Received September 3, 2015, accepted October 5, 2015, date of publication November 6, 2015, date of current version November 19, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2498191

Web Service QoS Prediction Based on Adaptive Dynamic Programming Using Fuzzy Neural Networks for Cloud Services

XIONG LUO, YIXUAN LV, RUIXING LI, AND YI CHEN
School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China

Corresponding author: X. Luo (xluo@ustb.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities, in part by the National Natural Science Foundation of China under Grant 61174103, Grant 61272357, and Grant 61300074, and in part by the National Key Technologies Research and Development Program of China under Grant 2015BAK38B01.

ABSTRACT Recently, more and more traditional services are being migrated into a cloud computing environment that makes the quality of service (QoS) becomes an important factor for service selection and optimal service composition while forming cross-cloud service applications. Considering the nonlinear and dynamic property of QoS data, it is so difficult to achieve dynamic prediction while designing a QoS prediction method with unsatisfactory prediction accuracy. It is thus desirable to explore how to design an effective approach by incorporating some intelligent techniques into the QoS prediction method to improve prediction performance. In this paper, motivated by the adaptive critic design and Q-learning technique, we propose a novel QoS prediction approach to serve this purpose through the combination of fuzzy neural networks and adaptive dynamic programming (ADP), i.e., an online learning scheme. This approach extracts fuzzy rules from QoS data and employs the ADP method to parameter learning of the fuzzy rules. Moreover, we provide a convergence boundedness result for our proposed approach to guarantee the stability. Experimental results on a large-scale QoS service data set verify the prediction accuracy of our proposed approach.

INDEX TERMS Quality of service (QoS), QoS prediction, fuzzy neural network, adaptive dynamic programming, cloud services.

I. INTRODUCTION

Recent years have seen the massive migration of traditional services to the cloud computing environment. These cloud services on different cloud platforms are often invoked in web services either deployed in the cloud by companies or deployed in accordance with cloud applications. It is clearly crucial to form cross-cloud service applications. There has been a tremendous surge of interest on web services in cloud environment [1]–[4]. Nowadays, these ever-increasing web services could provide more functions and features in cloud environment to enhance the user experience [1]. Within this trend, the performance of cloud services can be improved when constructing the personalized cloud applications through the integration of web services in cloud environment. Meanwhile, it poses a challenge in selecting qualified web services for developing web service-based integration cloud applications. The quality of service (QoS) is becoming more and more crucial in describing web services during the implementation of service selection and optimal service composition. Generally, QoS is a set of nonfunctional performance indices of web services, including popularity, response time, failure probability, and many others [5], [6]. Also, more and more attention has been paid to QoS which has been seen as a very hot topic in cloud computing [7], [8]. Among those available performance indices, response time that users can feel directly is one of the most important QoS properties used to evaluate web services. The predicted QoS values have been widely used in service selection, service ranking, and service composition under dynamic environment [9]–[11]. Here, we discuss the prediction method for response time in QoS.

QoS is fundamental for cloud users, who expect providers to offer services of the advertised quality, and for service providers, who need to find the right tradeoffs between

QoS levels and operational costs [12]. The predicted response time is valuable for service composition, component selection, task scheduling, and so on. At present, a number of QoS prediction approaches for web services have been developed. Generally speaking, the QoS prediction approaches can be decomposed in two families. The first one is the static prediction method. These methods (e.g. arithmetic average value method) are simple and easy to implement, ignoring the individual factors of users, the service status, and the network environment. These prediction methods do not reflect the nonlinear relationship of the QoS data [13], which may lead to a great error between the predicted value and the actual value. The second one is the dynamic prediction method, e.g. prediction via collaborative filtering, similarity measure, multi-dimensional weighting [14]–[19]. The main idea behind these methods is that web users who have similar historical QoS data on some services, would have similar experiences on other services. But the difficulty in getting sufficient QoS data for similarity calculation may lead to a large error under some cloud computing environment.

Considering the nonlinear and dynamic property of QoS data as well as the lack of QoS data for calculation in some cases, we present a prediction approach based on adaptive dynamic programming (ADP) to avoid the above limitations. ADP is a novel optimization technique which combines concepts of reinforcement learning (e.g. Q-learning) and dynamic programming [20], [21]. While dealing with complex systems, ADP uses a function approximation structure such as neural network (NN) to approximate cost function, which avoids the curse of dimensionality and reduces the computation time. The approximate optimal solution is obtained by using the offline iteration or the online update algorithms. In other words, ADP can learn from a dynamic environment with less information. The application of ADP is focused on nonlinear and complex systems, such as aircraft system [22], power system [23], energy management system [24], and many others [25], [26]. In view of this, due to its potential scalability of the adaptive critic designs and the intuitiveness of Q-learning, ADP may play an important role in dealing with the nonlinear prediction for the QoS data.

Meanwhile, we propose to use fuzzy logic to express the inner relationship of the QoS data that users who have similar QoS data on some services would have similar experiences on other services. Fuzzy logic can deal with the QoS of web service in context-aware environment, which can be expressed with linguistic variables and membership functions. Thus, fuzzy logic has been used in QoS management system [27] and QoS prediction with fuzzy clustering [28]. But, on the other hand, fuzzy logic lacks the learning ability and the adaptability while extracting fuzzy rules from QoS data. Then, it can be combined with NN to avoid such limitation. By combining fuzzy logic with NN, fuzzy neural network (FNN) provides a mathematical tool to deal with the nonlinear and dynamic property of QoS data in context-aware environment. Specifically, FNN not only extracts fuzzy rules from the data to express the dynamic property of the system, but also simply adjusts the parameters of the network during the learning process to enhance the adaptability of the network [29]. Therefore, FNN shows great adaptability and flexibility in developing a nonlinear model for QoS prediction.

In consideration of all the issues in QoS prediction mentioned above, we may conduct a QoS prediction though the combination of FNN and ADP. In our previous research work of [30], by using FNN, the direct heuristic dynamic programming as a special ADP was presented in a longitudinal control of hypersonic vehicles. Although the control system framework was implemented, there is only a single action output under its scheme. Moreover, there is no stability proof related to the convergence of proposed ADP approach, which limited its application. Motivated by it, the purpose of this paper is to present a novel QoS prediction service scheme which fuses ADP with FNN. Within the proposed scheme, FNN is employed to approximate the cost-to-go function and the action network in ADP. In addition to novel QoS prediction scheme, we also provide a associated Lyapunov stability proof and its convergence result to guarantee a uniform ultimate boundedness (UUB) property for the weight errors of NN in the proposed FNN-based ADP scheme. These results are better than the previous one in [30]. Therefore, due to its unique features of ADP as well as strong adaptability and flexibility of FNN in context-aware environment, our proposed scheme may serve the purpose of effectively predicting those QoS data for cloud services.

The rest of this paper is organized as follows. Section 2 presents the background of QoS prediction. Section 3 describes the characteristics of FNN. Section 4 presents the architecture and the convergence analysis of the FNN-based ADP prediction algorithm. The QoS prediction experiments using our proposed scheme are illustrated in section 5. Finally, section 6 provides a conclusion.

II. PROBLEM STATEMENT

Since many web services have the similar functions with cloud applications, consumers are not possible to use every web service. Therefore, QoS data of unused web services plays an important role in providing suitable web services to consumers.

TABLE 1. A user-component matrix.

	C_1	C_2	C_3	C_4
U_1	0.92		0.32	
U_2	0.44	0.71		
U_3			0.22	0.59
U_4		0.64		1.13

Let us first consider an example in Table 1. Each entry presents the response time of a web service invoked by a user, which is a particular QoS property value. The numeric entries in the table mean the response time for users who have invoked the services. The blank entries correspond to the users who have not invoked the services.

For the purpose of service composition or service ranking, we need to know all the QoS data of web services related to users. However, the real QoS data is similar to the example we present here. Therefore, it is significant to predict the blank entries before any QoS-based service selection or service ranking. As a result, the problem we study in this paper is how to precisely predict the blank entries in accordance with the existing entries.

III. FUZZY NEURAL NETWORK

With the development of intelligent techniques, fuzzy logic and NN have made remarkable achievements in their respective fields. With the improvement of the fuzzy theory and the new upsurge of NN research in 1980s, many researchers paid more attention to the fusion of those two fields. They utilized the superiority of fuzzy theory and NN, forming the concept of FNN.

Adaptive-network-based fuzzy inference system (ANFIS) is a NN implementation of Takagi-Suguno (T-S) fuzzy inference system [31]. ANFIS constructs a set of fuzzy if-then rules automatically and optimizes the rules through the self-learning of NN. This avoids such limitation that fuzzy if-then rules mainly come from the experience of experts. Compared with back propagate (BP) network, ANFIS has stronger self-learning ability, robustness, and adaptability while it can approximate any nonlinear function.

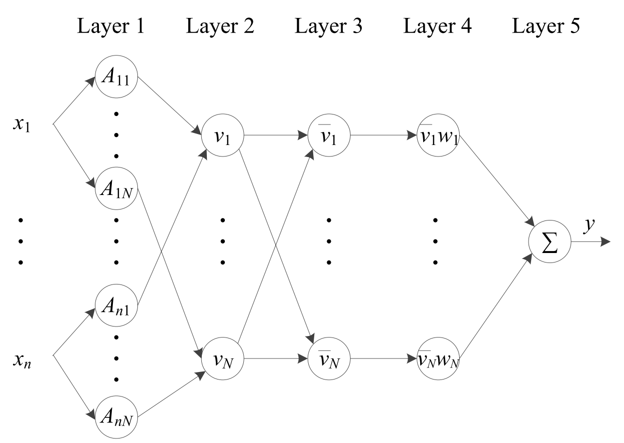


FIGURE 1. ANFIS architecture.

Fig. 1 shows an architecture of ANFIS. The ANFIS consists of five layers. Here, we assume that ANFIS has n inputs (i.e., $x_1, \cdots, x_n$) and 1 output (i.e., y). There are N fuzzy sets. Suppose that the rule base contains N fuzzy if-then rules of T-S type:

$$\begin{aligned} &\text{Rule } j: \text{ IF } x_1 = A_{1j} \text{ and} \cdots \text{and } x_n = A_{nj}, \\ &\quad \text{THEN } w_j = \sum_{i=1}^{n} z_{ij} x_i + z_{n+1,j}, \\ &z_{1j}, \cdots, z_{n+1,j} \in \mathbb{R}; \quad j = 1, 2, \cdots, N. \end{aligned}$$

Here, A_{ij} represents the fuzzy variable where $i = 1, \cdots, n$ and $j = 1, \cdots, N$.

Layer 1 is the fuzzification layer. This layer defines the membership function of the nodes. Normally the function is a Guassian function:

$$\mu_{A_{ij}}(x_i) = \exp\left\{ -\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2} \right\}. \tag{1}$$

where c_{ij} and σ_{ij} are the excepted value and standard deviation of Gaussian function, respectively.

Layer 2 is the rule layer. Each node multiplies those inputs and generates the outputs as follows:

$$v_j = \prod_{i=1}^{n} \mu_{A_{ij}}(x_i). \tag{2}$$

The outputs of this layer stand for the firing strength of a rule.

Layer 3 is the normalization layer. Nodes of layer 3 generate the weights according to:

$$\bar{v}_j = \frac{v_j}{\sum_{j=1}^{N} v_j}. \tag{3}$$

Layer 4 is the defuzzification layer. Nodes in this layer are calculated by:

$$\bar{v}_j w_j = \bar{v}_j \left(\sum_{i=1}^{n} z_{ij} x_i + z_{n+1,j} \right). \tag{4}$$

Layer 5 is the output layer. The node stands for the ANFIS output y by calculating the sum of outputs of defuzzification layer.

$$y = \sum_{j=1}^{N} \bar{v}_j w_j. \tag{5}$$

Due to the combination of BP algorithm and least square estimation (LSE) algorithm in its implementation, ANFIS achieves good performance with fast learning speed. In the training process, the parameters of if-then rules can be identified via LSE in the forward pass. In the backward pass, the error rates propagate backward while the expectations and variances of Guassian function are updated via BP algorithm. Therefore it is more accurate and efficient through the use of that hybrid algorithm.

IV. QoS PREDICTION SCHEME

In the past decades, the ADP method was proposed through the use of NN to approximate the cost function. According to the critic's output, the ADP can be categorized as: heuristic dynamic programming (HDP), dual heuristic programming (DHP), and globalized dual heuristic programming (GDHP) [21], [32]. The action dependent version of HDP and DHP are formed when the critic's inputs are augmented with the controller's output. In this paper, our approach is proposed on the basis of HDP.

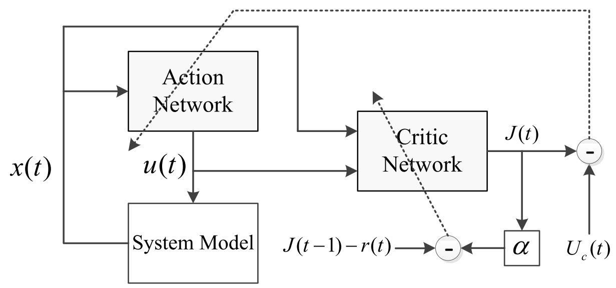


FIGURE 2. ADP architecture.

A. ADP STRUCTURE

As shown in Fig. 2, this ADP is structured to estimate the cost function $J(t)$ in the Bellman equation of dynamic programming:

$$J(t) = \sum_{k=t+1}^{\infty} \alpha^{k-t-1} r(t), \tag{6}$$

where α is a discount factor bounded within (0, 1) and $r(t)$ is the external reinforcement signal. In addition, $x(t)$ is the state of the plant and $u(t)$ is the control signal of the plant at time t.

Through the use of the critic network of ADP, the J is calculated as an approximator of the optimal value function J^*. It is implemented by minimizing the following error measure over time:

$$E_c(t) = \frac{1}{2} \times e_c^2(t), \tag{7}$$

$$e_c(t) = \alpha J(t) - (J(t-1) - r(t)). \tag{8}$$

Meanwhile, the action network of ADP is to backpropagate the error between the desired objective U_c and the approximator J^*. Since we have defined "0" as the reinforcement signal for "success" in our proposed approach, U_c is set to 0 all the time without loss of generality. Then, by minimizing the following performance error measure, the weights in the action network can be updated.

$$E_a(t) = \frac{1}{2} \times e_a^2(t), \tag{9}$$

$$e_a(t) = J(t) - U_c(t). \tag{10}$$

B. QoS PREDICTION BASED ON ADP USING ANFIS

Our proposed approach aims to predict the missing QoS value of web service mentioned in Section 2. To deal with the nonlinear and dynamic properties of QoS data in context-aware environment, the ADP using FNN is proposed to enhance the adaptability and flexibility of Internet. Unlike the traditional actor-critic design in ADP normally using feedforward networks with one hidden layer to implement the critic network and the action network, our proposed approach employs FNN to construct those two networks.

The online training and optimization algorithm in ADP with ANFIS is shown in Algorithm 1. In traditional ADP, $x(t)$ is the state of the plant and $u(t)$ is the control signal of the plant at time t. In our proposed approach, $x(t)$ stands for a vector of remaining QoS data used to predict a blank entry at time t. And $u(t)$ is the predictive result of the blank entry at time t.

Algorithm 1 The ADP Using ANFIS

1 initialize $t = 0$, $x(0)$, and $u(0)$;
2 apply $x(t)$ to the action network and obtain $u(t)$;
3 apply $u(t)$ to the system model and obtain $x(t+1)$;
4 apply $x(t+1)$ and $u(t)$ to the critic network, and obtain $J(t)$;
5 update weights of the critic network till the requirement on $E_c(t)$ is satisfied;
6 update weights of the action network till the requirement on $E_a(t)$ is satisfied;
7 $t = t + 1$, continue from 2.

C. REDESCRIPTION OF ADP USING ANFIS

To analyze the convergence of the ADP using ANFIS, we simplify the structure of ANFIS as illustrated in Figs. 3 and 4. Layer 1 and layer 2 in ANFIS are converted to the input layer of NN. The output layer of NN consists of the layer 3, layer 4 and layer 5 in ANFIS.

In the simple ANFIS, p_k and q_k are the input and output of the hidden layer. The subscripts "a" and "c" mean the action network and critic network, respectively. In addition, $c(t)$ and $\sigma(t)$ stand for the expectation and variance of Guassian function, respectively. And $\phi(\cdot)$ represents a function defined as:

$$\phi(x) = \exp\left\{-\frac{1}{2}x\right\}. \tag{11}$$

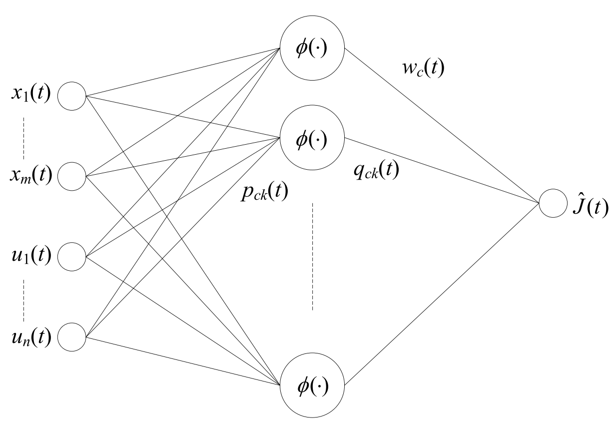


FIGURE 3. Schematic diagram of ANFIS in critic network.

According to Fig. 3 and Fig. 4, the inputs of the critic network include m states $x(t) = (x_1(t), \cdots, x_m(t))^T$ and n actions $u(t) = (u_1(t), \cdots, u_n(t))^T$. And $w_c(t)$ is the parameter matrix of fuzzy rules in ANFIS. Applying the computing rule of exponential function to the combination of layer 1 and layer 2 in ANFIS, the $p_c(t)$ can be written as follows.

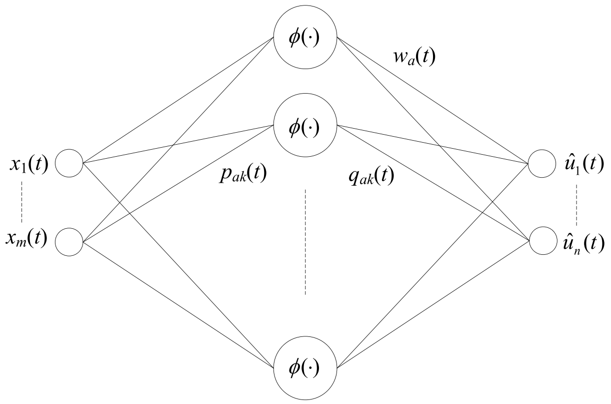


FIGURE 4. Schematic diagram of ANFIS in action network.

Let N_{fc} be the numbers of fuzzy rules in the critic network. The output $\widehat{J}(t)$ can be expressed as:

$$\widehat{J}(t) = \frac{\sum_{k=1}^{N_{fc}} w_{c_k}(t) q_{c_k}(t)}{\sum_{k=1}^{N_{fc}} q_{c_k}(t)}, \tag{12}$$

$$q_{c_k}(t) = \phi(p_{c_k}(t)), \tag{13}$$

$$p_{c_k}(t) = \sum_{i=1}^{m} \left(\frac{x_i(t) - c_{c_{i,k}}(t)}{\sigma_{c_{i,k}}(t)} \right)^2 + \sum_{j=1}^{n} \left(\frac{u_j(t) - c_{c_{j+m,k}}(t)}{\sigma_{c_{j+m,k}}(t)} \right)^2, \quad k = 1, \cdots, N_{fc} \tag{14}$$

Unlike the traditional action network, the inputs include m states $x(t) = (x_1(t), \cdots, x_m(t))^T$. Let N_{fa} be the numbers of fuzzy rules in the action network. The output $\widehat{u}(t)$ can be expressed as:

$$\widehat{u}_j(t) = \frac{\sum_{k=1}^{N_{fa}} w_{a_{jk}}(t) q_{a_k}(t)}{\sum_{k=1}^{N_{fa}} q_{a_k}(t)} \quad j = 1, \cdots, n \tag{15}$$

$$q_{a_k}(t) = \phi(p_{a_k}(t)), \tag{16}$$

$$p_{a_k}(t) = \sum_{i=1}^{m} \left(\frac{x_i(t) - c_{a_{i,k}}(t)}{\sigma_{a_{i,k}}(t)} \right)^2, \quad k = 1, \cdots, N_{fa} \tag{17}$$

D. CONVERGENCE ANALYSIS

In this section, we refer to a UUB result in [33] and provide a convergence bound for our proposed approach to show that the estimation errors of the network weights in this approach are UUB.

According to (13) and (16), $\phi_c(t) = (q_{c_1}(t), \cdots, q_{c_{N_{fc}}}(t))^T$ and $\phi_a(t) = (q_{a_1}(t), \cdots, q_{a_{N_{fa}}}(t))^T$ are the hidden layer neuron activation function vectors of the critic network and the action network, respectively. The output layer weights $w_a(t)$ and $w_c(t)$ are initialized randomly and updated during learning. Then, with (12) and (15), $\widehat{J}(t)$ and $\widehat{u}(t)$ can be written with matrix forms:

$$\widehat{J}(t) = \widehat{w}_c^T(t)\phi_c(t), \tag{18}$$

$$\widehat{u}(t) = \widehat{w}_a^T(t)\phi_a(t). \tag{19}$$

The weight of critic network is updated as:

$$\begin{aligned}\widehat{w}_c^T(t+1) &= \widehat{w}_c(t) - l_c \frac{\partial E_c(t)}{\partial \widehat{J}(t)} \frac{\partial \widehat{J}(t)}{\partial \widehat{w}_c(t)} \\ &= \widehat{w}_c(t) - \alpha l_c \phi_c(t) \\ &\quad \times [\alpha \widehat{w}_c^T(t)\phi_c(t) + r(t) - \widehat{w}_c^T(t-1)\phi_c(t-1)]^T\end{aligned} \tag{20}$$

where $l_c > 0$ is the learning rate.

The weight of action network is updated as:

$$\begin{aligned}\widehat{w}_a^T(t+1) &= \widehat{w}_a(t) - l_a \frac{\partial E_a(t)}{\partial \widehat{J}(t)} \frac{\partial \widehat{J}(t)}{\partial \widehat{u}(t)} \frac{\partial \widehat{u}(t)}{\partial \widehat{w}_a(t)} \\ &= \widehat{w}_a(t) - l_a \phi_a(t) [\widehat{w}_c^T(t) C(t)] [\widehat{w}_c^T(t)\phi_c(t)]^T\end{aligned} \tag{21}$$

where $l_a > 0$ is the learning rate, and

$$C(t) = \left(\left(-\phi_{ck}(t) \frac{u_j(t) - c_{c_{j+m,k}}(t)}{\sigma^2_{c_{c_{j+m,k}}}(t)} \right)_{k=1}^{N_{fc}} \right)_{j=1}^{n}. \tag{22}$$

Let w^* be the optimal weight of network. Then we denote the weight estimation error as $\widetilde{w}(t) \overset{def}{=} \widehat{w}(t) - w^*$. In the following analysis, $\| \cdot \|$ represents the 2-norm.

Definition 1: The solution $\widetilde{w}(t)$ is said to be uniformly ultimately bounded (UUB) within a bound $\varepsilon > 0$, if for any $\delta > 0$ and $t_0 > 0$, there exists a positive number $\underline{N} = \underline{N}(\delta, \varepsilon)$, independent of t_0, so that $\| \widetilde{w}(t) \| \leqslant \varepsilon$ for all $t \geqslant \underline{N} + t_0$ when $\| \widetilde{w}(t_0) \| < \delta$.

Then, we will adopt a Lyapunov stability analysis framework to prove the UUB property of our proposed approach. First, we present some assumptions as follows.

Assumption 1: Let w_c^* and w_a^* be the optimal weights for the critic and action network, respectively. They are bounded. Then,

$$\| w_c^* \| \leqslant w_{cm}, \quad \| w_a^* \| \leqslant w_{am}, \tag{23}$$

where w_c^* and w_a^* are two positive constants.

Theorem 1: Let Assumption 1 hold. Take the reinforcement signal as $r(t) \in [0, 1]$. Let the critic and action network settings be (18) and (19), respectively. Let the weight update for the critic and action network be (20) and (21), respectively. Then the errors of network weights, $\widetilde{w}_c(t)$ and $\widetilde{w}_a(t)$, are UUB, provided the following conditions hold:

$$\frac{\sqrt{2}}{3} < \alpha < 1, \ 0 < l_c < \frac{1}{\alpha^2 \| \phi_c(t) \|^2}, \ 0 < l_a < \frac{1}{\| \phi_a(t) \|^2}, \tag{24}$$

Proof: See Appendix A.

Theorem 2: Let Assumption 1 hold. Then, the errors of network weights, $\widetilde{w}_c(t)$ and $\widetilde{w}_a(t)$, are UUB, provided the following conditions hold:

$$\frac{\sqrt{2}}{3} < \alpha < 1, \quad 0 < l_c < \frac{1}{\alpha^2 N_{fc}}, \; 0 < l_a < \frac{1}{N_{fa}}. \qquad (25)$$

where N_{fc} and N_{fa} are the numbers of fuzzy rules in the critic and action network, respectively.

Proof: See Appendix B.

While using our approach, Theorem 2 gives a simple sufficient condition for selecting learning rates in the proposed ADP architecture to maintain stability of the weight updates.

V. EXPERIMENTS

A. DATA SET DESCRIPTION

In our experiments, the data set comes from the WS-DREAM [34], [35] including the QoS data of 5, 825 real-world web services from 73 countries. The QoS value is observed by 339 distributed computers and each computer invokes all the 5, 825 web services by sending null operating requests. In this paper, we use a 339 × 5825 user-component matrix which represents response time values for experiment. The statistics of the web service QoS response time data set is summarized in Table 2.

TABLE 2. The statistics of data set.

Statistics	Value
Scale of response time	0-20 s
Mean of response time	0.910 s
Number of users	339
Number of web services	5, 825

B. METRICS

We calculate mean absolute error (MAE) and mean absolute percentage error (MAPE) to compare the prediction quality of our proposed approach with other methods.

The MAE and MAPE are defined as follows:

$$\text{MAE} = \frac{1}{P}\sum_{i,j}\left|\widehat{v}_{ij} - v_{ij}\right|, \qquad (26)$$

$$\text{MAPE} = \frac{1}{P}\sum_{i,j}\left|\frac{\widehat{v}_{ij} - v_{ij}}{v_{ij}}\right|, \qquad (27)$$

where v_{ij} is the actual QoS value of web service item j invoked by service user i, $\widehat{v}_{ij}$ is the prediction value of each method, and P is the number of the predicted QoS values.

C. PERFORMANCE COMPARISON

In this section, the above metrics are compared between our proposed approach and the following popular prediction methods.

TABLE 3. Performance comparison.

Matrix density	Metrics	Average	UPCC	IPCC	UIPCC	ADP using ANFIS
10%	MAE	1.3417	1.2583	1.2574	1.2580	**1.1036**
	MAPE	1.6197	0.7375	0.7351	0.7364	**0.4926**
20%	MAE	1.0840	1.1589	1.1572	1.1582	**1.0106**
	MAPE	1.5539	0.7636	0.7726	0.7653	**0.5073**
30%	MAE	1.0882	1.0882	1.1013	1.0880	**0.9004**
	MAPE	1.5253	0.8033	0.8782	0.8166	**0.4718**
40%	MAE	1.1598	1.2091	1.2489	1.2151	**1.0143**
	MAPE	1.3582	0.8355	0.9437	0.8509	**0.4850**
50%	MAE	1.1648	1.1986	1.2335	1.1940	**0.9862**
	MAPE	1.5917	0.8688	1.0408	0.8739	**0.4954**
60%	MAE	1.1771	1.2726	1.2049	1.2795	**1.0442**
	MAPE	1.2828	0.9027	1.0356	0.9346	**0.4988**
70%	MAE	1.2671	1.1946	1.1805	1.1872	**0.9532**
	MAPE	1.7827	0.9316	0.9059	0.9082	**0.4987**
80%	MAE	1.2201	1.2708	1.2789	1.2612	**1.0167**
	MAPE	1.6088	0.9423	0.9980	0.9324	**0.5077**

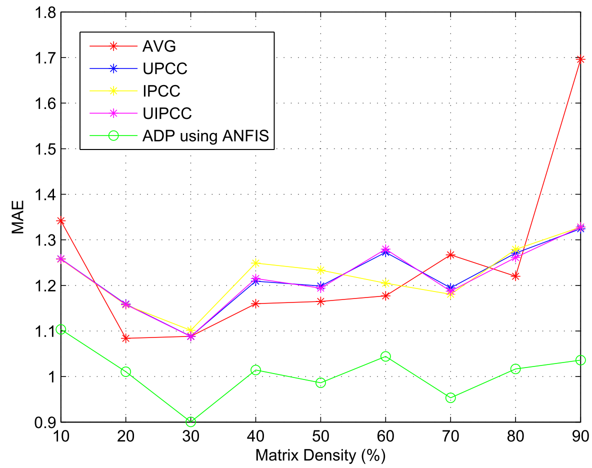


FIGURE 5. MAE of response time prediction.

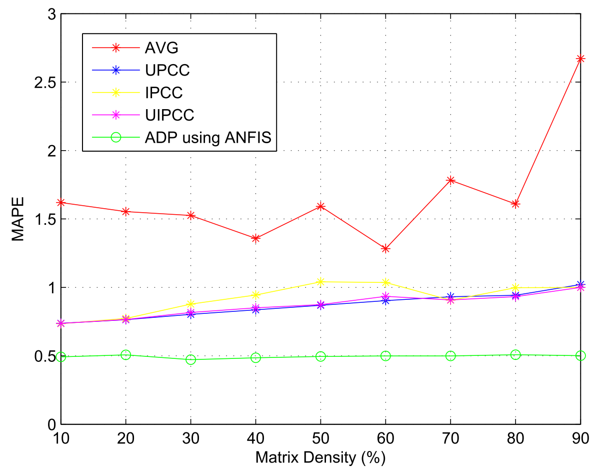


FIGURE 6. MAPE of response time prediction.

- Arithmetic average value method (AVG). This method is defined as

$$\text{average} = \frac{\sum_{i=1}^{N}\widehat{v}_{ij}}{N_{\text{exist}}} \qquad (28)$$

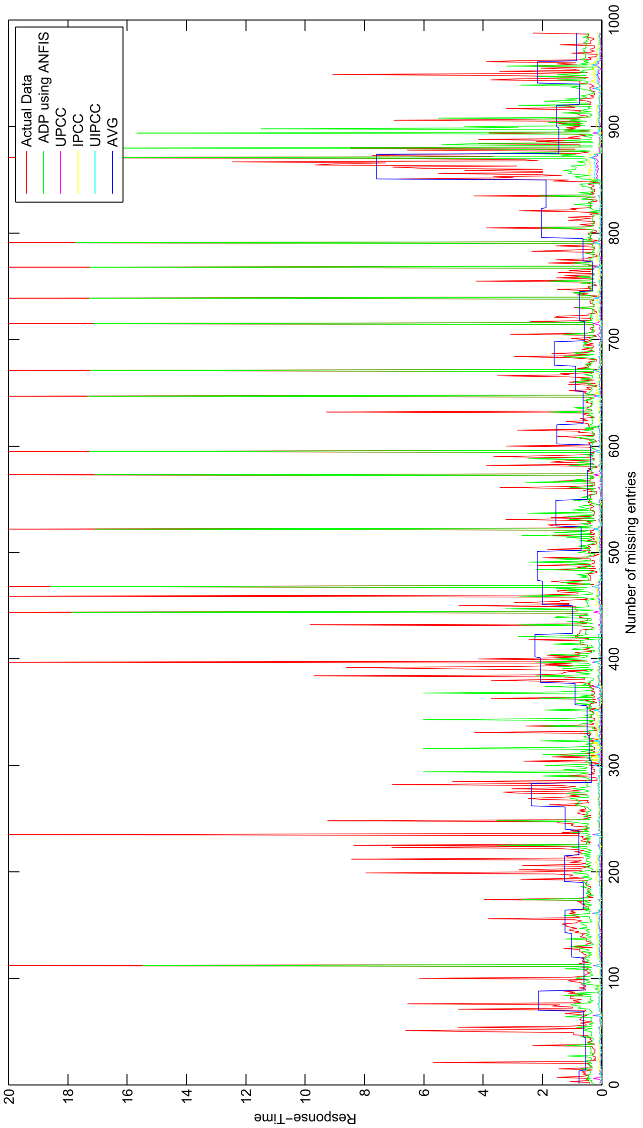


FIGURE 7. **Experimental result with 70% matrix density.**

- where $\widehat{v}_{ij}$ is the actual QoS response time value that the user i invokes the web service j, N is the number of all users, N_{exist} is the number of the remaining users in the matrix who invoke web service j.
- User-based collaborative filtering method using Pearson correlation coefficient (UPCC). This method predicts the missing QoS data of users based on the collected information from similar users [15], [17].
- Item-based collaborative filtering method using Pearson correlation coefficient (IPCC). This method predicts the missing QoS data of users based on the collected information from similar service items [14].
- UIPCC. This method combines UPCC with IPCC for missing QoS data prediction [16].

In our experiments, the number of the fuzzy rules in both the critic and the action networks are set to $N_{fc} = N_{fa} = 3$. According to Theorem 2, the discount factor is set to $\alpha = 0.95 \in (\frac{\sqrt{2}}{3}, 1)$, and the learning rates of the critic and the action networks should satisfy $l_c < \frac{1}{0.95^2 \times 3} \approx 0.369$ and $l_a < \frac{1}{3} \approx 0.333$, respectively. Then we choose the learning rate as $l_c = l_a = 0.3$. In our experiment, we find that this common fixed learning rate will result in slow convergence rate of ADP. So we use the learning rates $l_c = l_a = 0.3$ in the initial training phase of NN. And they should decrease to a value during the learning process. Here, the learning rates are decreased by 0.05 every 50 times until they reach 0.005.

In order to compare the performance of each approach, we set different sparsity to randomly remove some entries from the matrix. For instance, 10% means that 10% entries are removed randomly, and the remaining entries are used to predict the missing entries.

The experimental results are show in Table 3. From this table, we can find that our proposed approach has better performance (smaller MAE and MAPE values) than other methods with different matrix densities, especially with high matrix density. Figs. 5 and 6 show the experimental results of response time prediction. From those figures, we conclude that the performance of our proposed approach is stable as the performance of other methods becomes worse with the increase of the matrix density. An experimental result with 70% matrix density is shown in Fig. 7, which shows the differences of the performance in reflecting the trend of QoS data among these methods. Due to its potential scalability of the adaptive critic designs and the intuitiveness of Q-learning in ADP as well as strong adaptability and flexibility of FNN in context-aware environment, our proposed scheme has its unique features in dealing with those QoS data. Compared with other methods, the improvement of our proposed approach verifies that the idea involved in the fusion of ADP and FNN for QoS prediction is effective.

VI. CONCLUSION

QoS evaluates the web services while playing an important role in web service selection and service recommendation for cloud applications. In this paper, we present a novel QoS prediction scheme through the fusion of FNN and ADP. Considering the nonlinear and dynamic property of QoS data, we construct FNN in the critic as well as the action network under ADP architecture, and develop the weight updating method to adjust the parameters of the networks with the purpose of optimizing a set of fuzzy if-then rules. In addition, we provide the stability analysis for our proposed approach to guarantee the convergence of scheme. These convergence results are better than the previous one, which provide a guideline for the selection of parameters used in our FNN-based ADP scheme. Experiments are conducted on a large-scale real-world QoS service data set to predict the response time in QoS. Compared with other traditional methods, the experimental results shows the proposed approach improves the accuracy of prediction for response time in QoS.

APPENDIX A
PROOF OF THEOREM 1

Let $\zeta_c(t)$ and $\zeta_a(t)$ be the approximation errors of the critic and the action network outputs, respectively. Then,

$$\zeta_c(t) = (\widehat{w}_c(t) - w_c^*)^{\mathrm{T}}\phi_c(t) = \widetilde{w}_c^{\mathrm{T}}(t)\phi_c(t), \quad \text{(A.29)}$$

$$\zeta_a(t) = (\widehat{w}_a(t) - w_a^*)^{\mathrm{T}}\phi_a(t) = \widetilde{w}_a^{\mathrm{T}}(t)\phi_a(t). \quad \text{(A.30)}$$

Select the Lyapunov function as:

$$V(t) = V_1(t) + V_2(t) + V_3(t), \quad \text{(A.31)}$$

where

$$V_1(t) = \frac{1}{l_c}\mathrm{tr}\big[\widetilde{w}_c^{\mathrm{T}}(t)\widetilde{w}_c(t)\big], \quad \text{(A.32)}$$

$$V_2(t) = \frac{1}{\gamma l_a}\mathrm{tr}\big[\widetilde{w}_a^{\mathrm{T}}(t)\widetilde{w}_a(t)\big], \quad \text{(A.33)}$$

$$V_3(t) = \frac{2}{9}\|\zeta_c(t-1)\|^2. \quad \text{(A.34)}$$

where $\gamma > 0$ is a weighting factor.

The first difference of $V_1(t)$ is given by:

$$\begin{aligned}\triangle V_1(t) = &-\alpha^2\|\zeta_c(t)\|^2 - \alpha^2\big(1 - l_c\alpha^2\|\phi_c(t)\|^2\big)\\ &\times\Big\|\zeta_c(t) + (w_c^*)^T\phi_c(t) + \frac{1}{\alpha}r(t)\\ &\quad - \frac{1}{\alpha}\widehat{w}_c^T(t-1)\phi_c(t-1)\Big\|^2\\ &+\big\|\alpha(w_c^*)^{\mathrm{T}}\phi_c(t) + r(t) - \widehat{w}_c^T(t-1)\phi_c(t-1)\big\|^2.\end{aligned} \quad \text{(A.35)}$$

By applying the Cauchy-Schwarz inequality for (A.35), we have

$$\begin{aligned}\triangle V_1(t) \leqslant &-\alpha^2\|\zeta_c(t)\|^2 - \alpha^2\big(1 - l_c\alpha^2\|\phi_c(t)\|^2\big)\\ &\times\Big\|\zeta_c(t) + (w_c^*)^T\phi_c(t) + \frac{1}{\alpha}r(t)\\ &\quad - \frac{1}{\alpha}\widehat{w}_c^T(t-1)\phi_c(t-1)\Big\|^2\\ &+2\Big\|\alpha(w_c^*)^{\mathrm{T}}\phi_c(t) + r(t) - \frac{2}{3}\widehat{w}_c^T(t-1)\phi_c(t-1)\\ &\quad -\frac{1}{3}(w_c^*)^T\phi_c(t-1)\Big\|^2\\ &+\frac{2}{9}\|\zeta_c(t-1)\|^2.\end{aligned} \quad \text{(A.36)}$$

The first difference of $V_2(t)$ is given by:

$$\begin{aligned}\triangle V_2(t) = \frac{1}{\gamma}\Big[&-\Big[\|\widehat{w}_c^T(t)C(t)\|^2 - l_a\|\widehat{w}_c^T(t)C(t)\|^2\|\phi_a(t)\|^2\Big]\\ &\times\|\widehat{w}_c^T(t)\phi_c(t)\|^2 - \|\widehat{w}_c^T(t)C(t)\|^2\|\zeta_a(t)\|^2\\ &+\|\widehat{w}_c^T(t)\phi_c(t) - \widehat{w}_c^T(t)C(t)\zeta_a(t)\|^2\Big].\end{aligned} \quad \text{(A.37)}$$

By applying the Cauchy-Schwarz inequality for (A.37), we have

$$\begin{aligned}\triangle V_2(t) \leqslant \frac{1}{\gamma}\Big[&-\big(1 - l_a\|\phi_a(t)\|^2\big)\|\widehat{w}_c^T(t)C(t)\|^2\|\widehat{w}_c^T\phi_c(t)\|^2\\ &+4\|(w_c^*)^T\phi_c(t)\|^2 + \|\widehat{w}_c^T(t)C(t)\|^2\|\zeta_a(t)\|^2\\ &+4\|\zeta_c(t)\|^2\Big].\end{aligned} \quad \text{(A.38)}$$

The first difference of $V_3(t)$ is given by:

$$\triangle V_3(t) = \frac{2}{9}\left(\|\zeta_c(t)\|^2 - \|\zeta_c(t-1)\|^2\right). \quad \text{(A.39)}$$

Substituting (A.36), (A.38), and (A.39) into the first difference of $V(t)$, we have

$$\begin{aligned}\triangle V(t) &= \triangle V_1(t) + \triangle V_2(t) + \triangle V_3(t)\\ &\leqslant -\left(\alpha^2 - \frac{2}{9} - \frac{4}{\gamma}\right)\|\zeta_c(t)\|^2 - \alpha^2\big(1 - l_c\alpha^2\|\phi_c(t)\|^2\big)\\ &\quad\times\Big\|\zeta_c(t) + (w_c^*)^T\phi_c(t) + \frac{1}{\alpha}r(t)\\ &\qquad - \frac{1}{\alpha}\widehat{w}_c^T(t-1)\phi_c(t-1)\Big\|^2\\ &\quad-\frac{1}{\gamma}(1 - l_a\|\phi_a(t)\|^2)\|\widehat{w}_c^T(t)C(t)\|^2\|\widehat{w}_c^T(t)\phi_c(t)\|^2\\ &\quad+Z^2(t),\end{aligned} \quad \text{(A.40)}$$

where

$$\begin{aligned}Z^2(t) = 2&\Big\|\alpha(w_c^*)^T\phi_c(t) + r(t) - \frac{2}{3}\widehat{w}_c^T(t-1)\phi_c(t-1)\\ &-\frac{1}{3}(w_c^*)^T\phi_c(t-1)\Big\|^2 + \frac{1}{\gamma}\|\widehat{w}_c^T(t)C(t)\|^2\|\zeta_a(t)\|^2\\ &+\frac{4}{\gamma}\|(w_c^*)^T\phi_c(t)\|^2.\end{aligned} \quad \text{(A.41)}$$

Applying the Cauchy-Schwarz inequality for (A.41) and using Assumption 1, we have

$$\begin{aligned}Z^2(t) &\leqslant 8\Big(\alpha^2\|(w_c^*)^T\phi_c(t)\|^2 + r^2(t)\\ &\quad+\frac{4}{9}\|\widehat{w}_c^T(t-1)\phi_c(t-1)\|^2 + \frac{1}{9}\|(w_c^*)^T\phi_c(t-1)\|^2\Big)\\ &\quad+\frac{2}{\gamma}\|\widehat{w}_c^T(t)C(t)\|^2\Big(\|\widehat{w}_a^T(t)\phi_a(t)\|^2+\|(w_a^*)^T\phi_a(t)\|^2\Big)\\ &\quad+\frac{4}{\gamma}\|(w_c^*)^T\phi_c(t)\|^2\\ &\leqslant \left(8\alpha^2 + \frac{40}{9} + \frac{4}{\gamma}\right)w_{cm}^2\phi_{cm}^2\\ &\quad+\frac{4}{\gamma}w_{cm}^2C_m^2w_{am}^2\phi_{am}^2 + 8r_m^2 = Z_m^2,\end{aligned} \quad \text{(A.42)}$$

where w_{cm}, w_{am}, ϕ_{cm}, ϕ_{am}, C_m, and r_m are the upper bounds of w_c^*, w_a^*, $\phi_c(t)$, $\phi_a(t)$, $C(t)$, and $r(t)$, respectively.

According to (A.40), choose

$$\frac{\sqrt{2}}{3} < \alpha < 1,\ 0 < l_c < \frac{1}{\alpha^2\ \|\phi_c(t)\|^2},\ 0 < l_a < \frac{1}{\|\phi_a(t)\|^2}, \quad \text{(A.43)}$$

and select γ satisfying

$$\gamma > \frac{4}{\alpha^2 - \frac{2}{9}}, \quad \text{(A.44)}$$

then for any

$$\|\zeta_c(t)\| > \frac{Z_m}{\sqrt{\alpha^2 - \frac{2}{9} - \frac{4}{\gamma}}}, \quad \text{(A.45)}$$

the first difference $\triangle V(t) \leqslant 0$ holds.

Using the Lyapunov theorem, this implies that the NN weight estimation errors $\widetilde{w}_c(t)$ and $\widetilde{w}_a(t)$ are UUB. □

APPENDIX B
PROOF OF THEOREM 2

According to the definition of function in (11), each element of activation function vectors $\phi_c(t)$ and $\phi_a(t)$, i.e., $q_{c_1}(t), \cdots, q_{c_{N_{fc}}}(t), q_{a_1}(t), \cdots, q_{a_{N_{fa}}}(t)$, is bounded within (0, 1]. Then, we have

$$0 < l_c < \frac{1}{\alpha^2 N_{fc}} \leqslant \frac{1}{\alpha^2 \sum_{k=1}^{N_{fc}} (q_{c_k}(t))^2} = \frac{1}{\alpha^2 \|\phi_c(t)\|^2}, \tag{B.46}$$

$$0 < l_a < \frac{1}{N_{fa}} \leqslant \frac{1}{\sum_{k=1}^{N_{fa}} (q_{a_k}(t))^2} = \frac{1}{\|\phi_a(t)\|^2}. \tag{B.47}$$

According to the judging conditions (24) in Theorem 1, the estimate errors of network weights $\widetilde{w}_c(t)$ and $\widetilde{w}_a(t)$ are UUB. $\square$

REFERENCES

[1] T. H. Noor, Q. Z. Sheng, A. H. H. Ngu, and S. Dustdar, "Analysis of Web-scale cloud services," *IEEE Internet Comput.*, vol. 18, no. 4, pp. 55–61, Jul./Aug. 2014.

[2] V. Chang, "The business intelligence as a service in the cloud," *Future Generat. Comput. Syst.*, vol. 37, pp. 512–534, Jul. 2014.

[3] C. Gravier, J. Subercaze, A. Najjar, F. Laforest, X. Serpaggi, and O. Boissier, "Context awareness as a service for cloud resource optimization," *IEEE Internet Comput.*, vol. 19, no. 1, pp. 28–34, Jan./Feb. 2015.

[4] R. Ranjan, J. Kolodziej, L. Wang, and A. Y. Zomaya, "Cross-layer cloud resource configuration selection in the big data era," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 16–22, May/Jun. 2015.

[5] D. A. Menascé, "QoS issues in Web services," *IEEE Trans. Internet Comput.*, vol. 6, no. 6, pp. 72–75, Nov./Dec. 2002.

[6] D. A. Menascé, "Composing Web services: A QoS view," *IEEE Trans. Internet Comput.*, vol. 8, no. 6, pp. 88–90, Nov./Dec. 2004.

[7] S. Misra, P. V. Krishna, K. Kalaiselvan, V. Saritha, and M. S. Obaidat, "Learning automata-based QoS framework for cloud IaaS," *IEEE Trans. Netw. Serv. Manage.*, vol. 11, no. 1, pp. 15–24, Mar. 2014.

[8] C. Mao, J. Chen, D. Towey, J. Chen, and X. Xie, "Search-based QoS ranking prediction for Web services in cloud environments," *Future Generat. Comput. Syst.*, vol. 50, pp. 111–126, May 2014.

[9] H. Zheng, W. Zhao, J. Yang, and A. Bouguettaya, "QoS analysis for Web service compositions with complex structures," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 373–386, Jul./Sep. 2013.

[10] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu, "Web service recommendation via exploiting location and QoS information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1913–1924, Jul. 2014.

[11] S.-Y. Hwang, C.-C. Hsu, and C.-H. Lee, "Service selection for Web services with probabilistic QoS," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 467–480, May/Jun. 2015.

[12] D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing: Modeling techniques and their applications," *J. Internet Services Appl.*, vol. 5, no. 1, p. 11, 2014.

[13] A. Metzger, C.-H. Chi, Y. Engel, and A. Marconi, "Research challenges on online service quality prediction for proactive adaptation," in *Proc. Workshop Eur. Softw. Services Syst. Res.-Results Challenges (S-Cube)*, Zürich, Switzerland, 2012, pp. 51–57.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, New York, NY, USA, 1994, pp. 175–186.

[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, San Francisco, CA, USA, 1998, pp. 43–52.

[16] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, 2007, pp. 39–46.

[17] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services*, Salt Lake City, UT, USA, Jul. 2007, pp. 439–446.

[18] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Jul./Sep. 2013.

[19] L. Qi, J. Ni, X. Xia, H. Wang, and C. Yan, "A multi-dimensional weighting method for historical records in cloud service evaluation," in *Proc. IEEE 4th Int. Conf. Big Data Cloud Comput.*, Sydney, NSW, Australia, Dec. 2014, pp. 265–266.

[20] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 32, no. 2, pp. 140–153, May 2002.

[21] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, Eds., *Handbook of Learning and Approximate Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2004.

[22] S. N. Balakrishnan and V. Biega, "Adaptive-critic-based neural networks for aircraft optimal control" *J. Guid., Control, Dyn.*, vol. 19, no. 4, pp. 893–898, Jul./Aug. 1996.

[23] Y. Tang, H. He, J. Wen, and J. Liu, "Power system stability control for a wind farm based on adaptive dynamic programming," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 166–177, Jan. 2015.

[24] Q. Wei, D. Liu, G. Shi, and Y. Liu, "Multibattery optimal coordination control for home energy management systems via distributed iterative adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 62, no. 7, pp. 4203–4214, Jul. 2015.

[25] M. Fairbank, D. Prokhorov, and E. Alonso, "Clipping in neurocontrol by adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1909–1920, Oct. 2014.

[26] J. Wang and R. Y. K. Fung, "Adaptive dynamic programming algorithms for sequential appointment scheduling with patient preferences," *Artif. Intell. Med.*, vol. 63, no. 1, pp. 33–40, Jan. 2015.

[27] F. Xia, W. Zhao, Y. Sun, and Y.-C. Tian, "Fuzzy logic control based QoS management in wireless sensor/actuator networks," *Sensors*, vol. 7, no. 12, pp. 3179–3191, 2007.

[28] M. Zhang, X. Liu, R. Zhang, and H. Sun, "A Web service recommendation approach based on QoS prediction using fuzzy clustering," in *Proc. IEEE 9th Int. Conf. Services Comput.*, Honolulu, HI, USA, Jun. 2012, pp. 138–145.

[29] S.-I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 801–806, Sep. 1992.

[30] X. Luo, Y. Chen, J. Si, and F. Liu, "Longitudinal control of hypersonic vehicles based on direct heuristic dynamic programming using ANFIS," in *Proc. Int. Joint Conf. Neural Netw.*, Beijing, China, 2014, pp. 3685–3692.

[31] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

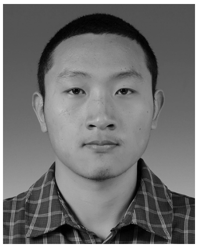
[32] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[33] F. Liu, J. Sun, J. Si, W. Guo, and S. Mei, "A boundedness result for the direct heuristic dynamic programming," *Neural Netw.*, vol. 32, pp. 229–235, Aug. 2012.

[34] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proc. IEEE Int. Conf. Web Services*, Miami, FL, USA, Jul. 2010, pp. 83–90.

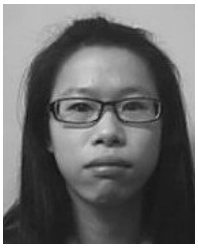
[35] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proc. 30th IEEE Symp. Rel. Distrib. Syst.*, Madrid, Spain, Oct. 2011, pp. 1–10.

XIONG LUO received the Ph.D. degree from Central South University, China, in 2004. From 2012 to 2013, he was with the School of Electrical, Computer and Energy Engineering, Arizona State University, USA, as a Visiting Scholar. He is currently an Associate Professor with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include machine learning, cloud computing, cyber-physical systems, and computational intelligence.



YIXUAN LV is currently pursuing the master's degree with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include cyber-physical systems and adaptive dynamic programming.

RUIXING LI is currently pursuing the master's degree with the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, China. Her research interests include cloud computing and computational intelligence.



YI CHEN received the master's degree from the University of Science and Technology Beijing, China, in 2014. Her research interests include fuzzy learning and cyber-physical systems.