

A Novel Mixed Wide and PSO-Bi-LSTM-CNN Model for the Effective Web Services Classification

K. Punitha

Assistant Professor (Sr), School of Computing Science and Engineering (SCOPE), VIT University, Chennai Campus. E-mail: dr.punitha.k@gmail.com

Received June 08, 2020; Accepted August 10, 2020

ISSN: 1735-188X

DOI: 10.14704/WEB/V17I2/WEB17026

Abstract

In software technology, over the diversified environment, services can be rendered using an innovative mechanism of a novel paradigm called web services. In a business environment, rapid changes and requirements from various customers can be adapted using this service. For service management and discovery, the classification of Web services having the same functions is an efficient technique. However, there will be short lengthened Web services functional description documents, having less information, and sparse features. This makes difficulties in modelling short text in various topic models and leads to make an effect in the classification of Web services. A Mixed Wide and PSO-Bi-LSTM-CNN model (MW-PSO-Bi-LSTM-CNN) is proposed in this work for solving this issue. In this technique, the Web service category's breadth prediction is performed by combining Web services description document's discrete features, which exploits the wide learning model. In the next stage, the PSO-Bi-LSTM-CNN model is used for mining Web services description document word's context information and word order, for performing the Web service category's depth prediction. Here, particle swarm optimization (PSO) is integrated with the Bi-LSTM-CNN network for computing various hyper-parameters in an automatic manner. In third stage, Web service categories, results of depth, and breadth prediction are integrated using a linear regression model as final service classification result. At last, MW-PSO-Bi-LSTM-CNN, Wide&Bi-LSTM, and Wide&Deep web service classification techniques are compared and a better result with respect to web service classification accuracy is obtained using the proposed technique as shown in experimental results.

Keywords

Web Service Classification, Linear Regression, Particle Swarm Optimization, Bi-LSTM Model, Convolutional Neural Network, Wide Learning Model.

Introduction

The ways in which organization performs their business with their customers and partners are revolutionized due to the Internet technologies growth. For highly automated and effective business process and to meeting requirements, Web operations are focused by companies. For competition, right software should be implemented by companies and recent technology trends must be followed. Over time, in order to overcome rapid changes in business requirements, they need find an integrated, e-business solution. A new paradigm termed as web services, produced due to the development in usage of web in conducting business [1].

Web services have software components with independent, distributed and loosely coupled services operating at web infrastructure. They are language independent and used as a platform for accessing from heterogeneous environments. Web services interfacing and functional aspects are focused on various researchers due to rapid introduction of web-services technologies, including XML and HTTP based messaging.

Open standards like XML and HTTP based protocols including UDDI, WSDL and SOAP are used in this for making communication. On the web, service location with provided services functionality are described using a document called WSDL and. In a UDDI registry, information related to web service are given as an input for permitting web service consumers for finding out and locating required services. In UDDI registry, using available information, SOAP messages are constructed according to web services and client developer uses instructions in WSDL to exchange data with service over HTTP attributes [2].

For engineers and application designers to utilize the administrations they have to scan for them inside the administration storehouses. This assignment is regularly alluded to as service discovery. Be that as it may, service discovery is as yet a challenging and blunder inclined errand, as most archives offer watchword coordinating based hunt instruments. Related to this issue is the way that administration stores are sorted out principally by static structures that don't permit adaptable and dynamic association of administrations [3].

At present, there have been a ton of looks into on Web administration characterization, among which they principally center around useful trait based Web service recommendation [3] and Web service classification [4]. As of late, deep learning has earned impressive enthusiasm for some examination fields, for example, computer vision

and normal language handling, owing not exclusively to heavenly execution yet in addition the alluring property of taking in highlight portrayals without any preparation.

This examination work has as its principle target to improve the hierarchical structure of Web administration stores in a manner that encourages the disclosure of administrations. The principle commitment of this article centers on a Web administrations order calculation utilizing profound learning. Accordingly, client get assortments of Web administrations sorted out by topics, your hunt is smoothed out, expending less assets, since it is done between administrations inside a similar class [4], [5]. The fundamental commitment of the work is as per the following:

- 1) A Mixed Wide and PSO-CNN- Bi-LSTM model is proposed in this work for performing memorization and generalization process using joint training for fully mining historical correlation features and deeper level service feature interaction in Web service description texts for producing highly accurate similarity measurement of services.
- 2) Web services description text's service features are predicted using this model. In this technique, the Web service category's breadth prediction is performed by combining the Web services description document's discrete features, which exploits the wide learning model.
- 3) In the next stage, PSO-Bi-LSTM-CNN model is used for mining Web services description document word's context information and word order, for performing the Web service category's depth prediction.
- 4) In the third stage, Web service categories, results of depth and breadth prediction are integrated using linear regression model as the final service classification result. From Programmable Web, real data are obtained for conducting experimentation and performance comparison to verify the effectiveness of the proposed technique.
- 5) At last, MW-PSO-Bi-LSTM-CNN, Wide&Bi-LSTM and Wide&Deep web service classification techniques are compared and a better result with respect to web service classification accuracy is obtained using the proposed technique as shown in experimental results.

The remainder of this paper is organized as follows: Section II presents an overview of 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 with future work.

Related Work

This section provides as well as devises a web service classification models based on deep learning taxonomy, in addition, it provides a comprehensive summary of state-of-the-art techniques. Service description technique's classification and investigations are provided in [5]. In different automated service composition studies, these techniques are used.

Five two-value dimensions are proposed for proper classification of service description techniques and for positioning then in an automated service composition throughout the service description world. Using these dimensions, a categorization is performed first and easy introductions to current representative industrial service description standards are provided.

A tuple-based service description paradigm is used in the majority of studied automated composition studies as discovered. With automated composition techniques, this paradigm is adapted and an exhaustive classification and discussion of this paradigm are given. A deep neural network is presented in [6] for automatically abstracting low-level service description representation to high-level features without feature engineering and predicting service classification on 50 service categories.

On 10,000 real-world web services, a comprehensive experimental study is conducted via a comparison of 10 machine learning techniques in order to demonstrate the effectiveness of the proposed techniques. The feature engineering quality defines conventional deep learning techniques performances.

A pool-based active learning idea is to leverage in [7] for realizing scalable service classification techniques. Manual labelling of a huge amount of services is not done for constructing a complete training set. This technique has a base classifier having a small training set and most informative services labels outside initial training set computed iteratively.

With a smaller training set, comparable accuracy is achieved by this proposed technique when compared with traditional techniques. In-text classification, because of the effectiveness of SVM, it is used as a base classifier. For addressing this issue produced due to the generation of sparse term vectors from service descriptions, probabilistic topic models are also incorporated for reducing dimensions for enhancing efficiency.

Solid mathematical foundation characteristics are utilized in [8] for producing naive Bayes classification techniques stable classification efficiency. Naïve Bayes' theory-based

semantic Web service classification technique is proposed in this. Usage of three Bayesian classification stages are elaborated for semantic Web services classification with execution capacity and service interface consideration.

Instructions to arrange and sort out the semantic Web administrations to assist clients with finding the administrations to address their issues rapidly and precisely is a key issue to be comprehended in the time of administration situated programming building. Different machine learning calculation like KNN is applied for web administration disclosure. The frameworks are adequately learning the information and assess the presentation precision with the given datasets.

For efficient web service classification, an improved fuzzy with the KNN algorithm is proposed in [9]. In the form of performance measures and accuracy, outcomes are enhanced using this technique. Implementation of work based on service-oriented engineering is explained in [10] and for reuse, it utilizes a service-oriented framework. In addition, it clarified the usage of web examination arranged foundation for reuse. A compelling component choice system is used in [11] to take after web administrations revelation using Centroid classifier calculation.

For report arrangement, the centroid classifier is not viably used due to computational multifaceted bigger and nature memory prerequisites, other than being straightforward and vigorous. A special type of auto-encoder called Variation Auto encoders used for learning service descriptions features in [12]. Encoded representation is restricted to model latent variables in this technique. Auto-encoder is a deep neural network, utilized in efficient coding's unsupervised learning. A real 17 113-service dataset, which is extracted from ProgrammableWeb.com API social repository, is used for training this auto encoder.

A novel self-adaptive semantic classification technique is proposed in [13]. Web user's logs frequent pattern and service knowledge ontology are used in this method for enhancing web service discovery. For service discovery, these techniques show limitation in classification to domain trained knowledge. A theoretical graph-based service composition analysis is presented in [14] with respect to dependency with service discovery.

Using this analysis, a composition framework is defined using the fine-grained I/O service discovery integration for enabling graph-based composition generation, which has services set that are semantically relevant to an input-output request. An optimal

composition search algorithm is included in this proposed framework for extracting the best composition for reducing the number of services and lengths from graph. System scalability is enhanced using various graph optimisations.

It also provides practical implementation used in the empirical analysis. Wang et al. [15] implemented an online QoS prediction technique for mining text's context information and word order. In this model, the service system's future reliability is learned as well as predicted using the LSTM model. Although, text's context information and word order are mined to a certain extent using this LSTM model. Historical context information like positive word order information is only considered in this and future context information like reverse word order information is not considered.

For solving this problem, the Bi-LSTM model exploited for simultaneously modelling the reverse and positive word order in a method proposed in [16]. Discrete features are combined using a wide learning model, where, in Web services description documents, context information is included using this for facilitating classification of Web services. However, effective feature extraction can be done using this method and ignored the context and accurate text semantics are not produced. This work uses the Bi-LSTM-CNN technique for solving web service classification problems.

Proposed Methodology

Figure 1 shows the proposed method's overall framework. There are three major stages, namely acquisition, Web service description document's pre-processing and MW-PSO-Bi-LSTM-CNN model training, and Web service classification. An MW-PSO-Bi-LSTM-CNN model based Web service classification technique is proposed in this work. The wide learning model is exploited for performing the Web service category's breadth prediction. In Web Services description document's acquisition and pre-processing, from Programmable Web, crawled the web service's description text and other related information, where, feature vector matrix is built by extracting respective feature columns.

The MW-PSO-Bi-LSTM-CNN components are used for training the Web service description document's computed feature vector matrices in MW-PSO-Bi-LSTM-CNN model training process for deriving its generalization and memorization classification vectors. According to MW-PSO-Bi-LSTM-CNN model performed the prediction and depth and breadth. In Web service classification process, implemented the joint training

and classification prediction and Web service categories breadth and depth prediction are integrated using a linear regression algorithm as final service classification results.

Pre-process of Web Services

A lot of noise and uninformative parts like punctuations, advertisements, scripts, and HTML tags are there in online text. So, a process needs to be applied for cleaning the text, for instance, noises is removed for producing better classification results.

The acquisition and collection of the description document of Web service: The Web APIs, five columns are extracted separated from selected Web services using natural language processing toolkit pandas in python. They are sub_primary, primary_category, desc, tags, and APIName.

Lower the text: Every instance of the dataset is converted into lowercase in the first step, which allows making a better comparison of words with all models that are created.

Removing Stop Words: In text categorization, a standard technique used is stopped word removal. Commonly used words like prepositions and articles are manipulated using this technique. In a classification task, their contributions are very less, so they are removed from the text. Natural Language Toolkit (NLTK) is used for removing irrelevant words. From the sentences, stop words like “is, the, a” etc. are removed as no information is carried by these words.

Tokenize the text: The given text is split into small pieces termed as tokens using a process called Tokenization. Punctuation marks, numbers, words, and others are considered as tokens. Words are tokenized for getting text tokens, for example, sentences are broken into a list of words. A text corpus is vectorised using this class by making every text into a vector or integer sequence, where, coefficient of every token may be binary according to word count and tf-idf.

Remove useless words contain numbers: From the text, numbers like “1,2,3,4,5...” are removed. Numbers are removed in a condition where much of information is not given by text clustering or key-phrases in getting the main words.

Remove punctuation: From the text, punctuations like “?!” And symbols like “@#” are removed.

Lemmatize the text: Stemming and Lemmatization are similar but context to words are given by Lemmatization. So, words having similar meanings are linked with one word.

Words morphological analysis is done using this. In specific, root forma re-obtained by lemmatizing the text like functionality and functions.

Removing xml: Xml tags are removed, but for weighting, enclosed terms are extracted and any owl related language keywords like owl, xmlns, xml, edf, rdfs are removed.

Removing Stemming words: At last, terms with the same root as teaching, teacher and teach are grouped for stemming words. It is performed using the most popular algorithm known as porter stemmer.

Word Embedding: Word vector-matrix needed for MW-PSO-Bi-LSTM-CNN model is build using pre-processed data through embedding function in Keras. In the matrix, feature items without data are filled with '0' for making it as a matrix in a standard shape.

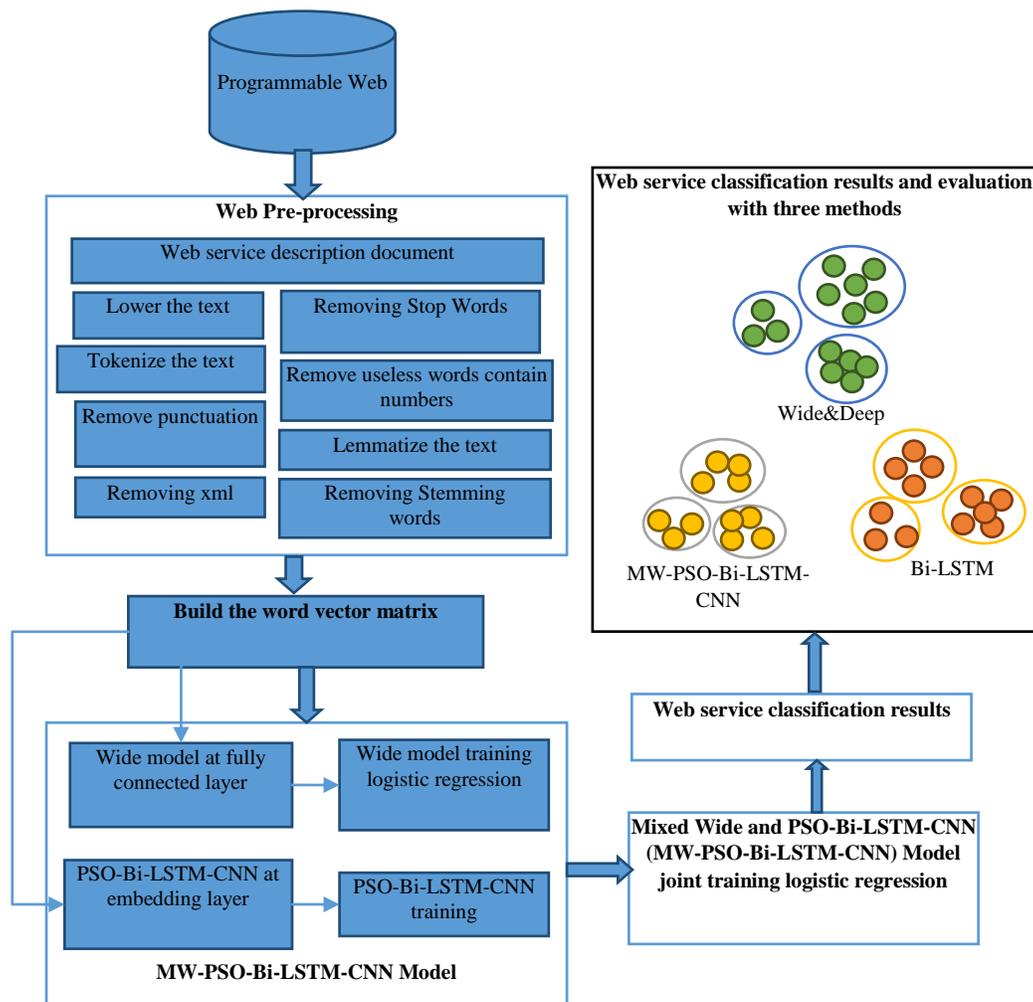


Fig.1 Overall framework of the web service classification using MW-PSO-Bi-LSTM-CNN Model

1) Proposed MW-PSO-Bi-LSTM-CNN Model

The MW-PSO-Bi-LSTM-CNN's joint training, PSO-Bi-LSTM-CNN component and wide component are included in MW-PSO-Bi-LSTM-CNN. Where the memorization process is performed using wide component for discovering a correlation between service features of service historical data; generalization process is achieved using PSO-Bi-LSTM-CNN component, which is used for service feature correlation transmission and discovering rare or never appeared feature composition in-service data history.

Here, Bi-LSTM-CNN's hyper parameter is selected using PSO. Generalization and memorization weights along with their weight sum are considered for optimizing all the parameters using mixed training of MW-PSO-Bi-LSTM-CNN Model and PSO. For enhancing the service classification, complementary nature between generalization and memorization is used.

Wide Methodology: Breadth prediction is performed using linear regression in Wide component. In general, sparse and bi-classification problem are the linear regression characteristics. In historical Web service data, direct features interactions are captured using this, which is interpretable, extensible, simple and linear regression model can be added with adaptive features and wide component's mathematical expression is given by,

$$PB_1 = W^T X + b \quad (1)$$

Where, Web service's predictive breadth (PB) classification vector is given by $Pb_1 = [pb_{11} + pb_{12}, \dots, pb_{1m}]$, Web service classifications count is given by m , and Web service description document's feature vector value having d dimensions is given by $\sum_{l=1}^m PB_{1l} = 1$, $X = [x_1, x_2, \dots, x_d]$, i.e. pre-processed word vector. Transformation and input of features include the wide component's main functions. Features transformation (Ft) is expressed as,

$$Ft(x) = \prod_{i=1}^d X_i^{\mathcal{B}_{ki}}, \mathcal{B}_{ki} \{0,1\} \quad (2)$$

Where the Boolean variable is represented as \mathcal{B}_{ki} , if Web service description document's i -th feature vector is a part of k -th transformation, then it will be 1, else it will be 0. The Web services description document feature vector's transformation condition for binary features are defined as: is 1 if and only if the feature vectors that makeup Web service description document are all 1, at this stage, for the wide model, effective feature corresponds to X_i and 0 otherwise, at this stage, for the wide model, X_i is corresponds to an invalid feature and it is not considered.

In indicates that, Web service description document's useful feature vectors are selected using the linear wide model in an independent manner. The \mathcal{B}_{ki} value is made 1, if wide model considers the Web service description document's feature vector, otherwise \mathcal{B}_{ki} is 0. Between Web service description document's feature vectors, interactions are captured using wide model in this manner and this model is added with adaptive characteristics and nonlinearity for making accurate as well as reasonable service classification.

PSO-Bi-LSTM-CNN: The deep learning neural network-based bidirectional LSTM (Bi-LSTM) component is a bi-directional LSTM neural network developed from LSTM. From Web service description document's original feature vector, features are learned automatically. In these implementations, two directions representations are obtained using the Bi-LSTM model, and using a convolutional neural network, a new expression is formed by combining two directions representations.

To the right and left text vector, every word expression is added by itself for indication. A loop structure is utilized for right and left texts, which is a previous word's non-linear transformation and text on the left side. Contextual information is preserved using this technique and a wide range of word order.

More abstract and deeper Web service features are computed for compensating a wide component effect by capturing and generalizing features. For invisible Web service features composition, better generalization is achieved by Bi-LSTM-CNN component through low-dimensional dense embedding for Web service description document's sparse feature vector. Web service description document's deep feature vector is summarized using deep a neural network with embedded characteristics if the Web service description document's feature interaction is sparse.

Long-term dependence, bidirectional, and word order context relationships are learned using it. Gradient explosion and gradient disappearance problem are solved effectively using this. If web service classification is done using Bi-LSTM-CNN, its two Bi-LSTM-CNN having forward and backward input are connected with the same output layer.

For enhancing Web service classification accuracy, for every point, entire pre-order and post-order context information are added in Bi-LSTM-CNN as an input sequence. In this study, CNN kernels count, units count in Bi-LSTM-CNN model's fully connected layer, hidden units count in Bi-LSTM are optimized using PSO.

Particle Swarm Optimization: Particle's movement is represented algorithmically using PSO. Various particles are used for searching space and if a specific particle found an optimal position, results are updated. Here, hyper parameters are assumed as a fully connected layer's output size, Bi-LSTM, and kernel size. Particle swarm's fitness corresponds to LSTM's initial output error then, based on the condition, the particle's performance is judged.

In this process, following lists the major combination steps of two algorithms: the relevant LSTM network parameter is computed using PSO parameters, the latter's output error is assumed as the former's fitness value, neural network's training parameters corresponds to an optimum solution derived from PSO. Hyper parameters space is searched by PSO using particles that swarm like birds flock. In general, every particle moves in the hyper parameters optimizing direction.

Every particle's movement is tracked using a heuristic PSO algorithm and in hyper parameter space, the optimum position of every particle is computed using this. In hyper parameter space, at arbitrary positions, particles are placed and assigned with a small initial velocity.

The other particle's direction and speed affect every particle's speed. Over the generations, particle motion is updated using the PSO algorithm. In order to find the optimum position, inertia, velocity, and position of every particle are changed. For optimizing Bi-LSTM-CNN model, every particle is brought together over time and generations. Every particle's best past location is tracked using the PSO algorithm. So, it makes neighboring particles to move to a good location. Every particle is updated using,

$$v_i(t + 1) = (c_1 \times rand \times p_i^{best} - p_i(t)) + (c_2 \times rand \times p_g^{best} - p_i(t)) + v_i(t) \quad (3)$$

Where, for *ith* particle, updated velocity is given by, $v_i(t + 1)$. With a uniformity random variable, best individual position is given by p_i^{best} and it is a best position of all particles. For personal best position's accelerations coefficient is given by c_1 and for global best position's accelerations coefficient is given by c_2 . Below expression is used for updating particle's position as,

$$p_i(t + 1) = p_i(t) + v_i(t) \quad (4)$$

At time t , position of *ith* particle is represented as $p_i(t)$.

Step 1:At first, web service features are loaded to input training Web service description documents and in every Bi-LSTM-CNN network layer, number of neurons are computed;

Step 2:PSO's relevant parameters are computed and respective LSTM network parameters are also computed;

Step 3:Particle's initial position and velocity are generated randomly.

Step 4:Neural network is trained and output error is computed, which is a particle swarm's fitness value.

$$Error = \frac{total\ incorrect\ decisions}{total\ number\ of\ services}$$

Step 5:Global and individual extremes are computed; every particle's position and velocity are updated.

Step 6:In order to stop the iteration, PSO conditions are verified; if it is satisfied, results are saved and move to step (7), else proceed to step (4).

Step 7:Bi-LSTM-CNN's weight is assumed as a PSO's global optimum value.

Step 8:Neural network are trained using Bi-LSTM-CNN algorithm and in order to stop the iteration, Bi-LSTM-CNN conditions are verified. If they are satisfied, results are saved and moved to step 9, else steps from (4) to (8) are repeated.

Step 9:Test samples are used for simulation and output is obtained.

Size of particle is set as 60, 100 is set as maximum count of iterations, 0.5 as inertia weight, 1 c and 2 c to 1.5 as acceleration factor, [-5,5] as position restriction interval and [-1, 1] as speed restriction interval according to experimental and experience comparison. Particle swarm optimization's fitness function is assumed as LSTM neural network model's output error. Increase in iteration time makes the reduction in function value and increase in individual's fitness.

Bi-LSTM-CNN: It has two parallel LSTM layers in reverse as well as in forward directions. Their behaviour is like a conventional LSTM neural network layers. At sentence's front and end, they start for storing sentence information in both the directions. In this way, Web service description document's historical context information, which is pre-order information, can be preserved while considering future context information of it, which is the post-order information.

In specific, Web service description document X 's every feature word in feature vector is converted as a receptive word embedding form after pre-processing. This produces a data format, which can be processed easily using neural network model. For t -th time step,

input from negative and positive direction are handled by positive and negative direction layers in two parallel LSTM and produce output as sum of hidden state (H_s) vectors, which is described as,

$$Hs_t = w_1 \overrightarrow{hs}_{tl} X + w_1 \overleftarrow{hs}_{tr} X + b_{DP_2} \quad (5)$$

Here, two parallel LSTM layers direction in forward direction, so called left direction, output result is represented as \overrightarrow{hs}_{tl} and its weight parameter is represented as w_1 , in backward direction, so called right direction, output result is represented as \overleftarrow{hs}_{tr} and its weight parameter is represented as w_2 , predictive depth vector bias is represented as b .

Web service description's sequential reading is referred as forward LSTM and reverse order reading of corpus is referred as backward Bi-LSTM. In this structure, both reverse order semantics, as well as forward order semantics are considered for enhancing Web service semantics expression. Every position of a word vector has two direction expressions, which are obtained using LSTM. At hidden state, expressed every word's left and right perspective, which are shown in below expression,

$$\begin{aligned} \overrightarrow{hs}_{tl}(x_i) &= g(w_l hs_{tl}(x_{i-1}) + w_{sl} P(x_{i-1})) \quad (6) \\ \overleftarrow{hs}_{tr}(x_i) &= g(w_r hs_{tr}(x_{i+1}) + w_{sr} P(x_{i+1})) \end{aligned}$$

At hidden state, word x_i 's left context vector is represented as $\overrightarrow{hs}_{tl}(x_i)$, hidden layers are converted into next hidden layer using a matrix w_l , next word's left context and current word's semantics are combined using a matrix called w_{sl} , previous word's left context is represented as $\overleftarrow{hs}_{tr}(x_i)$, previous vector's word vector is represented as $P(x_{i-1})$. Likewise, concluded the word x_i 's right context vector $\overleftarrow{hs}_{tr}(x_i)$.

Current location's left and right perspectives are combined using convolution through this Bi-LSTM layer for forming fusion context word's new expression. There exist a right context vector $\overleftarrow{hs}_{tr}(x_i)$, word vector $P(x_i)$ of a current word and left context vector's concatenation $\overrightarrow{hs}_{tl}(x_i)$ as shown in expression (6).

$$x_i = (\overrightarrow{hs}_{tl}(x_i), P(x_i), \overleftarrow{hs}_{tr}(x_i)) \quad (7)$$

Highly accurate representation of text semantics can be done using this expression and eliminated the word x_i 's ambiguity.

The tanh activation function is applied with linear transformation, after obtaining word x_i 's new expression as expressed in (5) and results are send to max-pooling layer. A fixed length vectors are formed by converting various text lengths using pool layer and throughout the Web service categories depth prediction (DP) vector's text, information can be captured.

$$DP_2 = \tan h(w_i x_i + b_i) Hs_t \quad (8)$$

At last, after processing softmax function, probability is formed by converting output number and it is converted as a respective probability p_i . It is expressed as,

$$p_i = \frac{\exp(DP_i)}{\sum_{i=1}^m \exp(DP_i)} \quad (9)$$

So, using the Hs_t expression, constructed the predictive depth classification (DP) vector of Web service $DP_i = [hs_1 hs_2, \dots, hs_m]$ and Web service classifications count is represented as m and $\sum_{t=1}^m hs_t = 1$.

Mixed Wide and PSO-Bi-LSTM-CNN (MW-PSO-Bi-LSTM-CNN) Model

Initially, the trained Wide component and PSO-Bi-LSTM-CNN component in the Wide & Bi-LSTM-CNN model for obtaining their results of prediction in a separate manner. Generalization and memorization weights along with their weight sum are considered for optimizing all the parameters using mixed training and PSO. For enhancing the service classification, complementary nature between generalization and memorization is used.

In this work, gradients from the output of Wide and PSO-Bi-LSTM-CNN model components output are back-propagated simultaneously via mini-batch stochastic optimization in order to perform mixed training of Wide and PSO-Bi-LSTM-CNN model. Web service categories breadth and depth prediction results are integrated using linear regression after performing MW-PSO-Bi-LSTM-CNN model as a final service classification result, which is expressed as,

$$Y = w_1^T PB_1 + w_2^T DP_2 + b'$$

Where, Web service categories breadth prediction vector is represented as PB_1 , which is derived from wide module, Web service categories width prediction vector is represented as DP_2 , which are derived from PSO-Bi-LSTM-CNN module and deviation is represented as b' . Web service classification's final prediction is represented as $y = [y_1, y_2, \dots, y_m]$ and number of Web service classification is represented as m and $\sum_{i=1}^m y_i = 1$. The respective y_i 's cluster with maximum value in vector is a most likely classification for Web service.

Experimental Results and Discussion

The Programmable Web platform provides the dataset for experimentation, which has 9121 Web APIs, 6673 Mashups and 13613 links between Mashup and Web APIs with Web service document description and their tagging information. For experimentation and analysis, 9121 Web APIs are selected from this dataset in experimentation.

In this experimentation, classification benchmark datasets are selected from top the 10, 20, 30, 40, and 50 Web service categories having the largest Web services number. In Sklearn, random generation is used for dividing 80% training set and 20% test set for each classification benchmark dataset.

For evaluating the proposed MW-PSO-Bi-LSTM-CNN model's performance in experimentation adapted closed test' principle and existing techniques like Wide&Bi-LSTM and Wide&Deep for performance comparison [16, 17]. Metrics like accuracy, F-measure, recall, and precision are used for evaluation.

$$Precision = \frac{\text{correctly classified services}}{\text{total classified services}}$$

$$Recall = \frac{\text{correctly classified services}}{\text{total correct services belong to that class}}$$

$$Accuracy = \frac{\text{total correct decisions}}{\text{total number of services}}$$

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

1) Precision

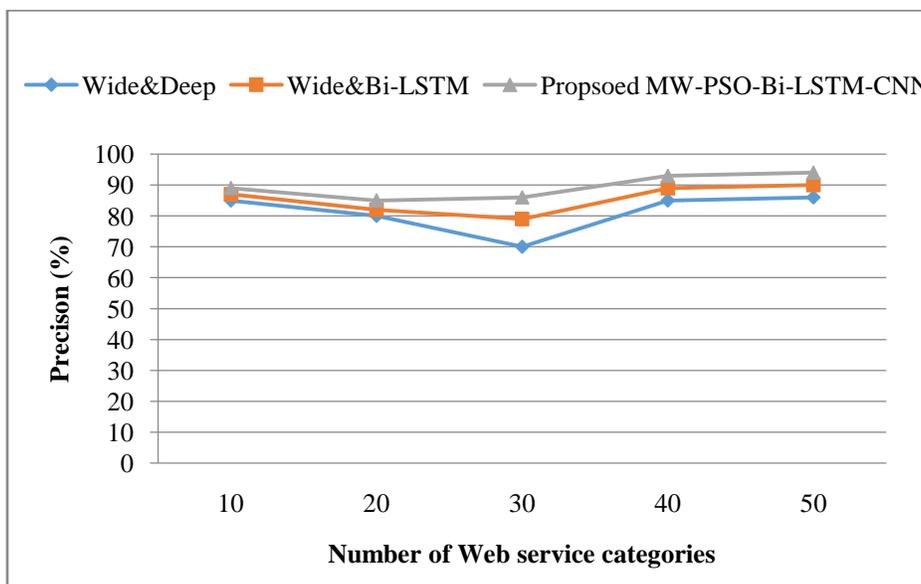


Fig.2 Result of Precision

Precision comparison of existing Wide&Bi-LSTM and Wide&Deep and proposed MW-PSO-Bi-LSTM-CNN is shown in figure 2. The dataset used has various Web service categories. The value of precision is increased with the increase in web service categories. Around 94% of the precision rate is produced in the proposed MW-PSO-Bi-LSTM-CNN technique which is a greater value when compared with existing techniques precision rate as shown in that graph. This is due to the fact that highly accurate features cannot be identified by existing techniques because of back propagation issues. So, better classification results can be achieved using proposed MW-PSO-Bi-LSTM-CNN technique, where PSO is used for selecting hyper parameters.

2) Recall Comparison

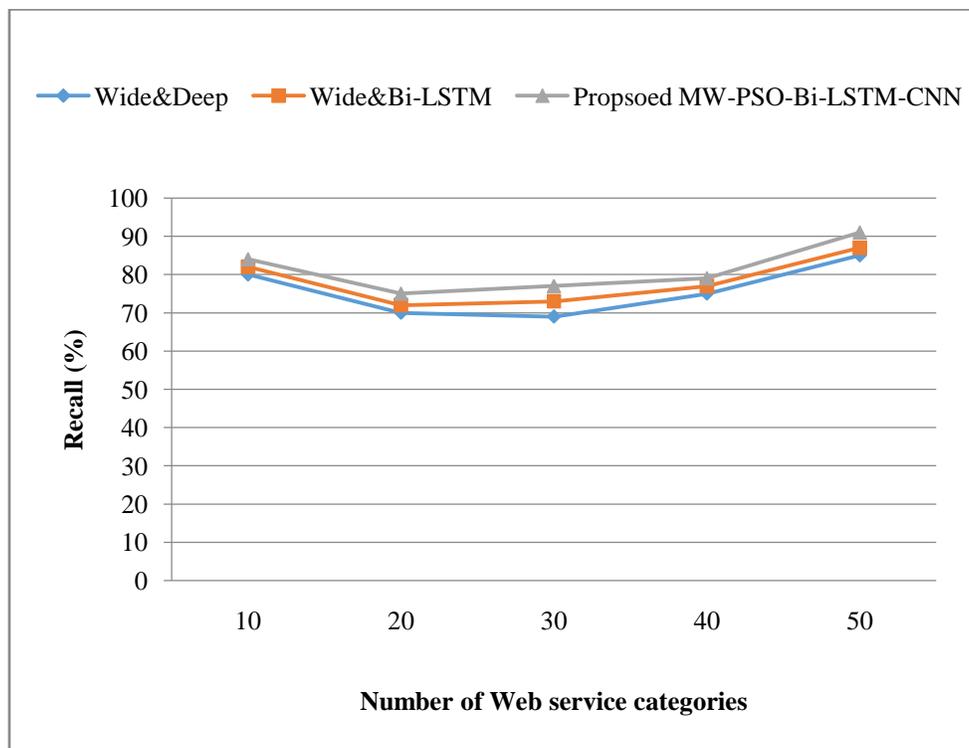


Fig.3 Result of Recall Rate

Recall comparison of existing Wide&Bi-LSTM and Wide&Deep and proposed MW-PSO-Bi-LSTM-CNN is shown in figure 3. The dataset used has various Web service categories. Value of recall is increased with the increase in web service categories. Around 91% of recall value is produced in the proposed MW-PSO-Bi-LSTM-CNN technique which is a greater value when compared with existing techniques recall value as shown in that graph. So, better classification results can be achieved using the proposed MW-PSO-Bi-LSTM-CNN technique.

In existing algorithms, adjustment of weights from very low to very high causes network paralysis. Various repeated presentations of input patterns are considered in proposed MW-PSO-Bi-LSTM-CNN and there is a need to adjust weights before the network is settling down into an optimum solution. In the proposed algorithm, the recall percentage is enhanced.

3) Accuracy

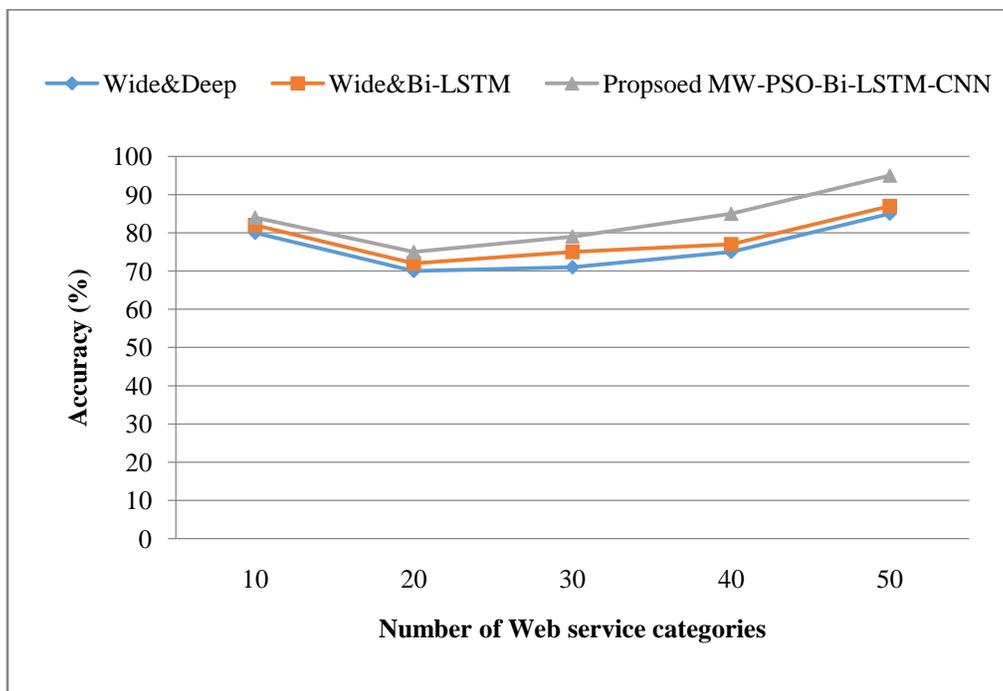


Fig.4 Result of Recall Rate

Accuracy comparison of existing Wide&Bi-LSTM and Wide&Deep and proposed MW-PSO-Bi-LSTM-CNN is shown in figure 4. The dataset used has various Web service categories. Value of recall is increased with the increase in a number of images. Around 95% of recall, value is produced in proposed MW-PSO-Bi-LSTM-CNN technique which is a greater value when compared with existing techniques recall value as shown in that graph. So, better classification results can be achieved using proposed MW-PSO-Bi-LSTM-CNN technique.

This is due to the fact that a feed-forward neural network is used as a deep component in the Wide&Deep model. In Web services, a description document's context information and word order cannot be mined using this. With context information and semantic measure, these words can be mined using MW-PSO-Bi-LSTM-CNN, which significantly enhances the Web service classification accuracy.

4) F-measure

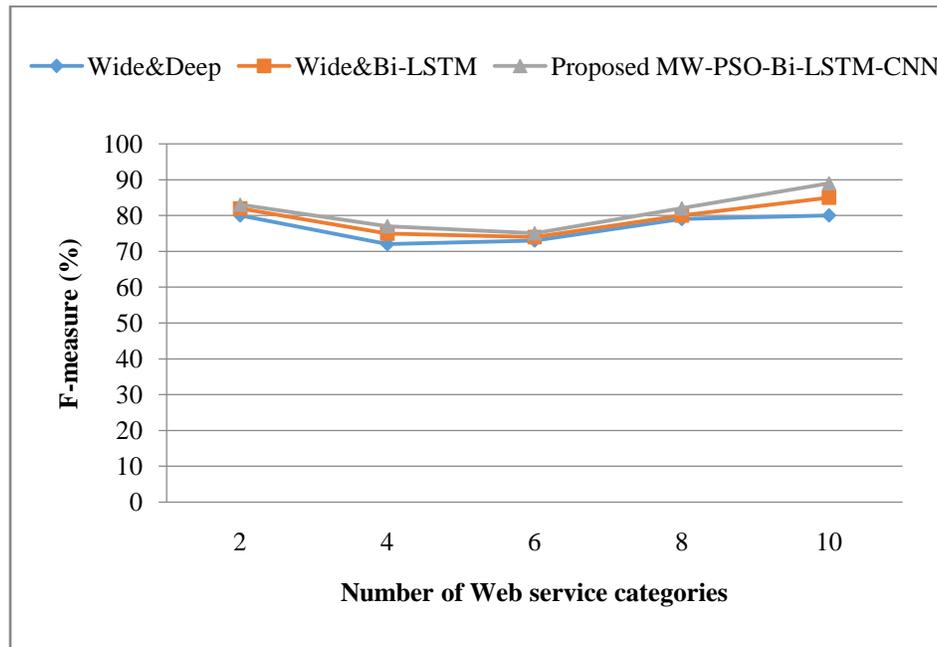


Fig.5 Result of F-measure Rate

F-measure comparison of existing Wide&Bi-LSTM and Wide&Deep and proposed MW-PSO-Bi-LSTM-CNN is shown in figure 5. The dataset used has various Web service categories. Value of f-measure is increased with the increase in the number of data. Around 89% of f-measure value is produced in proposed MW-PSO-Bi-LSTM-CNN technique which is a greater value when compared with existing techniques f-measure value as shown in that graph. So, better classification results can be achieved using proposed the MW-PSO-Bi-LSTM-CNN technique. In order to make a balance between Recall and Precision, F1 score is used as the best measure, especially in case of uneven class distribution with a huge amount of actual negative. Between Web service description documents feature vectors, interactions are captured using this linear model and adaptive characteristics and nonlinearity are added to this model, which makes highly accurate and reasonable web service classification.

Conclusion and Future Work

An MW-PSO-Bi-LSTM-CNN model based Web service classification technique is proposed in this work. The wide learning model is exploited for performing the Web service category's breadth prediction. Between Web service description document feature vectors, interactions are captured using this exploitation. The PSO-Bi-LSTM-CNN model is used for predicting Web service category's depth, which used for mining word order as

well as words context information in Web services description documents. The Bi-LSTM-CNN parameters are optimized using PSO in the PSO-Bi-LSTM-CNN model. The neural network's output error is assumed as PSO fitness in this for computing Bi-LSTM-CNN neural network's optimum weights. In order to optimize the LSTM-CNN neural network using PSO algorithm, the global optimum value is assumed as an initial value, which establishes a PSO-Bi-LSTM-CNN algorithm-based neural network. At last, Web service categories prediction about depth and breadth are integrated for achieving final Web service classification.

Programmable Web dataset is used for making comparative experimentation and proposed method's effectiveness is demonstrated using this comparative analysis. With respect to F-measure, recall and precision, the proposed method significantly enhances the Web service classification accuracy. In future, large scale service network can be constructed by investigating as well as exploiting link information or service relationship to facilitate Web service classification with other deep learning technique.

References

- Sheng, Q.Z., Qiao, X., Vasilakos, A.V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. *Information Sciences*, 280, 218-238.
- Zimmermann, O., Tomlinson, M., & Peuser, S. (2012). *Perspectives on web services: applying SOAP, WSDL and UDDI to real-world projects*. Springer Science & Business Media.
- Kasmi, M., Jamoussi, Y., & Ghézala, H.H.B. (2016). Recommendation in interactive web services composition: a state-of-the-art survey. *In International Conference on Electronic Commerce and Web Technologies, Springer, Cham*, 170-182.
- Yang, J., & Zhou, X. (2015). Semi-automatic web service classification using machine learning. *International Journal of u-and e-Service, Science and Technology*, 8(4), 339-348.
- Fanjiang, Y.Y., Syu, Y., Ma, S.P., & Kuo, J.Y. (2015). An overview and classification of service description approaches in automated service composition research. *IEEE Transactions on Services Computing*, 10(2), 176-189.
- Yang, Y., Ke, W., Wang, W., & Zhao, Y. (2019). Deep Learning for Web Services Classification. *In IEEE International Conference on Web Services (ICWS)*, 440-442.
- Liu, X., Agarwal, S., Ding, C., & Yu, Q. (2016). An LDA-SVM active learning framework for web service classification. *In IEEE International Conference on Web Services (ICWS)*, 49-56.
- Liu, J., Tian, Z., Liu, P., Jiang, J., & Li, Z. (2016). An approach of semantic web service classification based on Naive Bayes. *In IEEE International Conference on Services Computing (SCC)*, 356-362.

- Viji, C., Raja, J.B., Ponmagal, R.S., Suganthi, S.T., Parthasarathi, P., & Pandiyan, S. (2020). Efficient Fuzzy based K-Nearest Neighbour Technique for Web Services Classification. *Microprocessors and Microsystems*, 103097.
- Wang, H., Shi, Y., Zhou, X., Zhou, Q., Shao, S., & Bouguettaya, A. (2010). Web service classification using support vector machine. *In 22nd IEEE International Conference on Tools with Artificial Intelligence, 1*, 3-6.
- Venkatachalam, K., Balakrishnan, S., Prabha, R., & Premnath, S.P. (2018). Effective Feature Set Selection and Centroid Classifier Algorithm for Web Services Discovery. *International Journal of Pure and Applied Mathematics*, 119(12), 1157-1172.
- Lizarralde, I., Mateos, C., Zunino, A., Majchrzak, T.A., & Grønli, T.M. (2020). Discovering web services in social web service repositories using deep variational autoencoders. *Information Processing & Management*, 57(4), 102231.
- Kumar, V.V., & Satyanarayana, N. (2016). Self-Adaptive Semantic Classification using Domain Knowledge and Web Usage Log for Web Service Discovery. *International Journal of Applied Engineering Research*, 11(6), 4618-4622.
- Rodriguez-Mier, P., Pedrinaci, C., Lama, M., & Mucientes, M. (2015). An integrated semantic web service discovery and composition framework. *IEEE transactions on services computing*, 9(4), 537-550.
- Wang, H., Yang, Z., & Yu, Q. (2017). Online Reliability Prediction via Long Short Term Memory for Service-Oriented System. *In IEEE International Conference on Web Services (ICWS). IEEE, Honolulu, Hawaii, USA*, 81-88.
- Ye, H., Cao, B., Peng, Z., Chen, T., Wen, Y., & Liu, J. (2019). Web services classification based on wide & Bi-LSTM model. *IEEE Access*, 7, 43697-43706.
- Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., & Anil, R. (2016). Wide & deep learning for recommender systems. *In Proceedings of the 1st workshop on deep learning for recommender systems*, 7-10.