IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 7, 2019, accepted March 22, 2019, date of publication March 26, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907546

Web Services Classification Based on Wide & Bi-LSTM Model

HONGFAN YE, BUQING CAO, ZHENLIAN PENG, TING CHEN, YIPING WEN, AND JIANXUN LIU
School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China
Corresponding author: Buqing Cao (buqingcao@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873316, Grant 61872139, and Grant 61702181, in part by the Natural Science Foundation of Hunan Province under Grant 2017JJ2098, Grant 2018JJ2136, and Grant 2018JJ2139, and in part by the Educational Commission of Hunan Province of China under Grant 17C0642.

ABSTRACT With the rapid growth of Web services on the Internet, it becomes a great challenge for Web services discovery. Classifying Web services with similar functions is an effective method for service discovery and management. However, the functional description documents of Web services usually are short in their length, with sparse features and less information, which makes most topic models unable to model the short text well, consequently affecting the Web service classification. To solve this problem, a Web service classification method based on Wide & Bi-LSTM model is proposed in this paper. In this method, first, all the discrete features in the description documents of Web services are combined to perform the breadth prediction of Web service category by exploiting the wide learning model. Second, the word order and context information of the words in the description documents of Web services are mined by using the Bi-LSTM model to perform the depth prediction of the Web service category. Third, it uses the linear regression algorithm to integrate the breadth and depth prediction results of Web service categories as the final result of the service classification. Finally, compared with six Web service classification methods based on TF-IDF, LDA, WE-LDA, LSTM, Wide&Deep, and Bi-LSTM, respectively, the experimental results show that our approach achieves a better performance in the accuracy of Web service classification.

INDEX TERMS Wide learning model, Bi-LSTM model, linear regression, web service classification.

I. INTRODUCTION

Web Service is one of the most important implementation technologies in Service-Oriented Architecture (SOA) [1], and it is also an online application service that an organization publishes to perform certain functions, through the internet other consumers can access and use it. With the rapid development of the Internet, a variety of Web services is gradually known and applied, and the myriad of Web services is released every day to meet the needs of people. It is very necessary to manage Web services through applications which are based Web services such as service discovery [2], [3], service composition [4], [5], and so on. Hence, how to find suitable Web services effectively has become one of the core problems in the field of service-oriented computing. Classifying Web services with similar functions can greatly reduce the searching space of Web services and so effectively facilitate Web service discovery [6], [7], [12]. Therefore, effective Web services classification is a prerequisite for solving Web service discovery [8], [18].

At present, there have been a lot of researches on Web service classification, among which they mainly focus on functional attribute-based Web service recommendation [9], [21], [22], [25], [28] and Web service classification [10]. According to the existing researches, function description text of Web service usually has short-length, sparse-features and less-information, similar to short text [19]. The main problem of this short text classification is how to structure the short text in a form that can be understood by computer. Some researchers proposed many methods for the above problems. For example, the LDA (Latent Dirichlet Allocation) [11], [30] topic model or its extended topic model [4], [5], [16], [17] is used to extract implicit topic information from Web services, in order to represent Web services by the low-dimensional topic vector features, then calculate the similarity among Web services

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Ardagna.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

based on these topic vectors and finally classify Web services. The generated feature matrix on top of this short text usually is sparse. It is not accurate to measure the similarity between Web services only according to the degree of co-occurrence of words in the short text by using the traditional methods to model the function description text of Web service. As a result, some topic models add auxiliary information in the process of modeling [24], such as word clustering information [5], label information [22], tagging information [17], which are beneficial to extract more accurate implicit semantic information of Web services. We also observe the method based on semantic web, in which OWL is used for describing the web services, in order to facilitate Web service discovery [29]. Meanwhile, some approaches exploit services' content and network [23], [20], or QoS [9], [10], [31] to improve service recommendations. Moreover, LSTM (Long Short-Term Memory) model is applied to Web service recommendation and prediction [22], [26]. These researches improve the accuracy of Web service classification to some extent, but most of them ignore the deep mining of hidden information (word order between words, context information, and so on) in Web service description text, which can be exploited to promote the accuracy of service classification.

To address the above problems, an enhanced Wide & Deep model (called Wide & Bi-LSTM model) for Web service classification is proposed in this paper. This model performs the processes of memorization and generalization by joint training. The Wide model uses nonlinear feature transformations to memorize the sparse feature interactions of the function description of Web services automatically and selectively, and captures their key features. The Bi-LSTM model captures the feature interactions in the function description of Web services through low dimensional embedding, and obtains some generalized service features by producing more general and abstract representations of Web services. Based on this process, through the complementary advantages of Wide component and Bi-LSTM component, the hidden information in the function description texts of Web services can be fully mined. As we investigated, this is the first work integrates the wide learning and Bi-LSTM as a novel Wide&Bi-LSTM model to perform Web service classification. Aiming to the problem of the short, sparse features in the functional description documents of Web services, this model not only captures more informative features and their interactions of words, but also mines the word order and context information among words, which can be used as the important supplementary information to facilitate Web service classification. More concretely, the contributions of this paper are summarized as follows:

- We present a Wide & Bi-LSTM model, which performs the processes of memorization and generalization through joint training to fully mine the historical correlation features, and deeper level service feature interaction in the Web service description texts in order to achieve more accurate services' similarity measurement.
- We exploit the Wide & Bi-LSTM model to model and predict service features of the function description texts of Web services. The breadth prediction of Web service category is performed based on the wide learning model by combining the discrete features in the description documents of Web services. The depth prediction of Web service category is achieved by mining the word order and context information of the words in the description documents of Web services by using the Bi-LSTM model. The breadth and depth prediction results of Web service categories are integrated as the final result of service classification.
- Based on the real dataset from ProgrammableWeb, we conduct experimental comparison and analysis to verify the model and method proposed in this paper. Compared with six Web service classification methods based on TF-IDF, LDA, WE-LDA, LSTM, Wide&Deep and Bi-LSTM respectively, the experimental results show that our approach is better than other existing methods in the accuracy of Web services classification.

The remainder of this paper is organized as follows: Section I describes the background and summarizes the main work of this paper. Section II presents an overview on related work. The detailed introductions of the proposed model and method are presented in Section III. We show the experimental results and analysis in Section IV. Finally, this paper is concluded in Section V.

II. RELATED WORK

With the development of the service computing and the cloud computing, a variety of Web services are emerging. Thus, the discovery of Web services has become a hot research topic. As we know, Web service classification has been proved as an effective method to improve the performance of Web service discovery [7]. Some researchers have done a lot of works on classification or clustering of Web services, and a large part of them focuses on Web service classification based on functional attributes.

The functionality-based Web service clustering is based on the function similarity of Web services and classifies them into different clusters with similar function. For example, Yu et al. [11] proposed a service community learning algorithm based on joint clustering of services and operations, and clustered services with similar functions into homogeneous service communities. In the literatures [7], [12], many methods based on mining WSDL document features were also widely used. These methods firstly extract key features of WSDL document, such as description text, type, port, and Web service name. Then, based on the extracted feature information, the similarity between the Web services was calculated by cosine similarity or other computation method. According to the similarity, Web services were clustered. Cristina et al. [13] presented an ant colony-based method to perform Web services clustering based on semantic similarity. Furthermore, there were also many ways to exploit auxiliary ancillary information to improve the accuracy of

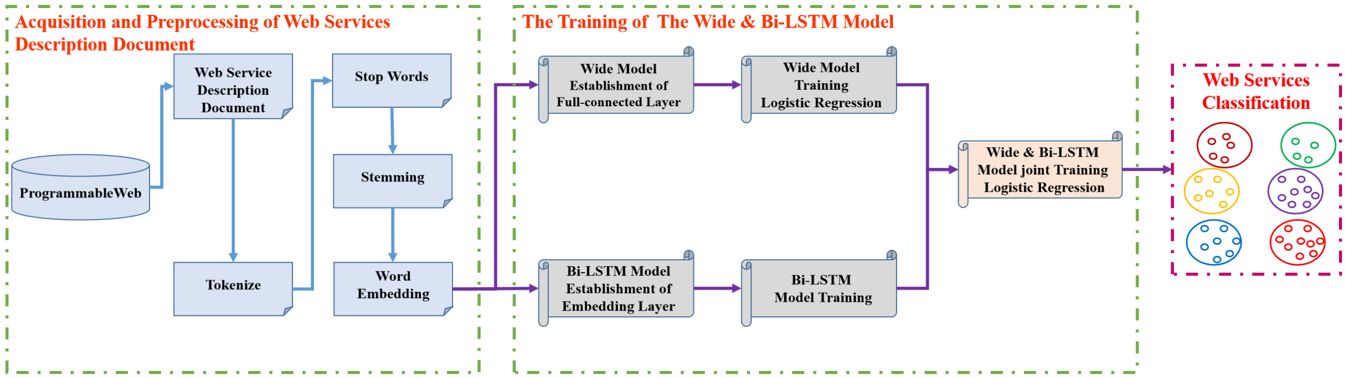

FIGURE 1. Overall framework of the proposed method.

Web services clustering. For example, Wu et al. [14] designed a hybrid Web service tag recommendation strategy, called as WSTRec, which reduce the impact of label distribution imbalance and irrelevant tags to Web service clustering. This method took into account the co-occurrence of tags in the label recommendation process by using tag mining techniques and semantic correlation measurement techniques simultaneously. Cao et al. [20] proposed a K-Means algorithm based on the similarity of Mashup services to cluster services by combining the description document and the corresponding tags of Mashup services. Sotiropoulos et al. [15] presented a service clustering model based on topic probability and domain characteristics. Shi et al. [16] provided an enhanced-LDA service clustering method based on word vectors to cluster all the words in the Web service description document, and to take these words clustering information into the training process of the LDA model. The above these methods took into account the problems of short-length, limited-corpus and sparse-features in Web service description documents, and introduced auxiliary information (such as word clustering information, label information, and so on) into the process of service corpus training [27]. It is helpful to extract more accurate service implicit semantic information and promote Web service classification. However, most of them did not take into account the fine-grained semantic information in service description documents, such as word order between words and contextual information, which are very important to improve the accuracy of service classification.

Google developed a Wide & Deep model based on TensorFlow for classification and regression on June 2016 [21]. In this model, wide linear component can selectively memorize sparse feature interactions by using cross-product feature transformation, and deep neural network component can generate unseen feature interactions through low-dimensional embeddings. Therefore, the model can perform text classification and prediction by combining the memorization ability of linear model and the generalization ability of deep neural network model. However, the deep neural network module in this model only uses the simple feed-forward neural network built on the advanced machine learning Api estimator (tf.estimator) based on TensorFlow, which do not fully exploit the word order and contextual information of the text. To further mine word order and context information of text, Wang et al. [22] proposed an online QoS prediction method that uses the LSTM model to learn and predict the future reliability of the service system. Although the LSTM model mines word order and context information to some extent, it only considers the historical context information (i.e. the positive word order information), and cannot consider the future context information (i.e. the reverse word order information). To solve this problem, in our proposed model and method, we exploit the Bi-LSTM model to simultaneously model the positive and reverse word order, and use the wide learning model to combine the discrete features which include context information in the description documents of Web services, which facilitate Web service classification.

III. CLASSIFICATION OF WEB SERVICES BASED ON WIDE & BI-LSTM MODEL

The overall framework of the proposed method in this paper is shown in Figure 1, which consists of three parts, that are the acquisition and preprocessing of Web service description document, the training of Wide & Bi-LSTM model and Web service classification. In the process of acquisition and preprocessing of Web Services description document, the description text of the Web service and other related information are crawled from the ProgrammableWeb, in which the corresponding feature columns are extracted to build feature vector matrix. In the training process of Wide & Bi-LSTM model, the obtained feature vector matrices of Web service description document are trained respectively using Wide and Bi-LSTM components to derive their memorization classification vector and generalization classification vector. The breadth and depth prediction are performed respectively based on Wide & Bi-LSTM model. In the process of Web service classification, the joint training and classification prediction is implemented, and the linear regression algorithm is performed to integrate the breadth and depth prediction

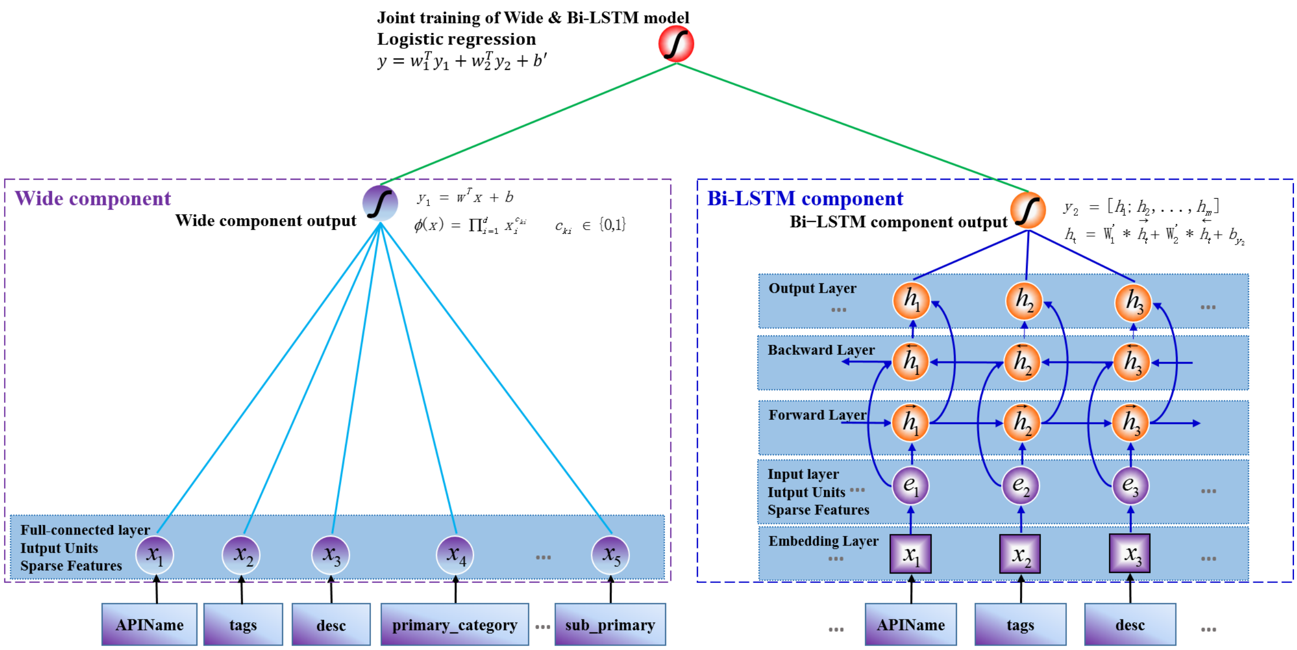

FIGURE 2. The Wide&Bi-LSTM model.

results of Web service categories as the final result of service classification.

A. PREPROCESSING OF WEB SERVICE DESCRIPTION DOCUMENT AND ESTABLISHMENT OF WORD VECTOR

The description document of Web service describes their whole functionality. The description document of Web service is the source of Web service classification. Since some terms in the description document of Web service contain a lot of useless information, the preprocesses are needed to refine the description document of Web service. The preprocesses consist of the following steps:

Step 1: The acquisition and collection of the description document of Web service. The natural language processing toolkit pandas in python is used to extract the five columns of Web APIs (i.e., APIName, tags, desc, primary_category, sub_primary) separately from the selected Web services.

Step 2: The tokenization of the description document of Web service. Words are segmented by spaces and punctuation is separated from words. The NLTK (Natural Language Toolkit) in python is used for the tokenization process.

Step 3: Filter stop words. There are many invalid words and punctuation marks in English, such as, 'a', 'to', ',', and so on. These words or symbols that have no practical meaning are called as stop words. The stop vocabulary table in the NLTK is exploited to remove stop words.

Step 4: Stemming. In English, the same word will have different expressions depending on the tense, person, such as, provide, providing, provides, and so on, but they are actually the same word provide. If these words are treated as different words, the accuracy of similarity computation will be reduced. Therefore, it is necessary to perform stemming.

Step 5: The preprocessed data is used to build the word vector matrix required by the Wide & Bi-LSTM model through the embedding function in Keras, and the feature items in the matrix without data are filled with '0' to make it a matrix with standard shape.

B. THE TRAINING OF THE WIDE & BI-LSTM MODEL

The Wide & Bi-LSTM model is shown in Figure 2, which includes wide component, Bi-LSTM component and joint training of Wide & Bi-LSTM Model. Where, wide component performs the process of memorization, which discovers the correlation between service features from service historical data; Bi-LSTM component achieves the process of generalization, which justly is the transmission of service feature correlation and discovers those rare or never appeared, new service feature composition in service history data. The joint training of the Wide & Bi-LSTM Model is done by considering the weights of the memorization and generalization as well as the weight of their sum to optimize all parameters. The advantages of memorization and generalization are complementary to each other to improve the accuracy of service classification.

1) WIDE COMPONENT

The wide component mainly uses linear regression to perform the breadth prediction. the characteristics of linear regression are generally bi-classification problem and sparse. in statistics, linear regression is a regression analysis that uses the least square function called linear regression equation to model the relationship between one or more independent and dependent variables. therefore, linear regression can accurately measure the degree of correlation and regression fitting between various factors, and also can find the topic words in the web service description document quickly. however, in linear regression, a feature vector of web service description document is not only affected by a single factor. therefore, we use one-hot coding in order to apply a generalized linear model with nonlinear feature transformation to the large-scale regression and classification problem of web service description document with sparse input. it can effectively memorize feature interaction through a series of nonlinear feature transformations and independently select the feature vector of web service description document that need to be memorized. so it can capture direct features interaction in historical web service data, which is simple, extensible, interpretable, and also can add adaptive features to linear regression model. the mathematical expression of wide component is as below:

$$y_1 = w^T x + b \tag{1}$$

Here, $Y_1 = [Y_{11} Y_{12}, \cdots, Y_{1M}]$ is the predictive breadth classification vector for web service, m is the number of web service classifications, and $\sum_{l=1}^{M} Y_{1L} = 1$. $X = [X_1, X_2, \cdots, X_D]$ is the feature vector value with d dimensions of web service description document, i.e. the preprocessed word vector after step III.A .$W = [W_1, W_2, \cdots, W_D]$ IS the models' weight parameter and b is bias.

The main functions of the wide component include features' input and transformation. features transformation is defined as:

$$\varphi_k(x) = \prod_{i=1}^{d} x_i^{c_{ki}} \quad c_{ki} \in \{0, 1\} \tag{2}$$

Here, C_{KI} is a Boolean variable that is 1 if the i-th feature vector of web service description document is a part of the k-th transformation, and 0 otherwise. For binary features, the transformation condition of feature vector of web services description document is defined as: C_{KI} is 1 if and only if the feature vectors that make up web service description document are all 1, at this time X_I is considered as an effective feature for the wide model; and 0 otherwise, at this time X_I is considered as an invalid feature for the wide model and is not considered. that is to say, the linear wide model selects the useful feature vectors of web service description document independently. when the wide model deems that the feature vector of web service description document needs to be considered, the value of C_{KI} is 1, otherwise C_{KI} is 0. in this way, the wide model captures the interactions between the feature vectors of web service description document, and adds nonlinearity and adaptive characteristics to this model, which make web service classification more reasonable and accurate.

2) BI-LSTM COMPONENT

The Bi-LSTM component based on the deep learning neural network is a bi-directional LSTM neural network developed from the LSTM which can learn autonomously from the original feature vector of Web service description document. It can capture and generalize features to find deeper and more abstract Web service features to compensate for the effect of wide component. Therefore, Bi-LSTM component can better generalize for invisible Web service features composition through low-dimensional dense embedding for sparse feature vector of Web service description document. So, when the feature interaction of Web service description document is sparse, the deep neural network with embedded characteristics can summarize the deep feature vector of Web service description document. Furthermore, it can learn the long-term dependence, word order and bidirectional context relationship, and can effectively solve the problem of gradient disappearance and gradient explosion. When Bi-LSTM is used to classify Web services, its two LSTM neural networks with forward and backward input are connected to the same output layer. To improve the accuracy of Web service classification, Bi-LSTM adds the complete pre-order and post-order context information for each point as a part of the input sequence. The basic structure of a typical LSTM unit consists of three gate structures and one cell state, which is shown as Figure 3.

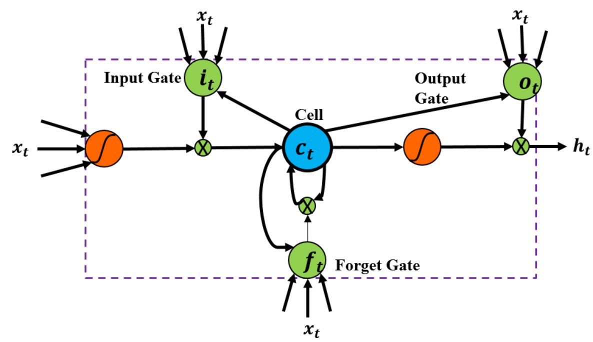

FIGURE 3. The structure of LSTM unit.

As can be seen from Figure 3, the preservation and updating of the cell state in the LSTM unit are determined by Input Gate, Forget Gate and Output Gate. Where the Input Gate controls which parts of the new information are saved to the Cell, Forget Gate determines the preservation information of historical cell state, and Output Gate regulates which parts of the updated cell state are outputted. The specific workflow of the LSTM unit can be shown by the following formula (3)–(7):

$$i_t = \delta(W_i * \left[h_{t-1}, x_t^{'}\right] + b_i) \tag{3}$$

$$f_t = \delta(W_f * \left[h_{t-1}, x_t^{'}\right] + b_f) \tag{4}$$

$$o_t = \delta(W_o * \left[h_{t-1}, x_t^{'}\right] + b_o) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * tanh(W_c * \left[h_{t-1}, x_t^{'}\right] + b_c) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

Among them, i_t, f_t, o_t, and C_t represent the output of Input Gate, Forget Gate, Output Gate, and Cell at time t respectively, and $x_t^{'}$ and h_t represent the input vector and hidden layer vector at time t respectively. δ denotes the sigmoid activation function, W and b represent the weight matrix and bias vector respectively, and their subscripts denote their categories, such as W_i and b_i, which indicate the weight matrix and bias vector that belong to the Input Gate structure. In this way, the Bi-LSTM can store the context information through the LSTM unit.

The structure of the Bi-LSTM is shown in the right part of the Figure 3, which has two parallel LSTM layers in both forward and reverse directions. The two parallel LSTM layers run in the same way as conventional LSTM neural networks. They start at the front and end of the sentence, so they can store sentence information from both directions. In this way, the Bi-LSTM can not only preserve the historical context information of Web service description document (i.e. the pre-order information), but also consider the future context information of Web service description document (i.e. the post-order information). Specifically, after the pre-processing at the step III.A, each feature word x_i in the feature vector of Web service description document $x = [x_1, x_2, \cdots, x_d]$ is converted into corresponding word embedding form, which is a data format that is easy to process by the neural network model. For the t-th time step, two parallel LSTM layers in both forward and reverse directions respectively handle the input x_t from their opposite direction, and then output the sum of the hidden state vectors, which can be described as:

$$h_t = w_1^{'} \overrightarrow{h_t} + w_2^{'} \overleftarrow{h_t} + b_{y_2} \tag{8}$$

Here,$\overrightarrow{h_t}$, $\overleftarrow{h_t}$ are the output results of two parallel LSTM layers in both forward and reverse directions respectively, $w_1^{'}$, $w_2^{'}$ are the weight parameters of two parallel LSTM layers in both forward and reverse directions respectively, and b_{y_2} is the bias. Therefore, starting from formula (8), the predictive depth classification vector for Web service $y_2 = [h_1 h_2, \cdots, h_m]$ can be constructed, m is the number of Web service classifications, and $\sum_{t=1}^{m} h_t = 1$.

3) JOINT TRAINING OF WIDE & BI-LSTM MODEL

In the Wide & Bi-LSTM model, firstly the Wide component and Bi-LSTM component are trained to obtain their prediction results separately. Then all parameters are optimized by joint training by considering the weights of memorization and generalizations as well as the weight of their sum. The memorization and generalization are complementary to each other in order to improve the accuracy of service classification. In this paper, the joint training of the Wide & Bi-LSTM model is done by back-propagating the gradients from the output to both the Wide and Bi-LSTM components of the model simultaneously using mini-batch stochastic optimization. After performing the joint training of the Wide & Bi-LSTM model, the linear regression is used to integrate the breadth and depth prediction results of Web service categories as the final result of service classification, which can be defined as:

$$y = w_1^T y_1 + w_2^T y_2 + b^{'} \tag{9}$$

Here, $\mathbf{y_1}$ is the breadth prediction vector of Web service categories derived by the wide module, $\mathbf{y_2}$ is the depth prediction vector of Web service categories derived by the Bi-LSTM module, and $\mathbf{b}^{'}$ is the deviation. $\mathbf{y} = [\mathbf{y_1 y_2}, \cdots, \mathbf{y_m}]$ is the final prediction vector of Web service classification, m is the number of Web service classification, and $\sum_{\mathbf{l=1}}^{\mathbf{m}} \mathbf{y_l} = \mathbf{1}$. The corresponding cluster of $\mathbf{y_i}$ with the maximum value in the vector $\mathbf{y}$ is the most likely classification for the Web service.

IV. EXPERIMENT

The experimental dataset is crawled from the Programmable Web platform, which contains 6673 Mashups, 9121 Web APIs, and 13613 links between Web APIs and Mashup, as well as the description document of Web service and their tags information. During the experiment, we select 9121 Web APIs from this dataset for experimental comparison and analysis.

A. EXPERIMENT SETTINGS

During the experiment, the top 10, 20, 30, 40, and 50 Web service categories with the highest number of Web services are selected as the classification benchmark datasets, which are divided into 80% training set and 20% test set for each classification benchmark dataset by using the random segmentation tool in Sklearn. In the Bi-LSTM model, some important parameters are set, for example, Embedding_size= 128, Hidden_layer_size= 64, Batch_size= 32, Num_epochs= 1, and dropout and recurrent_dropout are both equal to 0.2.

B. EVALUATION METRICS

To evaluate the performance of the proposed model and method, the 'closed test' principle is adopted in the experiment. The evaluation metrics include precision, recall, F-measure, purity and entropy. We set a standard classification result $SWSC = \{SC_1 SC_2, \ldots, SC_K\}$ for the preprocessed Web service document set, and the experimental Web service classification result as $EWSC = \{C_1 C_2, \ldots, C_{K'}\}$. The precision and recall of the *i-th* Web service category C_i are respectively defined as follows:

$$Precision(C_i) = \frac{|SC_i \cap C_i|}{|C_i|} \tag{10}$$

$$Recall(C_i) = \frac{|SC_i \cap C_i|}{|SC_i|} \tag{11}$$

Here, SC_i is the number of Web services in category SC_i, $|C_i|$ is the number of Web services in category C_i, $|SC_i \cap C_i|$ is the number of Web services jointly appeared in categories SC_i and C_i.

By integrating precision and recall, F-Measure signifies an overall assessment for Web service classification result, which is calculated as below:

$$F - Measure(C_i) = \frac{2 * Precision(C_i) * Recall(C_i)}{Precision(C_i) + Recall(C_i)} \tag{12}$$

Besides, we also employ purity and entropy to evaluate the accuracy of service clustering. The purity of each $\mathbf{C_i}$ and the mean purity of all Web service classification in **EWSC** are respectively defined in the formulas (13) and (14). Similarly, the entropy of each $\mathbf{C_i}$ and the mean entropy of all Web service categories in **EWSC** are respectively defined in the formulas (15) and (16).

$$Purity\,(C_i) = \frac{1}{|C_i|} \max_j n_i^j, \quad 1 \leq i \leq K',\ 1 \leq j \leq K \tag{13}$$

$$Purity\,(\mathbf{EWSC}) = \sum_{i=1}^{K'} \frac{|C_i|}{|EWSC|} Purity\,(C_i) \tag{14}$$

$$Entropy\,(C_i) = -\frac{1}{logK} \sum_{j=1}^{K} \frac{n_i^j}{|C_i|} log(\frac{n_i^j}{|C_i|}) \tag{15}$$

$$Entropy\,(\mathbf{EWSC}) = \sum_{i=1}^{K'} \frac{|C_i|}{|EWSC|} Entropy\,(C_i) \tag{16}$$

Here, $|C_i|$ is the number of Web services in category C_i, n_i^j is the number of Web services belong to SC_j which are successfully divided into C_i, and $|EWSC|$ is the total amount of Web services which need to be classified during the experiment. In short, the bigger recall,precision,purity and the smaller entropy, mean that Web services clustering accuracy is the better.

C. BASELINE METHODS

We select the below state-of-the-art methods as the baseline methods:

- TF-IDF:it exploits the word frequency-inverse document frequency of Web services description document to compute the similarity between Web services and classify them into different categories with similar function.
- LDA: Web services are classified by using LDA topic model. Each Web service belongs to the category is with the corresponding highest probability of topic. That is to say, Web services with the same topic are divided into one category.
- WE-LDA [16]: it leverages the high-quality word vectors to improve the performance of Web services clustering. The word vectors obtained by Word2vec are clustered into word clusters by K-means++ algorithm and these word clusters are incorporated to the semi-supervise LDA training process, which can elicit better distributed representations of Web services.
- LSTM [22]: it is suitable for processing and predicting important events with relatively long intervals and delays in time series. The input and output LSTM neural network are respectively the feature vector matrices of Web services description document and the classification prediction matrix of Web services. The LSTM only exploits the historical context information of Web service description document, i.e. the pre-order information, to classify Web services.
- Bi-LSTM: the input and output Bi-LSTM neural network are same to those of the LSTM neural network. But, the Bi-LSTM has two parallel LSTM layers in both forward and reverse directions. That is to say, it not only extracts the historical context information of Web service description document (i.e. the pre-order information), but also considers the future context information of Web service description document (i.e. the post-order information), to classify Web services.
- Wide & Deep [21]: Wide & Deep learning jointly trains wide linear models and deep neural network to combine the benefits of memorization and generalization for classify Web services.

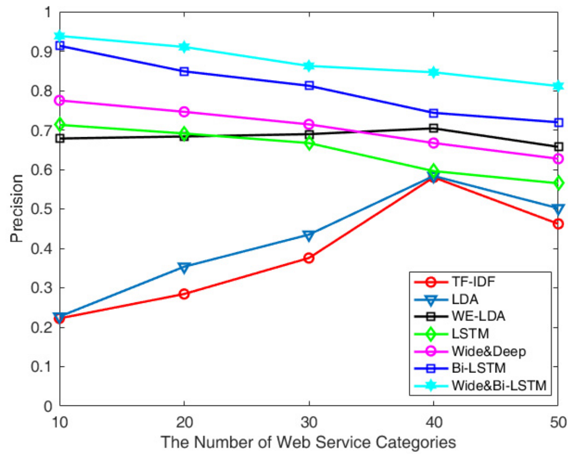

FIGURE 4. Precision Comparison of Different Web Service Classification Approaches.

D. EXPERIMENTAL RESULTS AND ANALYSIS

In this sections, we compare different approaches to evaluate the performance of Web services classification in terms of recall, precision and F-measure. Figures 4–8 reports the comparisons of classification performance for different approaches when the number of Web service categories is ranging from 10 to 50 with a step 10. In the Figures 4–8, the value of the horizontal coordinate is the number of Web service categories, and the value of the vertical coordinate represents the value of the corresponding evaluation metric. The comparisons show that our proposed method significantly improves the classification accuracy in terms of average recall, precision, F-measure, entropy and purity and outperforms all other baseline methods in all cases. Specifically, we have the following observations:

- The classification performance of the Wide&Bi-LSTM model under the same category number is higher than that of the other six models. For example, when the number of service categories is 50, the precision of Wide&Bi-LSTM model respectively has 75.6% improvement over TF-IDF, 61.7% improvement over LDA, 23.4% improvement over WE-LDA. The reason for this is that the Wide & Bi-LSTM model can fully mine the content information contained in the Web services description document by memorization and

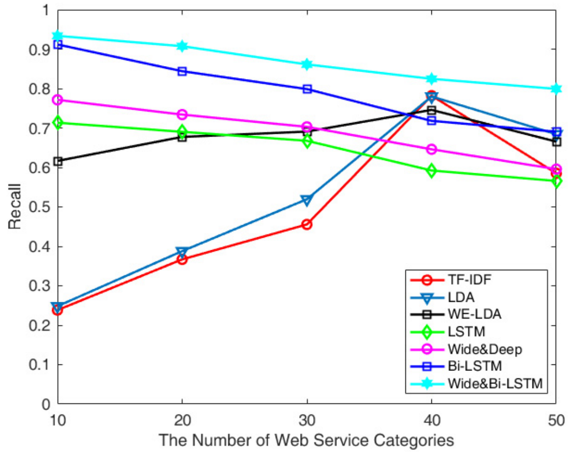

FIGURE 5. **The Recall Comparison of Different Web Service Classification Approaches.**

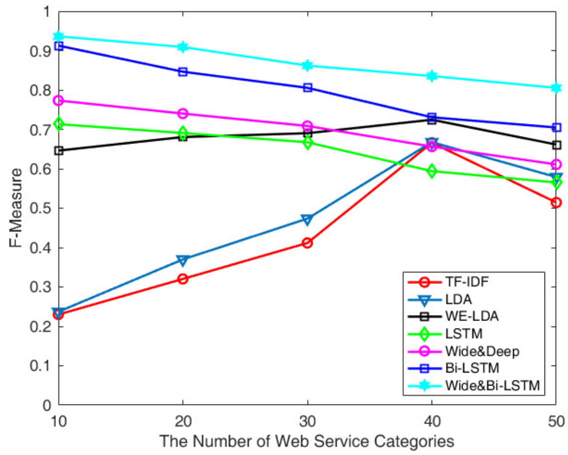

FIGURE 6. **The F-Measure Comparison of Different Web Services Classification Approaches.**

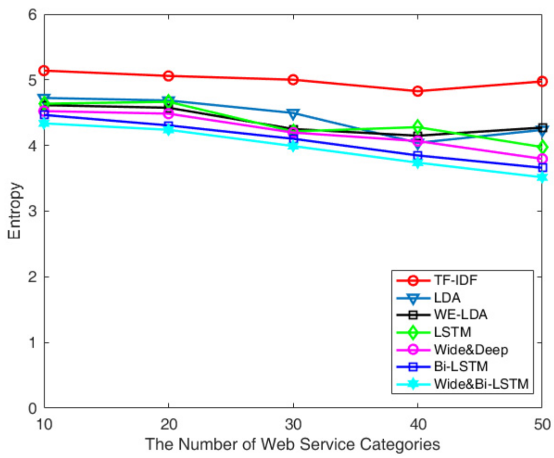

FIGURE 7. **The Entropy Comparison of Different Web Services Classification Approaches.**

generalization so as to achieve a more accurate classification result.

- The performances of TF-IDF, LDA and WE-LDA when the number of Web service categories is equal to 40 are

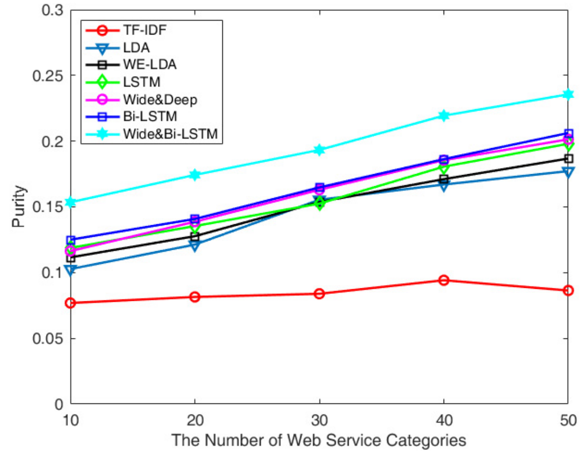

FIGURE 8. **The Purity Comparison of Different Web Services Classification Approaches.**

the best in all cases. When the number of categories is small, an increase in the number of categories from 10 to 40 improves the performance of Web services classification, since more Web services in these categories can be used to learn more hidden information, such as word frequency co-occurrence, semantic correlations, topics, to achieve better classification accuracy. However, when the number of categories continues to increase from 40 to 50, the classification accuracy decreases. The reason for this is that additional categories only have fewer number of Web services with less content information, which weakens the classification accuracy. Furthermore, the performance of TF-IDF is the worst in all cases. This is because TF-IDF only uses the term-based vector space model to represent the features of Web services description document, without considering latent semantic correlation behind them.
- The precision of Wide&Bi-LSTM model has a sharp improvement nearly 43.7% compared to the LSTM model, and the precision of Bi-LSTM model also has a significant improvement compared to the LSTM model. The Bi-LSTM neural network not only extracts the historical context information of Web service description document (i.e. the pre-order information), but also mines the future context information (i.e. the post-order information) compared to the traditional LSTM neural network. Thus, the accuracy of similarity measurement and Web service classification of Wide&Bi-LSTM model is greatly improved.
- Compared with the Wide&Deep model, the accuracy of our Wide&Bi-LSTM model is improved by nearly 29.4%. This is because the deep component in the Wide&Deep model is a feed-forward neural network, which cannot mine the word order and context information of the description document in Web service. While the Bi-LSTM neural network in our Wide&Bi-LSTM model can mine these word order and context

information to significantly improve the accuracy of Web service classification.

- Compared with Bi-LSTM model, the precision of Wide&Bi-LSTM model is slightly improved by 12.8%. This shows that the linear wide model with breadth feature transformation can integrate discrete features in the description documents of Web services, find the topic words of Web services from a large number of Web services description documents accurately and quickly, and exploit joint training process to make up for the deficiency of Bi-LSTM neural network. More importantly, the linear wide model captures the interactions between the feature vectors of Web service description document, and adds nonlinearity and adaptive characteristics to this model, which make Web service classification more reasonable and accurate.
- As far as entropy is concerned, the difference in the number of Web services included in each category (i.e., some categories contain only one or two services, while other categories contain more services), resulting in a high confusion degree in the distribution of Web services. That is to say, the greater the entropy value, the worse the classification effect. It can be seen from Figure 7 that the entropy of Wide&Bi-LSTM model is the smallest when the number of Web service categories is equal to 50, which shows its classification effect is the best compared to other models. Furthermore, we observe that the purity of Wide&Bi-LSTM model has 14.3% improvement over Bi-LSTM model when the number of Web service categories is equal to 50. Totally speaking, the tendency of entropy and purity is roughly consistent with those of the precision, recall and F-Measure.

V. CONCLUSION AND FUTURE WORK

This paper presents a Web service classification method based on Wide&Bi-LSTM model. The breadth prediction of Web service category is performed by exploiting the wide learning model, which captures the interactions between the feature vectors of Web service description document. The depth prediction of Web service category is done by using the Bi-LSTM model to mine he word order and context information of the words in the description documents of Web services. The final Web service classification is achieved by integrating the breadth and depth prediction results of Web service categories. The comparative experiments performed on ProgrammableWeb dataset demonstrate the effectiveness of the proposed method and show that our method significantly improve the accuracy of Web service classification in terms of precision, recall and F-Measure. In the future work, we will investigate and exploit service relationship or link information to construct large-scale service network for facilitating Web service classification.

REFERENCES

[1] C. Peter and R. Stewart, "Service-oriented architecture," *Facultad De Ingenierla*, vol. 118, no. 2, pp. 743–743, 2003.

[2] R. Nayak and B. Lee, "Web service discovery with additional semantics and clustering," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Fremont, CA, USA, Nov. 2007, pp. 555–558.

[3] X. Zhang, Y. Yin, M. Zhang, and B. Zhang, "Web service community discovery based on spectrum clustering," in *Proc. Int. Conf. Comput. Intell. Secur.*, Beijing, China, Dec. 2009, pp. 187–191.

[4] L.-J. Zhang and B. Li, "Requirements driven dynamic services composition for Web services and grid solutions," *J. Grid Comput.*, vol. 2, no. 2, pp. 121–140, 2004.

[5] S. Dustdar and W. Schreiner, "A survey on Web services composition," *Int. J. Web Grid Services*, vol. 1, no. 1, pp. 1–30, 2005.

[6] L. Chen *et al.*, "WTCluster: Utilizing tags for Web services clustering," in *Proc. Int. Conf. Service-Oriented Comput.* Berlin, Germany: Springer-Verlag, 2011, pp. 204–218.

[7] K. Elgazzar, A. E. Hassan, and P. Martin, "Clustering WSDL documents to bootstrap the discovery of Web services," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2010, pp. 147–154.

[8] M. Bruno, G. Canfora, M. Di Penta, and R. Scognamiglio, "An approach to support Web service classification and annotation," in *Proc. IEEE Int. Conf. E-Technol., E-Commerce E-Service*, Hong Kong, Mar./Apr. 2005, pp. 138–143.

[9] 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.

[10] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "A clustering-based QoS prediction approach for Web service recommendation," in *Proc. IEEE 15th Int. Symp. Object/Compon./Service-Oriented Real-Time Distrib. Comput. Workshops*, Shenzhen, China, Apr. 2012, pp. 93–98.

[11] Q. Yu and M. Rege, "On service community learning: A Co-clustering approach," in *Proc. IEEE Int. Conf. Web Services*, Miami, FL, USA, Jul. 2010, pp. 283–290.

[12] M. H. Hasan, J. Jaafar, and M. F. Hassan, "Fuzzy-based clustering of Web services' quality of service: A review," *A Rev. J. Commun.*, vol. 9, no. 1, pp. 81–90, 2014.

[13] C. B. Pop *et al.*, "Semantic Web service clustering for efficient discovery using an ant-based method," in *Intelligent Distributed Computing IV*, Tangier, Morocco: DBLP, 2010, pp. 23–33.

[14] J. Wu, L. Chen, Z. Zheng, M. R. Lyu, and Z. Wu, "Clustering Web services to facilitate service discovery," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 207–229, 2014.

[15] D. N. Sotiropoulos, C. D. Kounavis, and G. M. Giaglis, "Semantically meaningful group detection within sub-communities of Twitter blogosphere: A topic oriented multi-objective clustering approach," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining.*, Aug. 2013, pp. 734–738.

[16] M. Shi, J. Liu, D. Zhou, M. Tang, and B. Cao, "WE-LDA: A word embeddings augmented LDA model for Web services clustering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Honolulu, HI, USA, Jun. 2017, pp. 6–9.

[17] L. Chen, Y. Wang, Q. Yu, Z. Zheng, and J. Wu, "WT-LDA: User tagging augmented LDA for Web service clustering," in *Proc. Int. Conf. Service Oriented Comput. (ICSOC)*. Berlin, Germany: Springer, 2013, pp. 162–176.

[18] Q. Yu, H. Wang, and L. Chen, "Learning sparse functional factors for large-scale service clustering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, New York, NY, USA, Jun./Jul. 2015, pp. 201–208.

[19] B. Cao *et al.*, "Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model," in *Proc. Int. Conf. Web Services (ICWS)*, San Francisco, CA, USA, Jun./Jul. 2016, pp. 212–219.

[20] B. Cao, X. Liu, M. M. Rahman, B. Li, J. Liu, and M. Tang, "Integrated content and network-based service clustering and Web APIs recommendation for Mashup development," *IEEE Trans. Services Comput.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/document/7885122. doi: 10.1109/TSC.2017.2686390.

[21] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. ACM 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, 2016, pp. 7–10.

[22] H. Wang, Z. Yang, and Q. Yu, "Online reliability prediction via long short term memory for service-oriented systems," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Honolulu, HI, USA, Jun. 2017, pp. 81–88.

[23] S. Li, Z. Yan, X. Wu, A. Li, and B. Zhou, "A method of emotional analysis of movie based on convolution neural network and bi-directional LSTM RNN," in *Proc. IEEE 2nd Int. Conf. Data Sci. Cyberspace*, Shenzhen, China, Jun. 2017, pp. 156–161.

[24] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," *Mach. Learing*, vol. 55, no. 3, pp. 311–331, 2002.

[25] M. Shi, J. Liu, D. Zhou, and Y. Tang, "A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data," *IEEE Trans. Services Comput.*, to be published.

[26] M. Shi, Y. Tang, and J. Liu, "Functional and contextual attention-based LSTM for service recommendation in Mashup creation," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[27] M. Shi, J. Liu, B. Cao, Y. Wen, and X. Zhang, "A prior knowledge based approach to improving accuracy of Web services clustering," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, San Francisco, CA, USA, Jul. 2018, pp. 1–8.

[28] B. Xia, Y. Fan, W. Tan, K. Huang, J. Zhang, and C. Wu, "Category-aware API clustering and distributed recommendation for automatic mashup creation," *IEEE Trans. Services Comput.*, vol. 8, no. 5, pp. 674–687, Sep. 2015.

[29] C. V. S. Prazeres, M. da Graça Campos Pimentel, E. V. Munson, and C. A. Teixeira, "Toward semantic Web services as MVC applications: From owl-s via UML," *J. Web Eng.*, vol. 9, no. 3, pp. 243–265, 2010.

[30] B. Cao, X. Liu, J. Liu, and M. Tang, "Domain-aware Mashup service clustering based on LDA topic model from multiple data sources," *Inf. Softw. Technol.*, vol. 90, pp. 40–54, Oct. 2017.

[31] B. Cao, J. Liu, Y. Wen, H. Li, Q. Xiao, and J. Chen, "QoS-aware service recommendation based on relational topic model and factorization machines for IoT Mashup applications," *J. Parallel Distrib. Comput.*, Apr. 2018. doi: 10.1016/j.jpdc.2018.04.002.

HONGFAN YE received the bachelor's degree from the School of Computer Science and Engineering, Hunan University of Science and Technology. His research interests include service computing and software engineering.

BUQING CAO received the Ph.D. degree in computer science from Wuhan University, China, in 2011. From 2015 to 2016, he was a post-Ph.D. Researcher with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR, USA. He is currently an Associate Professor, and also the Vice-Dean of the School of Computer Science and Engineering, Hunan University of Science and Technology, China. His research interests include service-oriented software engineering, service computing, and social networks. He has published over 70 papers in the international conferences and journals, such as the IEEE TSC, IST, JPDC, JWSR, ICWS, SCC, IEEE BigData, APSCC, GCC, and HPCC. He is on the editorial boards of several journals and has served as a Program Committee Member for multiple international conferences.

ZHENLIAN PENG received the Ph.D. degree in computer science from Wuhan University, China, in 2016. He is currently a Lecturer with the School of Computer Science and Engineering, Hunan University of Science and Technology. He has published over 10 papers in the international conferences and journals. His research interests include service computing, requirement engineering, and software engineering.

TING CHEN received the bachelor's degree from the School of Computer Science and Engineering, Hunan University of Science and Technology. His research interests include service computing and cloud computing.

YIPING WEN received the Ph.D. degree in computer science from Central South University, in 2013. He is currently a post-Ph.D. Researcher with the Department of Computer Science and Technology, Nanjing University, China. He is also an Associate Professor with the School of Computer Science and Engineering, Hunan University of Science and Technology, China. He has published over 30 papers in the international conferences and journals. His current research interests include business process management, big data, and cloud computing.

JIANXUN LIU received the Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2003. He is currently a Professor and the Dean of the School of Computer Science and Engineering, and also the Associate Director of the Key Lab of Knowledge Processing and Networked Manufacturing, Hunan University of Science and Technology. His research interests include workflow management, service computing, and cloud computing. He has published over 100 papers in well-known conferences and journals, such as the IEEE TPDS, the IEEE TSC, the IEEE TNSM, IR, IM, FGCS, JCSS, ICWS, and SCC. He has served as a Program Committee Member or the Program Chair of the major conferences in his area (ICWS, SCC, SCA, CGC, APBPM, SKG, and so on) and a Reviewer for TKDE, TSC, and TPDS.