HYBRID PARTICLE SWARM OPTIMIZATION MULTI LAYER PERCEPTRON FOR WEB-SERVICES CLASSIFICATION

A.Syed Mustafa[1], Dr. Y.S. Kumaraswamy [2]

[1] Dept. of Computer Science and Engineering, Sathyabama University, Chennai, India
[2] Dept. of CSE, Nagarjuna College of Engineering & Technology, Bengaluru, India
Email: [1] Syedmustafa_blr@yahoo.com

Abstract

The Web services are applications that perform specific tasks and are accessible via the network through a communication protocol. QoS plays an important role in finding out the performance of web services. Multi-layer perceptron neural network (MLP) is the most popular and widely used nonlinear network for solving many practical problems in applied science, biology, and engineering. In this paper, hybrid Particle Swarm Optimization (PSO) with MLP is performed for medical web-service classification. MLP performance is based on initial weights setting. Conventional training algorithms like Back propagation (BPP) and Levenberg- Marquardt (LM) have slow convergence and local minima problems. Results show that the proposed method performs better accuracy, average precision, average recall and RMSE.

Key words: Web services, Multi-layer perceptron (MLP), hybrid Particle Swarm Optimization (PSO) with MLP, Back propagation (BPP) and Levenberg- Marquardt (LM).

I. INTRODUCTION

In a web service composition issue [1, 2], the question of how to construct a composite web service scheme according to its Quality of Service (QoS) has become a research focus in recent years. Due to the growing number of candidate services that provide the same functionality but differ in QoSs, it brings more challenges to select a combination of services with optimal QoS performance, while satisfying users' QoS constraints. An execution path of composite service can be constructed by a sequence of tasks including an initial task and a terminal task. There are various service compositions for each execution path of composite service. Moreover, while the number of candidate services is increasing with the proliferation of web services, the size of service composition will become larger and larger.

QWS dataset consists of different web service implementations and their attributes. The classification is measured based on the overall quality rating provided by all the attributes. The functionality of the web services can be helpful to differentiate between various services. The attributes G1 to G10 are used as explanatory variables and the attribute G11 is used as the target variable. The web services in the QWS dataset are classified into four categories, such as: Platinum (high quality); gold; silver and bronze (low quality). The classification is measured based on the overall quality rating provided by WSRF. It is grouped into a particular web service based on classification. The functionality of the web services can be helpful to differentiate between various services [3].

The problem with web page classification is split into multiple sub-problems like functional classification and other types. Subject classification is about a subject or topic of a web page. Based on number of problem classes, classification is divided into binary classification and multi class classification. Binary classification classifies instances into one or two classes whereas multi class classification classifies more than two classes [4]. MLPNN is a different paradigm to compute and an inspiration from neuroscience. These are effective to predict events when networks have large databases.

In order to classify the services, a number of classifiers are applied over the vector set obtained in previous step. The classifiers are Naive-Bayes, Decision Tree, Random Forest, Bagging and Regression. Nave-Bayes classifier is a simple probabilistic classifier which applies Bayes theorem and assumes strong independence between the features [5]. Decision tree

classifier is based on using a decision tree for building a predictive model. Here, numerous test questions and conditions are taken in a tree structure. Each internal node and the root of the tree denote a testing attribute while each leaf will correspond to a class label.

The goal of this work is to present and evaluate an approach based on Particle Swarm Optimization (PSO) that overcomes this limitation and enables Web service composition and selection with regards to functional correctness and Quality of Service properties. In this work, MLP based PSO is proposed for Web service composition approach that addresses the limitation of existing approaches. Unlike the other approaches, the one shown here does not require the selection of an initial configuration, and thus does not depend on users with domain expertise.

PSO with MLP is proposed for medical web service classification. Section 2 shows the literatures related to proposed work, section 3 shows the methods and techniques need for work, section 4 shows the results and discussions with the results obtained and finally section 5 concludes the work.

II. LITERATURE SURVEY

Mohanty et al[6] employed Naïve Bayes, Markov blanket and Tabu search to rank web services. The Bayesian Network is demonstrated on a dataset taken from literature. The dataset consists of 364 web services whose quality is described by 9 attributes. Here, the attributes are treated as criteria, to classify web services. From the experiments results show that Naïve based Bayesian network performs better than other two techniques comparable to the classification done in literature.

Karande & Kalbande [7] addressed the choosing of web services through tModel of SOA that is formulated through feed forward network. The designing is carried out through XML language. Through supervised learning technique of feed forward neural networks, ontologies of various domains may be matched. Feed forward neural networks may be utilized for pattern matching with back propagation methods. Patterns defined here are quality variables. The quality variable may be chosen through tModel structure of UDDI. Web Service Providers in UDDI are capable of differentiating services through Quality Categorization through labeling of qualities, that is, performance, security and so on. This differentiation may be performed through QoS ontology for Service Identifications. ANN matching models comprise of training phases as well as matching phases on the basis of ontology domains.

Silva et al [8] presented a graph-based PSO technique which simultaneously determines the optimal workflow and the optimal Web services to be included in the composition based on their QoS properties, as well as a greedy-based PSO technique which follows the commonly utilized approach. The comparison of the two techniques shows that despite requiring more execution time, the graph-based approach provides equivalent or better solutions than the greedy-based approach, depending on the workflow preselected by the greedy-based PSO. These results demonstrate that under certain circumstances, the graph-based approach is capable of producing solutions whose fitness surpasses that of the solutions obtained by employing the greedy-based approach.

Kamath et al [9] proposed an approach for web service classification based on conversion of services into a class dependent vector by applying the concept of semantic relatedness and to generate classes of services ranked by their semantic relatedness to a given query. Proposed method used the OWLS-tc service data set for evaluating our approach and the experimental results are presented in this work.

Multi-Layer Perceptron Neural Network (MLPNN) is used for classification problems. Mustafa& Swamy [10] proposed a Multi-Layer Perceptron optimized with Tabu search (MLP-TS) for learning. Experimental results demonstrate that the proposed MLP-TS outperforms Multi-Layer Perceptron-Levenberg-Marquardt (MLP-LM) and Multi-Layer Perceptron Back Propagation (MLP-BPP) for web service classification.

QoS-aware binding of composite services yields the capacity of binding all service invocations in composition to services selected amongst a set of functionally equivalent ones for achieving QoS goals. But current methods do not consider computation time. Zhang [11] proposed a fast QoS- aware web service selection approach. This approach adopts a particle swarm optimization algorithm select the most service with users'

QoS requirements Experimental results show that our approach can find best suitable services with lower time cost than other approaches in web service selection.

III. METHODOLOGY

Classifiers such as random forest, multi-layer perceptron is used with Levenberg-Marquardt (LM) algorithm and particle swarm optimization algorithm.

A. Random Forest (RF)

Random forests classifier is an ensemble classifier which involves growing many classification trees [12]. Each object is classified by passing its input vector to each of the trees. Each tree assigns a class label to the object. This process is called as voting as each tree votes for a class label. The forest chooses the class label which has the highest number of votes. Bagging is another ensemble classifier that applies base classifiers on random subsets of the original dataset. The subsets are made by drawing random samples and replacing them later. Each base classifier gives their individual predictions and these are then aggregated either by averaging or voting. Regression is a probabilistic statistical model which uses the relationship between the categorical dependent variable (which needs to be predicted) and various independent variables.

B. Multi-Layer Perceptron (MLP)

MLP, also known as feed forward neural networks, is used for information processing and pattern recognition in seismic activities prediction. It is a feed forward Artificial Neural Network (ANN) model that maps input data sets to a set of appropriate output. It is a variation of standard linear perceptron which uses 3 or more node layers with nonlinear activation functions and is powerful than a perceptron as it distinguishes data not linearly separable or separable using a hyper plane. Neurons for MLP input layer pattern classification is determined by features representing relevant feature space patterns. Input layer neurons, acting as sensory units, compute identity function, y = x. A hidden layer neuron and output layers compute sigmoidal function of sum of products of input values and weight values of corresponding connections.

If "O_k and t_k pair constitutes the input and output parameters in the training data set and a sigmoid function is used, the output of the kth neuron, O_k, in nth layer" is computed as in equation (1):

$$O_k = \frac{1}{1+\exp(-net_k)}, net_k = \sum_j W_{jk} O_j \quad (1)$$

And "O_j represents the output of a neuron from the previous layer and W_{jk} is weight between neurons jth neuron and kth neuron". The adjustable network parameters are optimized based on the BP algorithm as in equation (2):

$$W_{jk}^{new} = W_{jk} + \Delta W_{jk} \quad (2)$$

Where "ΔW_{jk} is weight update for the connection W_{jk}". The weight update is obtained as in equation (3):

$$\Delta W_{jk} = -\eta\left(\frac{\partial E}{\partial W_{jk}}\right) \quad (3)$$

Where "η is the learning rate which is selected between 0 and 1 and E is the error defined" as in equation (4):

$$E = \frac{1}{2}\sum_k (t_k - O_k)^2 \quad (4)$$

Where "t_k is the target for the kth neuron in the output layer".

C. Back Propagation (BP) Network

MLP with BPP algorithm is a standard algorithm for supervised learning pattern recognition process. For the MLP neural network trained with BPP algorithm, a complex network's error surface is full of hills and valleys. Due to gradient descent a network can be trapped in local minimum when a deeper minimum is nearby. Probabilistic methods help avoid this trap, but they are slow. Another possibility is increasing the hidden units. Though this works because of error space's higher dimensionality and chances of being trapped is smaller, there is an upper limit to hidden units which, when exceeded, result in system being trapped in local minima [13]. Nodes, and K be Number of output nodes for MLP NN. Consider V to

be hidden layer weight vector and W weight vector for output layer. The size of W matrix is K X J. Size of V matrix is J X I Training cycle steps are [14]:

1. By applying feature vectors one by one to input layer output of hidden layer is computed as $y_i = f_1(v_j^t z)$ for j=1,2,….,J function $f_1(.)$ is unipolar Sigmoid Function defined by using (1). Output of output layer is computed by using $O_k = f_2(w_k^t y)$ for j=1,2,….,k function $f_2(.)$ is Generalized Sigmoid Function defined by using

2. Error value is computed by using $E = \frac{1}{2}(d_k - o_k)^2 + E$ for k=1,2,…..,k

3. Error signal vectors δ_0 and δ_{0y} of both layers are computed. Dimension of Vector is $\delta_0(K \times 1)$ and dimensions of $\delta_y(\mathrm{J}\times 1)$. Error signal terms of output layer is given by using $\delta_{ok} = (d_k - o_k)(1-o_k)o_k$ for k=1,2,…..,k error signal terms of hidden layer is given by using $\delta_{yi} = y_i(1-y_i)\sum_{k=1}^{k}\delta_{ok}w_{kj}$ for j=1,2,….,J.

4. Output layer weights are adjusted as $w_{kj} = w_{kj} + \eta\delta_{ok}y_i$ for k=1,2,….,k and j=1,2,….,J

5. hidden layers weights are adjusted by using $v_{ij} = v_{ij} + \eta\delta_{yj}z_i$ for j=1,2,….,J and i=1,2,….,I

6. Repeat steps 1) to 5) for all feature vectors Training cycle is repeated till the solution obtained.

D.Levenberg-Marquardt(LM) algorithm

Levenberg-Marquardt (LM) algorithm achieve second- order training speed without computing a Hessian matrix. When performance function for training feed forward networks has form of a sum of squares, then Hessian matrix is approximated as $H = J^T J$ and gradient can be computed as T $g = J^T e$ where J is Jacobian matrix, having first derivatives of network errors regarding weights and biases, and e is a network errors vector. The Jacobian matrix is computed through a standard back-propagation technique that is less complex than computing a Hessian matrix. LM algorithm uses this approximation to Hessian matrix in following Newton-like update [15].

E.Particle Swarm Optimization (PSO)

PSO [16] is a swarm intelligent techniques inspired by birds flocking. As a population based evolutionary techniques, each particle of PSO searches the domain space with position and velocity information and preserves the best position. Each particle enhances itself by keeping the tracks of two optimal solution found by the particle swarm. PSO is a technique that evolved from modelling the behaviour of groups of social animals, such as flocking birds and schooling fish [17]. The intuition behind this technique is that particles independently explore the search space and communicate with each other to identify the best possible solution — that is, the best possible search space location. PSO is a relatively simple technique to implement and does not require expensive computations.

A standard particle swarm optimizer maintains a swarm of particles and each individual is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position x_i the previous best position p_i and the velocity v_i. The current position $x_i = (x_{i,1}, ..., x_{i,\mathrm{D}})$ can be considered as a set of coordinates describing a point in space. New points are chosen by adding $v_i = (\mathrm{v}_{i,1}, ..., \mathrm{v}_{i,\mathrm{D}})$ co-ordinates to x_i and the algorithm operates by adjusting v_i which can be seen as a step size.

In essence, the trajectory of each particle is updated according to its own flying experience as well as to that of the best particle in the swarm. Experimental results suggest that it is preferable to initialize the inertia weight to a large value (usually less than 1), giving priority to global exploration of the search space, and gradually decreasing so as to obtain refined solutions.

$$v_{i,d}^{k+1} = \omega v_{i,d}^k + c_1 r_1^k(\mathrm{p}_{i,d}^k - x_{i,d}^k) + c_2 r_2^k(\mathrm{p}_{g,d}^k - x_{i,d}^k)$$
$$x_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1} \quad (5)$$

Where $v_{i,d}^{k}$ is the d-th dimension velocity of particle i in cycle k; $x_{i,d}^{k}$ is the d-th dimension position of particle i in cycle k; $\mathrm{p}_{i,d}^{k}$ is the d-th dimension of personal best (pbest) of particle i in cycle k; $\mathrm{p}_{\mathrm{g},d}^{k}$ is the d-th dimension of global best (gbest) of particle i in cycle k; ω is the inertia weight; c_1 is the cognition weight and c_2 is the social weight; and r_1 and r_2 are two random values uniformly distributed in the range of (0,1).

3.6 Multi-Layer Perceptron-Particle Swarm Optimization (MLP PSO)

A method based on PSO to adjust solely the MLP weights, combined with a weight decay heuristics, is proposed. Figure 1 shows the pseudo code for the MLP-PSO.

Particles initialization: the algorithm begins with a random generation of n particles. The velocities, dimensions relative to weights, connections and bias are set to random values generated by a uniform distribution on interval [-1, 1]. In this initialization, both hidden neurons and connections are deactivated.

Stopping criteria: the execution stops if the number of iterations reaches a certain maximum number (MAX_ITER) or if the current generalization loss (gLoss) is greater than 5% or if the validation error does not improve during 300 iterations.

Pbest and Gbest selection: The classification error on training set is used as fitness function, thus we select Pbest and Gbest in terms of this measure.

It is necessary to note that the velocity of each particle is limited to a range $[V_{\min}, V_{\max}]$ to avoid that particles fly out of the search space.

Early stopping calculation: after the 500 initial iterations, in every 100 iterations Generalization Loss (GL) is calculated based on the validation error of the current best network in the swarm ($vnet_{curr}$) and the validation error of the best historical network ($vnet_{opt}$), i.e. considering the best network in previous iterations.

Selection of the final best network: after the stopping criterion is reached, the network in $vnet_{opt}$ is returned. This network has the lower classification error on the validation set.

```
1:  randomly initialize the population
2:   while i< MAX_ITER and gLoss< 5 and
3:        stagnationCount< 3 do
4:        for each particle Pi in the swarm do
5:        choose the Pi local best Pbest with
              lower training error
6:        choose the swarm best Gbest with
              lower training error
7:         update position and velocity of Pi
8:    end for
9:         if i> 500 and i mod 100 = 0 then
10:         if  error(vnet_curr ) <  error(vnet_opt ) then
11:           vnet_opt = vnet_curr
12:           stagnationCount = 0
13:   else
14:         gLoss=
      generalizationLoss(vnet_opt , vnet_curr )
15:    stagnationCount = stagnationCount + 1
16:  end if
17:    end if
18: end while
19: return vnet_opt
```

IV. RESULTS AND DISCUSSION

For experiments, techniques such as random forest, MLP with LM and PSO is used. Table 1 shows the summary of results. Figure 1 to 4 shows the results of classification accuracy, average precision, average recall and RMSE respectively.

Table 1 Summary of Results

	Random forest	MLP-LM	MLP-PSO
Classification accuracy	82.19	95.34	96.72
Avg Precision	0.8224	0.9533	0.9673
Avg Recall	0.8225	0.9534	0.9674
RMSE	0.2648	0.1247	0.1184

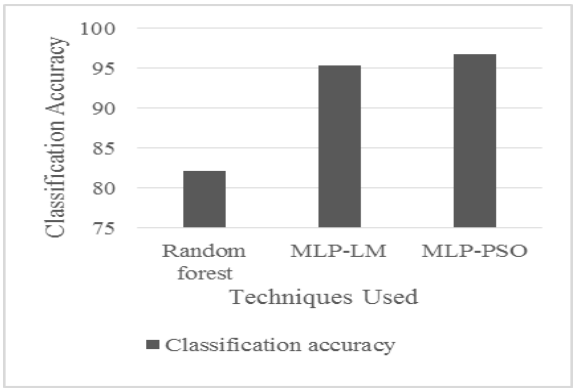

Figure 1 Classification Accuracy

From table 1 and figure 1 it is observed that the classification accuracy of MLP-PSO performs better by 16.24% than random forest and by 1.44% than MLP-LM.

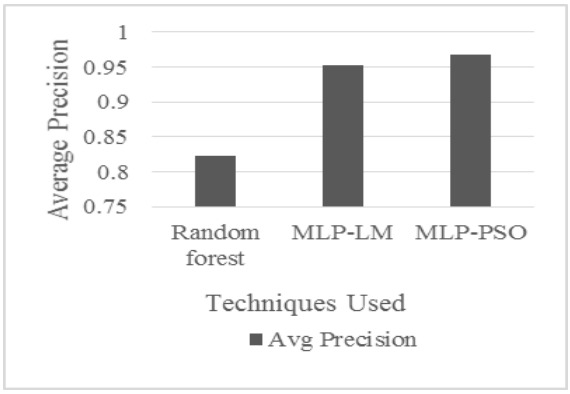

Figure 2 Average Precision

From table 1 and figure 2 it is observed that the average precision of MLP-PSO performs better by 16.19% than random forest and by 1.46% than MLP-LM.

From table 1 and figure 3 it is observed that the average recall of MLP-PSO performs better by 16.19% than random forest and by 1.46% than MLP-LM.

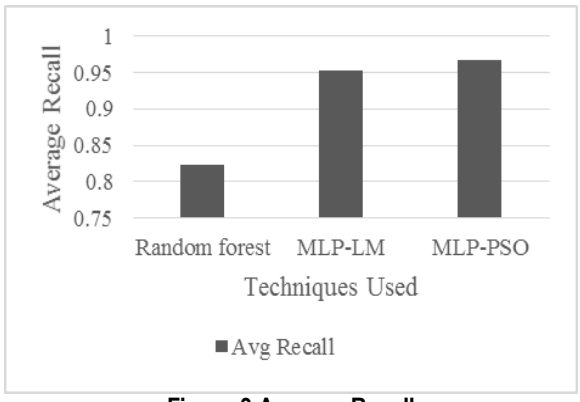

Figure 3 Average Recall

From table 1 and figure 4 it is observed that the RMSE of MLP-PSO performs better by lowering RMSE value by 76.41% than random forest and by 5.18% than MLP-LM

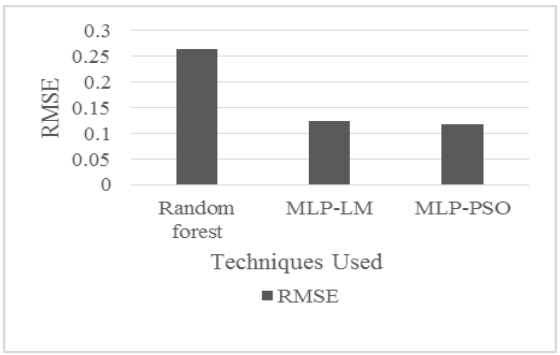

Figure 4 RMSE

V. CONCLUSION

Service composition enables the reuse of Web services for solving different problems, leading to faster and intuitive development of service-oriented solutions. PSO algorithm is proposed to contribute the following issues. The improved local best first strategy assures that the component weights in the local fitness of a task and the fitness of a composite web service are equivalent. Thus the relative importance of QoS attributes of a composite web service is as the same as the size of candidate services. Thus the local improvement of a web service can guide the search of algorithm properly when a better candidate service is selected to finish the corresponding task. Results show that the classification accuracy of MLP-PSO performs better by 16.24% than random forest and by 1.44% than MLP-LM. As a future work, investigation can be of using the other variations of PSO proposed in the literature in the task of training neural networks.

REFERENCES

[1] Cutello, V., Nicosia, G., & Pavone, M. (2004). Exploring the capability of immune algorithms: A characterization of hypermutation operators. In Artificial Immune Systems (pp. 263-276). Springer Berlin Heidelberg.

[2] Zhao, X., Song, B., Huang, P., Wen, Z., Weng, J., & Fan, Y. (2012). An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition. Applied Soft Computing, 12(8), 2208-2216.

[3] O. Ardaiz, F. Freitag and L. Navarro, "Improving Service Time of Web Clients Using Server Redirection," ACM SIGMETRICS Performances Evaluation Review, Vol. 29, No. 2, 2001, pp. 39-44.

[4] Kaur, P. (2014). Web Content Classification: A Survey. arXiv preprint arXiv:1405.0580.

[5] P. Murphy, "Naive bayes classifiers," University of British Columbia, 2006.

[6] Mohanty, R., Ravi, V., & Patra, M. R. (2012). Classification of web services using Bayesian network.

[7] Karande, A. M., & Kalbande, D. R. (2014, February). Web service selection based on QoS using tModel working on feed forward network. In Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on (pp. 29-33). IEEE.

[8] da Silva, A. S., Ma, H., & Zhang, M. (2014, July). A graph-based particle swarm optimisation approach to qos-aware web service composition and selection. In Evolutionary Computation (CEC), 2014 IEEE Congress on (pp. 3127-3134). IEEE.

[9] Sowmya Kamath, S., Ahmed, A., & Shankar, M. (2015, June). A composite classification model for web services based on semantic & syntactic information integration. In Advance Computing Conference (IACC), 2015 IEEE International (pp. 1169-1173). IEEE

[10] Syed Mustafa, A., & Swamy, K. (2015, June). Web Service classification using Multi-Layer Perceptron optimized with Tabu search. In Advance Computing Conference (IACC), 2015 IEEE International (pp. 290-294). IEEE.

[11] Zhang, T. (2014). QoS-aware web service selection based on particle swarm optimization. Journal of Networks, 9(3), 565-570.

[12] Breiman, "Random forests," Machine learning, vol. 45, no. 1, pp. 5–32, 2001.

[13] Rui, L., Xiong, Y., Xiao, K., & Qiu, X. (2014, September). BP neural network-based web service selection algorithm in the smart distribution grid. In Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific (pp. 1-4). IEEE.

[14] Alsmadi,M. K. S.,Omar, K. B.,&Noah,S. A, "Back propagation algorithm: the best algorithm among the multi-layer perceptron algorithm". International Journal of Computer Science and Network Security,(2009), 9(4), 378-383.

[15] KISI,Ö.,& Uncuoglu, E., "Comparison of three back- propagation training algorithms for two case studies" Indian journal of engineering & materials sciences,(2005), 12(5), 434-442.

[16] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intelligence 1 (2007) 33–57.

[17] J.Kennedy, "Particle swarm optimization," in Encyclopedia of Ma- chine Learning. Springer, 2010, pp. 760–766.