

# Quality Assessment of Web Services using Soft Computing Techniques

<sup>1</sup>V. Mohan Patro, <sup>2</sup>Bonomali Khuntia and <sup>3</sup>Manas Ranjan Patra

Computer Science Department, Berhampur University, India

<sup>1</sup>vmpatro@gmail.com; <sup>2</sup>bonomalikhuntia@gmail.com; <sup>3</sup>mrpatra12@gmail.com

## ABSTRACT

With the ever growing number of web services in the Internet the selection of a suitable web service has become a mind boggling task. Several quality parameters are being considered for a judicious selection of a web service in a given service context. One of the popular measures for classifying web services is the use of Web Service Relevance Function (WSRF). A number of classifiers have been used which have yielded accuracy up to 99% considering WSRF as one of the attributes of QWS dataset. But, the maximum accuracy achieved by any classifier without WSRF is only 89.99%. In this paper, a feed-forward backpropagation neural network (BPNN) with adaptive momentum factor has been employed to further improve the accuracy. Furthermore, a similar neural network is modelled for determining the WSRF of the web services included in the QWS dataset. As an alternative, Genetic Algorithm is used to find the weight factors associated with each parameter to calculate WSRF, assuming it to be a linear function of those parameters. The average errors for WSRF per pattern obtained in the two approaches are 1.0447 and 1.176 respectively. These calculated WSRF values can be used for classification to enhance the accuracy of a classifier.

**Keywords:** Web services, Quality of Services (QoS), Backpropagation, Continuous Genetic Algorithm

## 1 Introduction

Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web [1]. A Web Service is a means of communication between two electronic devices over a network. It is a software function provided at a network address over the web and represents a software system designed to support interoperable machine-to-machine interaction over a network. Different software systems often need to exchange data with each other, and a web service is a means of communication that allows two software systems to exchange data over the internet. The software system that requests data is called a service requester, whereas the one that processes the request and provide the service is called a service provider. An increasing amount of companies and organizations only implement their core business and outsource other application services over the Internet. So the ability to efficiently and effectively select and integrate inter-organizational, heterogeneous services on the Web at runtime is an important step towards development of Web service applications.

The web services, a novel paradigm in software technology, have innovative mechanism for rendering services over diversified environment. They promise to allow businesses to adapt rapidly to changes in the business environment and the needs of different customers. With the increasing use of web services, standardization of basic content integration, support of complex service-oriented architectures, provision of seamless integration of business processes and applications etc. has led to an increase in the number of both web service consumers and providers [2]. The rapid

introduction of new web services into a dynamic business environment can adversely affect the service quality and user satisfaction. Consequently, assessment of the quality of web services is of paramount importance in selecting a web service for an application. From a service consumer's perspective, Quality of Service (QoS) plays a crucial role while selecting a particular web service from among many alternatives found in the UDDI registry.

## 2 Related Work

In paper [3], Mohanty et al. developed various classification models based on intelligent techniques namely BPNN, PNN, GMDH, TreeNet, CART, SVM and J48 to predict the quality of a web service based on a number of QoS attributes. They observed that the accuracy with WSRF for GMDH and J48 is maximum i.e. 100%, where the accuracies without WSRF for GMDH and J48 are 89.75% and 67.77% respectively. In case of PNN accuracy without WSRF is maximum i.e. 89.99%. Therefore, there is significant difference in accuracies of classifiers with WSRF and without WSRF. Also, the classification accuracy greatly depends on the WSRF value which helps GMDH and J48 to approach accuracies up to 100%. However, the accuracies of all those techniques fall down miserably while not considering the WSRF as one of the input attributes.

Li Yuan-jie et al.[4] compared 3 classification algorithms, Naïve Bayes, SVM, REPTree, in classifying the WSDL data, and then analyzed the ensemble learning classification trained by AdaBoost. In this paper, they applied automatic web service semantic annotation and used these three classification methods and furthermore ensemble learning is applied. According to the experimental results using 951 WSDL files and 19 categories, the accuracy was 87.39%.

In [5], authors employed Naïve Bayes, Markov blanket and Tabu search techniques to classify web services. They noted that the average accuracy of Naïve Bayes classifier is 85.62%, followed by Tabu search of 82.45% and Markov blanket of 81.36%. In this context, they employed Back propagation trained neural network to find the importance of different attributes in web services and they found that WSRF plays a vital role for classifying the web services. Excluding the WSRF from dataset they observed that the average accuracy of Naïve Bayes is 75.01%, Markov Blanket is 65.48% and Tabu search is 71.38%.

Web service selection has been extensively studied in [6-11]. Xu et al (2007) [12] have presented a QoS-based web service selection approach that is based on ratings indicating the level of client's satisfaction with a web service following certain interaction with it. A drawback of the approach is the assumption that the ratings are objective and valid. Thus, the web service selection process becomes untrustworthy. Al-Masri and Mahmoud (2007) [13] introduce the notion of web service relevancy function in order to measure the relevancy ranking of a particular web service. The function calculates the distance between a particular QoS value and the maximum normalized value in its corresponding set.

In [14], the QoS is utilized to differentiate a single web service, which best meets client requirements among multiple web services with the same functional properties. The QoS data collected during web service executions is used for prediction of the future values of QoS properties, based on probability evaluation. Thus, the client will be able to select a web service with the largest probability of having QoS properties that are as close as possible to the user defined requirements.

The suitability of ANN for ranking quality of web services has been reported in [15, 16]. In [15] performance of various Artificial Neural Network (ANN) training algorithms in predicting the ranking

of a web service has been reported. In [16], the average performance rate of ANN recorded is 95%. It is also observed that by removing some of the input parameters of ANN, the accuracy degrades significantly.

### 3 Methodology

The ease of implementation and greater generalization capability made ANNs popular among the researchers for various applications. In this paper, ANNs has been implemented to address the problem of accurate determination of WSRF parameter of web services. The back propagation algorithm, which is a gradient decent algorithm, is used for training the network in a supervised manner. The Back Propagation algorithm is widely used in many applications. However, this algorithm is sometimes trapped by local minima which limit its efficiency. To overcome this limitation many variations to the standard algorithm has been proposed. These variations to standard backpropagation include introduction of learning coefficient to speed up the learning. The value of the learning rate should be sufficiently large to allow a fast learning process but small enough to guarantee its effectiveness. To avoid oscillation, momentum factor is introduced with small values of learning coefficient. The performance of the algorithm can further be improved by adaptively changing the learning rate and momentum coefficient. In the first phase of this work, backpropagation algorithm is used for quality assessment of web services. Different neural network structures have been modelled for classification of web services and determination of the web services relevance function. As an alternative approach to compute WSRF value, Genetic Algorithm is used for weight determination for each parameter. The QWS dataset is chosen to formulate the objective function. The total absolute error is minimized by optimizing the weight factors of the web service parameters. A simple Genetic Algorithm is described as follows.

Genetic Algorithm (GA) is a global search method based on natural selection procedure consisting of genetic operators such as selection, crossover and mutation. GA optimizers are particularly effective in a high-dimension, multi-modal function, in which the number of variables tend to be higher, for their easy searching process. GA performs its searching process via population-to-population (instead of point-to-point) search. Parallel architecture of GA makes it robust which uses probabilistic and deterministic rules. A member in a population called a chromosome, is represented by a binary string comprising 0, 1 bits. Bits of the chromosome are randomly selected and the length of bit strings is defined in relevance. However, real values are taken in continuous genetic algorithm. In order to apply the methodology, an initial randomly generated population is required. From initial population, child population is born guided by three operators such as reproduction, crossover and mutation. New born child members are judged by their fitness function values. These child members act as parents in the next iteration. This procedure is repeated till the termination criteria are met.

The pseudo code of a genetic algorithm is as follows:

Simple Genetic Algorithm ( )

```
{
    Initialize the Population;
    Calculate Fitness function;
    While (Fitness Value! = Optimal Value)
    {
        Selection;
        Crossover;
        Mutation;
        Calculate fitness Function;
    }
}
```

## 4 Experimental Set Up

The QWS (Quality of Web Service) dataset is chosen for this purpose. Artificial Neural Network is used for two different purposes like (i) classification of web services and (ii) finding the value of WSRF. Continuous Genetic Algorithm (CGA) is used for finding the values of WSRF.

### 4.1 Data set 1

The QWS dataset consists of data from over 5000 web services out of which the public dataset consists of a random 364 web services which have been chosen. The services were collected using Web Service Crawler Engine (WSCE). The majority of Web services were obtained from public sources on the Web including Universal Description, Discovery, and Integration (UDDI) registries, search engines, and service portals. The public dataset consists of 364 Web services each with a set of nine Quality of Web Service (QWS) attributes that have been measured using commercial benchmark tools. Each service was tested over a ten-minute period for three consecutive days. WSRF is used to measure the quality ranking of a Web service based on the parameters 1 through 9 listed in table-1.

**Table 1: QWS Parameters with their description and units**

| P-ID | Parameter Name         | Description  | Units              |
|------|------------------------|--|--------------------|
| 1    | Response Time (RT)     | Time taken to send a request and receive a response                    | ms                 |
| 2    | Availability (AV)      | Number of successful invocations/total invocations                     | %                  |
| 3    | Throughput (TP)        | Total Number of invocations for a given period of time                 | Invokes per second |
| 4    | Success ability (SA)   | Number of responses / number of request messages                       | %                  |
| 5    | Reliability (REL)      | Ratio of the number of error messages to total messages                | %                  |
| 6    | Compliance (CP)        | The extent to which a WSDL document follows WSDL specification         | %                  |
| 7    | Best Practices (BP)    | The extent to which a Web service follows WS-I Basic Profile           | %                  |
| 8    | Latency (LT)           | Time taken for the server to process a given request                   | ms                 |
| 9    | Documentation (DOC)    | Measure of documentation (i.e. description tags) in WSDL               | %                  |
| 10   | WSRF                   | Web Service Relevancy Function: a rank for Web Service Quality         | %                  |
| 11   | Service Classification | Levels representing service offering qualities (1 through 4)           | Classifier         |
| 12   | Service Name           | Name of the Web service  | None               |
| 13   | WSDL Address           | Location of the Web Service Definition Language (WSDL) file on the Web | None               |

### 4.2 Data set 2

This is a new version of QWS Dataset that includes a set of 2,507 Web services and their QWS measurements that were carried out using the Web Service Broker (WSB) framework [17]. Each row in this dataset consists of 11 parameters separated by commas for each Web service. The first nine parameters are QWS parameters measured using Web service benchmark tools over a six-day period. The QWS values represent averages of the measurements collected during that period. The last two parameters represent the service name and reference to the WSDL document. Example:

67.5,86,6,86,73,78,80,1.5,95,check,http://ws.cdyne.com/spellchecker/check.aspx?wsdl.

### 4.3 Platform

Both the backpropagation neural networks and Continuous Genetic Algorithm are implemented in Pentium i5, 1.80 GHz processor, and 4 GB RAM.

## 5 Results and Discussion

### 5.1 Classification of web services using Neural Networks

Neural networks are being efficiently used for classifying web services. Since WSRF is the most important parameter of web services, two different networks have been considered for this purpose.

In the first case, P-IDs (Parameter-IDs) 1 to 10 from table-1 have been used as the inputs.

A 10x10x1 network is modelled as shown in figure 1. A single output represents the class of the web service.

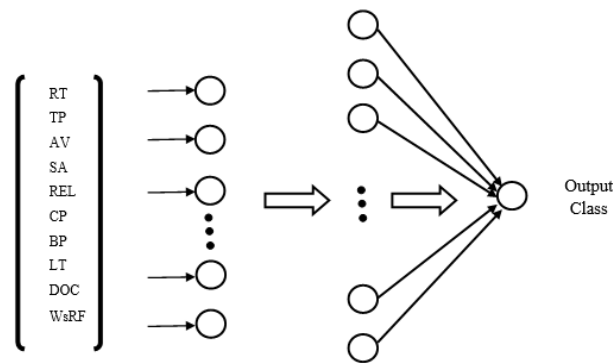


Figure 1: Network Structure

The network is trained for 2000 iterations without momentum coefficient and with variations to momentum coefficient.

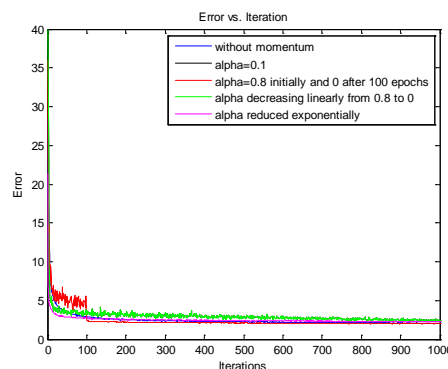


Figure 2: Effect of adaptive momentum coefficient in training NN for classification

The learning constant is taken as 0.04. At first the network is trained without momentum. Then, it is trained with a momentum of 0.1 keeping learning constant 0.04. To introduce adaptive momentum, the momentum variations are considered in three different ways. In the first case, the momentum is kept high at 0.8 for 100 iterations and then reduced to zero. In the second case, the momentum is reduced linearly from high value to zero. Finally, the network is trained with exponentially decreasing momentum coefficient keeping the learning constant at 0.04. From above it is clearly noticeable that use of adaptive momentum improves learning the neural network. It is seen that the

learning with exponentially decaying momentum is more effective. A 10 fold validation technique has been used to train and test the network until maximum classification accuracy is attained. In our case we could achieve 100% accuracy. In the second case, a network structure 9x7x1 is modelled for classification of the web services without WSRF parameter from the inputs in figure 1. A similar adaptive momentum coefficient is considered for training the network. Existing classifiers show maximum 89.99% accuracy without considering WSRF as one of the input parameters. The proposed model with 10-fold validation achieves an accuracy of 97.22%.

## 5.2 Computation of WSRF using Neural Networks

The WSRF is the most important parameter for classifying various web services which depends on other parameters like reliability, Successability, Documentation and Response time etc. However, to our knowledge there is no tool to compute WSRF parameter directly. Even though there exist many detailed guidelines for measuring the quality of web services, there is still a debate about what actually constitute a good web service. Furthermore, there is a lack of empirical validation for good quality web service guidelines. Thus, in our view defining a quality of service metrics is necessary in order to overcome this problem. In this work we have considered the parameters Response time, Throughput, Availability, Successability, Reliability, Compliance, Best Practices, Latency and Documentation to compute the WSRF value using the QWS dataset.

Therefore, a 9x10x1 network is modelled which takes inputs such as Response Time (RT), Throughput (TP), Availability (AV), Successability (SA), Reliability (REL), Compliance (CP), Best Practices (BP), Latency (LT), and Documentation (DOC). The output of the ANN is the WSRF value, which is used to measure the quality ranking of a Web service based on the quality metrics.

The learning coefficient is kept constant at 0.04 and the momentum coefficient is changed adaptively. In order to analyze the effect of adaptive momentum coefficient, the momentum is decremented with training iterations in three ways such as: i) momentum coefficient is decremented linearly from a high value 0.8 to zero, ii) momentum coefficient is kept constant at a high value 0.8 for 100 iterations and zero for the rest, and iii) momentum coefficient is decremented exponentially with iterations.

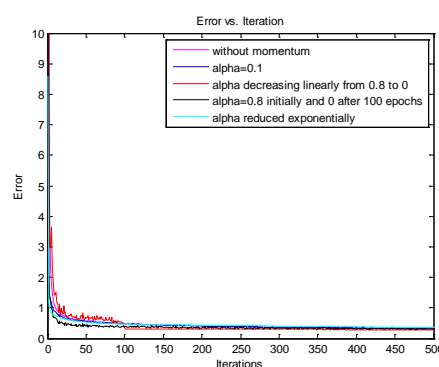


Figure 3: Effect of adaptive momentum coefficient in training NN for WSRF calculation

A sample of calculated WSRF by using neural networks with actual WSRF values is recorded in Table 2. The average error per pattern in testing mode is 1.0447.

**Table 2: WSRF as predicted by ANN**

| Sl. No. | P1      | P2  | P3   | P4 | P5   | P6 | P7 | P8      | P9 | Actual WSRF | Computed WSRF |
|---------|---------|-----|------|----|------|----|----|---------|----|-------------|---------------|
| 1       | 999.71  | 99  | 0.8  | 94 | 38.5 | 78 | 74 | 837.41  | 93 | 67          | 66.51937      |
| 2       | 2987.9  | 56  | 0.2  | 17 | 38.8 | 78 | 82 | 2987.9  | 8  | 39          | 41.1091       |
| 3       | 125.44  | 100 | 13.5 | 86 | 86.4 | 78 | 80 | 125.33  | 91 | 86          | 86.77515      |
| 4       | 326.56  | 84  | 6.2  | 77 | 15.9 | 89 | 89 | 303.92  | 5  | 55          | 56.56019      |
| 5       | 582.36  | 28  | 1.1  | 16 | 22.1 | 89 | 69 | 181.86  | 88 | 47          | 46.49747      |
| 6       | 1075.33 | 100 | 1.2  | 75 | 50   | 78 | 84 | 1014    | 92 | 67          | 68.23855      |
| 7       | 115     | 86  | 6.6  | 57 | 76.6 | 78 | 84 | 112.5   | 37 | 69          | 69.81351      |
| 8       | 502     | 24  | 3.4  | 9  | 21.8 | 78 | 85 | 499.28  | 12 | 35          | 36.37946      |
| 9       | 7170.79 | 85  | 0.6  | 89 | 32   | 78 | 77 | 7170.43 | 89 | 62          | 60.94153      |
| 10      | 2049.5  | 63  | 0.4  | 38 | 50.7 | 89 | 91 | 2049.4  | 66 | 55          | 55.53001      |

### 5.3 Genetic Algorithm for finding WSRF of web services

The QWS dataset of 364 web services is used for this purpose. Since, the parameters response time (p1), availability (p2), throughput (p3), successability(p4), reliability(p5), compliance(p6), best practices(p7), latency(p8) and documentation(p9) vary in units, they are first normalized. It is assumed that the normalized WSRF is a linear function of remaining normalized QoS parameters. Therefore, we can formulate a function for normalized (WSRF)<sub>n</sub> considering all normalized web service parameters(pi)<sub>n</sub> values as:

$$WSRF_n = \sum_{i=1}^{n=9} w_i (p_i)_n \quad (1)$$

A continuous genetic algorithm is used to determine the coefficients or weights ( $w_i$ ). A population size of 80 is chosen. The probability of crossover and mutation are 0.85 and 0.001 respectively. Since some of the QoS parameters contribute negatively to WSRF, the range of nine coefficients is taken [- 1 1]. The objective function is formulated by taking 90% of the QWS data, i.e., 328 web services at random. Total absolute error for those 328 records is minimized. The fitness function taken is:

$$Fitness = \frac{1}{(1 + E)} \quad (2)$$

Where, E is the total absolute error of 328 web services.

The optimized weights obtained by genetic algorithm are tabulated below.

| w1      | w2       | w3       | w4       | w5       | w6       | w7       | w8       | w9      |
|---------|----------|----------|----------|----------|----------|----------|----------|---------|
| 0.17591 |          |          |          |          |          |          |          | 0.20415 |
| 1       | 0.162317 | 0.263999 | 0.115322 | 0.285133 | 0.073854 | -0.08905 | -0.19998 | 9       |

It is interesting to note that the negative signs with 7<sup>th</sup> and 8<sup>th</sup> weights corresponding to best practices and latency indicate their negative effect on WSRF measures. These optimized weights are used to compute WSRF for the remaining 36 web services. Finally, all the values are denormalized to obtain the actual WSRF values. The average error for per pattern is found to be 1.176.



A sample of calculated WSRF using the weights optimized by CGA (Continuous Genetic Algorithm) with actual WSRF values is recorded in Table 3.

**Table 3: WSRF as Computed by CGA**

| Sl. No. | P1      | P2  | P3   | P4 | P5   | P6  | P7 | P8      | P9 | Actual WSRF | Computed WSRF |
|---------|---------|-----|------|----|------|-----|----|---------|----|-------------|---------------|
| 1       | 166.11  | 100 | 10.3 | 86 | 75.7 | 78  | 80 | 158.89  | 90 | 81          | 81.13802      |
| 2       | 308     | 100 | 4.2  | 82 | 90.3 | 78  | 86 | 305.69  | 8  | 66          | 66.89934      |
| 3       | 105.4   | 100 | 16.5 | 80 | 89.4 | 78  | 72 | 105.4   | 91 | 87          | 88.984        |
| 4       | 213     | 83  | 20.4 | 50 | 77.2 | 78  | 91 | 210     | 42 | 72          | 73.40467      |
| 5       | 795.42  | 88  | 2.6  | 76 | 19.3 | 78  | 77 | 353.17  | 87 | 61          | 61.79505      |
| 6       | 7170.79 | 85  | 0.6  | 89 | 32   | 78  | 77 | 7170.43 | 89 | 62          | 63.79179      |
| 7       | 615     | 100 | 7.8  | 88 | 48.5 | 100 | 84 | 599.33  | 58 | 71          | 71.81647      |
| 8       | 141.66  | 26  | 9.2  | 10 | 87.6 | 78  | 86 | 135.33  | 12 | 53          | 53.96712      |
| 9       | 213.25  | 78  | 5.8  | 67 | 56.5 | 89  | 84 | 138.25  | 94 | 73          | 71.23432      |
| 10      | 456.75  | 71  | 4.3  | 62 | 30.3 | 89  | 84 | 432.5   | 4  | 51          | 49.79714      |

The above result shows that an empirical formula can be established by incorporating the weights, normalization and denormalization of each attribute, as an alternative means to compute the WSRF value as follows:

$$\text{WSRF} = 70 * [5.72329E-06(\text{RT}-45) + 0.001887407 (\text{AV}-14) + 0.008979558 (\text{TP}-0.1) + 0.0012535 (\text{SA}-7) + 0.003106024 (\text{REL}-5.9) + 0.002238 (\text{CP}-67) - 0.002406757 (\text{BP}-58) - 6.50352E-06(\text{LT}-31.5) + 0.002126656 (\text{DOC}-1)] + 30 \quad (3)$$

#### 5.4 Comparison of WSRF Calculation by ANN and CGA

In previous two sections, ANN and CGA were used for calculating WSRF with dataset 1. The results are in good agreement with actual values. Here, both the techniques are applied to dataset 2, with similar functionalities as dataset 1 for which WSRF value is not available in order to compare the two techniques. A few samples of the result are tabulated below.

**Table 4: Comparison of WSRF Calculated by ANN and CGA**

| RT      | AV | TP   | SA | REL | CP  | BP | LT     | DOC | WSRF(ANN) | WSRF(CGGA) | Difference |
|---------|----|------|----|-----|-----|----|--------|-----|-----------|------------|------------|
| 319.25  | 89 | 2.6  | 96 | 73  | 100 | 80 | 99.5   | 10  | 68.158455 | 66.76079   | 1.39767    |
| 198     | 96 | 14.7 | 99 | 67  | 78  | 72 | 50     | 10  | 71.086480 | 72.12520   | 1.03872    |
| 523.75  | 94 | 6.9  | 95 | 73  | 100 | 80 | 64.5   | 94  | 82.666750 | 82.63907   | 0.02768    |
| 320.48  | 86 | 1.2  | 86 | 53  | 89  | 66 | 125.18 | 10  | 58.665986 | 60.88265   | 2.21667    |
| 196.33  | 87 | 9.5  | 95 | 53  | 100 | 71 | 38.12  | 86  | 76.558963 | 79.20620   | 2.64724    |
| 474.91  | 91 | 5.8  | 97 | 60  | 78  | 74 | 162.87 | 86  | 73.921122 | 75.20932   | 1.28820    |
| 414     | 86 | 10.5 | 86 | 73  | 89  | 84 | 74.5   | 4   | 69.571611 | 67.21170   | 2.35991    |
| 514.33  | 47 | 3    | 47 | 73  | 78  | 75 | 123.5  | 9   | 52.836561 | 54.47793   | 1.64137    |
| 174.11  | 60 | 7.5  | 61 | 60  | 89  | 74 | 104.28 | 88  | 70.359578 | 70.95054   | 0.59096    |
| 205.33  | 90 | 3.5  | 97 | 60  | 100 | 74 | 48.33  | 28  | 69.180367 | 68.38793   | 0.79244    |
| 2836.25 | 79 | 2.4  | 79 | 73  | 67  | 84 | 78.75  | 7   | 56.793737 | 58.54982   | 1.75609    |



It is seen that the WSRF calculated by the two techniques are very close to each other. Besides accuracy, the proposed empirical formula inferred by CGA is very simple and efficient. Hence, it is much better than a trained neural network to compute WSRF values.

## 6 Conclusion

In this work, two different back propagation neural network models have been used for classification of web services with constant learning coefficient and adaptive momentum coefficient. In the first model using WSRF as one of the input parameter we achieved a classification accuracy of 100%. The second model which does not consider WSRF as an input parameter could yield only 89.9% accuracy as far as reported by others. However, in our proposed model the classification accuracy has been significantly improved to 97.22%.

Further, ANN is also used to predict WSRF values. The average error per pattern as predicted by the proposed neural network model is found to be 1.0447. As an alternate approach, an empirical formula for calculating the WSRF value is established by using CGA. The average error per pattern in this case is found to be 1.176. Finally, we compared the WSRF values obtained by applying the neural network model and the proposed formula using the dataset 2 with similar characteristics.

## ACKNOWLEDGEMENTS

We wish to thank Dr. E. Al-Masri and Dr. Q.H. Mahmoud for providing us with their valuable QWS dataset which has been used for this work.

## REFERENCES

- [1] Jinghai Rao and Xiaomeng Su., "A Survey of Automated Web Service Composition Methods", Lecture Notes in Computer Science, 2005, Volume 3387/2005, 43-54.
- [2] S.Susila and S.Vadivel, "Web service Selection through QoS agent Web service", IJWS 13-325; © 2013, pp.18-23.
- [3] Ramakanta Mohanty, V. Ravi and M.R. Patra, "Web-services classification using intelligent techniques", Elsevier, Expert Systems with Applications 37 (2010), pp. 5484–5490.
- [4] LI Yuan-jie and CAO Jian, "Web Service Classification Based on Automatic Semantic Annotation and Ensemble Learning", 2012, IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, 2274-2279.
- [5] Ramakanta Mohanty, V. Ravi and M. R. Patra, "Classification of Web Services Using Bayesian Network", Journal of Software Engineering and Applications, 2012, 5, 291-296.
- [6] L. Ai and M. Tang, "A hybrid genetic algorithm for the optimal constrained web service selection problem in web service composition", IEEE Congress on Evolutionary Computation, Barcelona, Spain, 2010, 8 pages.
- [7] E. Maximilien and M. Singh, "A framework and ontology for dynamic Web services selection" IEEE Internet Computing, vol. 8, no. 5, pp. 84–93, Sept.-Oct. 2004.
- [8] K. Verma, R. Akkiraju, R. Goodwin, P. Doshi, and J. Lee, "On accommodating inter service dependencies in web process flow composition," in Proc. AAAI Spring Symposium on SWS, 2004, pp. 37–43.

- [9] G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms", In Proc. of the 2005 conference on Genetic and evolutionary computation., New York, NY, USA: ACM, 2005, pp. 1069–1075.
- [10] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes", IEEE Transactions on Software Engineering, vol. 33, no. 6, pp. 369–384, June 2007.
- [11] Yu, T., Zhang, Y., and Lin, K.J., 2007, "Efficient algorithms for Web services selection with end-to-end QoS constraints", ACM Trans. Web 1, 1, Article 6 (May 2007), 26 pages.
- [12] Xu, Z.; Martin, P.; Powley, W. and Zulkernine, F.; "Reputation-Enhanced QoS-based Web Services Discovery", 2007, Proceedings of IEEE International Conference on Web Services. Salt Lake City, USA, pp. 249-256.
- [13] Al-Masri, E. and Mahmoud, Q., 2007, "Discovering the Best Web Service", Proceeding of the 16th international conference on World Wide Web. Banff, Canada, pp. 1257-1258.
- [14] Petrova-Antonova, Dessislava., "A qos-aware approach for web service selection based on probability evaluation", M CC SSI MCCSIS (2011): 43.
- [15] M. Kamalahasan, "Applications of Neural Networks for Ranking of Web Services using QoS Metrics", SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) – volume 1, Issue 1–Feb 2014, pp. 4-8.
- [16] Eyhab Al-Masri and Qusay H. Mahmoud, "Discovering the Best Web Service: A Neural Network-based Solution", Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October 2009, 4250-4255.
- [17] Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services", IEEE, 2007, pp.529-534