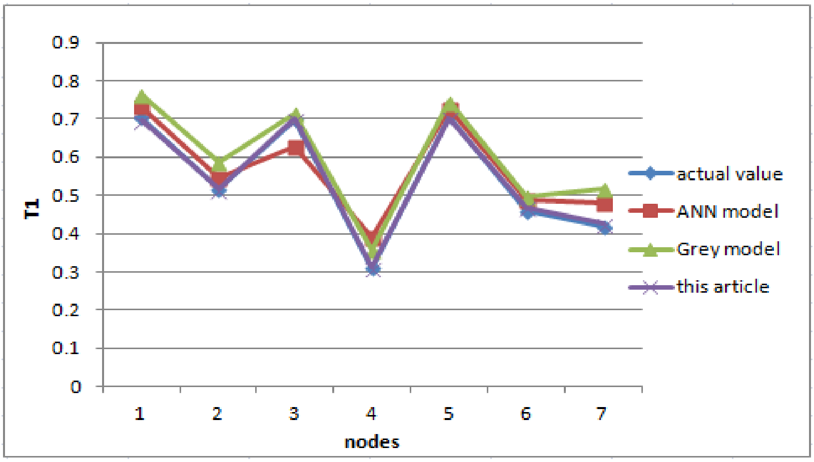


**Figure 2. Comparison between Predicted Value and Actual Value of Online Time**

Through the simulation results, we can see that the purple curve of the predicted values is almost coincident with the purple one which is the curve of the actual value. This shows that the support vector regression machine with arbitrary penalty has a good prediction effect on QoS of cloud manufacturing. But the artificial neural network and the gray forecast model has poor effect on it. In addition, this study is aimed at short-term prediction of Qos. Long term forecasting is the need to consider the mutation of all the indicators. We will study it in future research.

## 5. Conclusion

Because of the dynamic, uncertainty and some other reasons of the Internet, the service quality of the cloud services is constantly changing. Therefore, this paper proposes a five dimensional QoS evaluation criteria and the corresponding calculation formula. Then, we establish the QoS prediction model based on the support vector machine with arbitrary penalty band. Next, we predict 5 important QoS indexes. Then, the QoS value of the seven nodes in the future is predicted by the QoS value of the sample data set. Finally, the experiment shows that the support vector regression machine with arbitrary penalty has a good prediction effect on QoS of cloud manufacturing. But the artificial neural network and the gray forecast model has poor effect on it.

## References

[1] L. Zhiguo, “Real time material management system based on Internet of things technology”, [J].Mobile communications, vol. 35, no. 5, **(2011)**, pp. 32-34.
[2] W. Baoyun, “Overview of Internet of things technology”, [J].Journal of electronic measurement and instrument, vol. 23, no. 12, **(2009)**, pp. 55-57.
[3] L. Bohu, “Cloud manufacturingCloud Computing in the manufacturing sector”, [J].China manufacturing industry information, no. 5, **(2011)**, pp. 24-26.
[4] Z. Yuan, “Research on cloud computing technology and its application”, [J], Scientific and technological information, vol. 25, no. 26, **(2011)**.
[5] Y. Sheng, L. Fei and Y. Chao, “Integrated service model and semantic description of outsourcing processing resources in cloud manufacturing environment”, [J].Computer integrated manufacturing system, vol. 17, no. 3, **(2011)**, pp. 525-532.
[6] S. Ran, “A Model for Web Services Discovery with QoS”, [J]. SIGccom Exch, vol. 4, no. 1, **(2003)**, pp. 1-10,
[7] Y. Liu, A. H. Ngu and L. Z. Qos, “computation and policing in dynamic web service selection”, [C]. In Proceedings of the 13th International World Wide Web Conference, **(2004)**, pp. 66-73.
[8] R. Jurca. B. Faltings and W. Binder, “Reliable qos monitoring based on client feedback”, [C]. In Proceedings of the 16th International World Wide Web Conference, **(2007)**, pp.1003-1012.

[9] J. Cardoso, A. Shetli, J. Miller, J. Arnold and K. Kochut, "Quality of Service for Workflows and Web Service Processes", [J]. Web Semantics, vol. 1, no. 3, **(2004)**, pp. 281-308.
[10] T. Yu and K. I. Lin, "Service selection algorithms for web services with end-to-end qos constraints", [J]. Information Systems and E-Business Management, vol. 3, no. 2, **(2005)**, pp. 103.
[11] E. M. Maximilien and M. P. Singh, "Toward autonomic web services trust and selection", [C]. In Proceedings of the 2nd International Conference on Service Oriented Computing, **(2004)**, pp. 212-221.
[12] L. Zeng, B. Benatallah, A. H.H. Ngu, M. Dumas. J. Kalagnanam and H. Chang, "QoS-Aware Middleware for Web Services Composition", [J], IEEE Transactions on Software Engineering, vol. 30, no. 5, **(2004)**, pp. 311-327.
[13] A. Alichlmayr. F. Roscenberg, P. Leitncr and S. Dustdar, "Comprehensive qos monitoring of veb services and event-based sla violation detection", [C]. In Proceedings of the 4th International Workshop on Middleware for Service Oriented Computing, **(2009)**, pp. 1-6.
[14] S. Rosario, A. Benveniste. S. Haar and C. Jard, "Probabilistic qos and soft contracts for transaction-based web services orchestrations", [J]. IEEE Transactions on Services Computing, vol. 1, no. 4, **(2008)**, pp. 187,
[15] W. Wiesemann, R. Hochreiter and D. Kuhn, "A stochastic programming approach for qos-aware service composition", [C]. In Proceedings of the 8th IEEE/ACM International Symposium on Cluster Computing and the Grid, **(2008)**, pp. 226-233.