Grenze International Journal of Engineering and Technology, Jan Issue

Web Service Ranking and Classification using Intelligent Techniques

Dr. Ramakanta Mohanty
Methodist College of Engineering and Technology, king Koti Road, Abids, Hyderabad, TS
Email: ramakanta5a@gmail.com

***Abstract*—In the fastidious ever-growing world of businesses when the constant exchange of web services takes place, it is important for the buyer and seller to understand where their web service stands. A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems. Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer. The QWS dataset version II is collected from literature which is not having class level. The dataset is normalized using min-max method and employ stratified cross validation to make different fold systems. To make the different web services into different class levels, we employ two different clustering algorithm viz. K-means, and Fuzzy C-means to cluster different web services into different clusters. To test the different class level simulated by clustering algorithm, machine learning algorithm is employed i.e. Genetic Programming (GP), and Random Forest to test the efficacy of the model. From our experimental results, it is observed that C-Fuzzy means clustering provided the best clustering level compared to other techniques. The average accuracy of 99.23% provided by Genetic programming.**

***Index Terms*— Fuzzy C-means, K-means, Genetic Programming, Random Forest, Intelligent Techniques, Web Services.**

I. INTRODUCTION

The inter connectivity of thousands of different types of computers across the globe that are a component of various networks is known as the Internet. A web service is a defined technique for message transmission between client and server applications on the World Wide Web. A software module called a web service is designed to perform a certain set of tasks. In cloud computing, web services are searchable and dynamically typed across a network. The client that called the web service would be able to receive functionality from the web service [1].

A web service is a collection of open protocols and standards that enable the transfer of data across various software programs or computer systems. In a manner similar to inter-process communication on a single computer, web services may be used by software programs created in a range of programming languages and running on a variety of platforms to exchange data across computer networks like the Internet.

A web service is any piece of software, application, or cloud technology that connects to the internet and exchanges data messages, typically in the form of XML (Extensible Markup Language), using defined web

Grenze ID: 01.GIJET.9.1.740

protocols (HTTP or HTTPS). Web services have the advantage of allowing programs developed in different languages to connect with one another by exchanging data over a web service between clients and servers. A client invokes a web service by submitting an XML request, which the service responds with an XML response**.**

A. The functions of web services are

- It can be accessed through intranet networks or the internet due to the standardized XML message protocol.
- Independent of programming language or operating system.
- It is self-describing when using the XML standard.
- It is easy to find using a location-based strategy.

The basic Components of Web Service are XML and HTTP is the most fundamental web services platform. The following components are used by all typical web services:

A SOAP (Simple Object Access Protocol):

It is a message protocol that is independent of transport. Sending XML data in the form of SOAP Messages is the foundation of SOAP. Each communication includes an XML document that serves as an attachment. Only the XML document's structure, not its content, follows a pattern. The most advantageous aspect of Web services and SOAP is that all communication is done via HTTP, the industry-standard web protocol [2].

B. UDDI (Universal Description, Discovery, and Integration)

UDDI [2] is a standard for specifying, publishing and discovering an online service provider. It offers a specification that makes hosting data through web services easier. In order for a client application to find a WSDL file and learn about the various activities that a web service performs, UDDI offers a repository where WSDL files can be hosted. As a result, the UDDI, which functions as a database for all WSDL files, will be completely accessible to the client application. The necessary data for the online service will be stored in the UDDI register, just as a phone book would have a person's name, address, and contact information. So that a client application can determine its location [3].

C. WSDL (Web Services Description Language)

A web service cannot be used if it cannot be located. The web service's location should be known to the client making the web service request. In order to invoke the relevant web service, the client application must, second, comprehend what the web service does. This is done using the WSDL, or Web services description language. Another XML-based file that describes what the web service performs to the client application is the WSDL file. Using the WSDL document, the client application will be able to comprehend where the web service is located and how to access it [2].

In this paper, QWS version II [4, 5] dataset is collected from literature which is not having any class labels. To categorize the web service, it is propose to employ K-means and Fuzzy C-means clustering and followed by machine learning algorithm to the accuracy of categorization.

The remainder of the paper is organized as follows. Section II provides a concise discussion of the literature survey. A summary of machine learning approaches is given in Section III, and a thorough explanation of the results and discussions is given in Section IV. Finally, the paper's conclusion is presented in section V.

II LITERATURE SURVEY

To evaluate and categorize web services, a variety of machine learning approaches have frequently been used. As a WSDL component is crucial for classifying web services, Fokaefs et al. [6] used the VTracker tool to represent a WSDL file by calculating the minimum distance between two trees. From this, they discovered that the simulation results' accuracy varied significantly if the WSDL component was removed from the data. The tool's output is the proportion of XML models for two WSDL interfaces that have had items added, altered, or removed.

Romano et al. [7] use of a comparable technique known as WSDLDiff allows for the analysis of WSDL interface evolution without the need for manual inspection of the XML modifications. In order to extract highly dominating attributes using feature selection, Aversano et al. [8] compare the associations between the various attributes and also extract the relationship among attributes by employing machine learning. To represent various UML diagrams, Xing et al. [9] chose UMLDiff and discovered the function of each attribute in the classification

of web services. In order to categorize and forecast whether web services will be suitable for a given purpose, Zarras et al. [10] concentrated on Amazon Web services' progress in the real world.
A number of SOA anti patterns were discovered by Kral et al. [11] and several anti-patterns in SOA architecture were also addressed. Recently, Moha et al. [12] used the SODA for SCA systems rule-based approach (Service Component Architecture). Later, this technique was expanded for Web service ant patterns in SODA by Palma et al. [13]. Using a set of WSDL metrics, W is used to specify/identify the primary symptoms that characterize an anti-pattern. WSDL is a declarative rule specification based on a domain specific language (DSL). Based on eight bad practises in the development of WSDL for Web services, Rodriguez et al. [13] and Mateos et al. [14] presented a set of principles for service providers to follow when producing WSDLs. Recently, Ouni et al. [15] proposed a search-based approach based on standard GP to find regularities, from examples of Web service anti patterns, to be translated into detection rules. Most recently, Mohanty et al.[2, 16,17] employed neural networks, decision Tree and support Vector Machine to classify the To uncover regularities from samples of Web service anti patterns to be converted into detection rules, Ouni et al. [14] recently presented a search-based technique based on conventional GP. Most recently, Mohanty et al. [2, 16, 17] suggested data envelopment analysis and FMADMA to categorize various web services based on the quality of web services. They also used neural networks, decision trees, and support vector machines to identify the quality of web services. Quality of web services and also proposed data envelopment analysis and FMADMA to classify different web services based on the quality of web services.

III. METHODOLOGY

In this paper, two different types of clustering method used to classify the web services into different class labels based on their quality attributes.

(i) K-means algorithm
(ii) Fuzzy C-means Clustering Algorithm

A. K-means Algorithm

K-Means Clustering [19, 20, 21] is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here, K specifies how many pre-defined clusters must be produced as part of the process; for example, if K=2, there will be two clusters, if K=3, there will be three clusters, and so on. It gives us the ability to divide the data into various groups and provides a practical method for automatically identifying the groups in the unlabeled dataset without the need for any training. Each cluster has a centroid assigned to it because the algorithm is centroid-based. This algorithm's primary goal is to reduce the total distances between each data point and its corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.
The two major functions of the k-means clustering algorithm are:

- Through an iterative approach, chooses the best value for K Centre points or centroids.
- Each data point is assigned to the nearest k-center. A cluster is formed by the data points that are close to a specific k-center.

Hence each cluster has data points with some commonalities, and it is away from other clusters.

B. Fuzzy C means Clustering

It is an approach for unsupervised learning that enables us to create a fuzzy division out of data. The approach is dependent on a parameter m that represents how fuzzy the result is. The classes will become muddled for high values of *m*, and all items will tend to belong to all clusters. The parameter *m* affects how the optimization problem is solved. In other words, different *m* choices will likely result in distinct partitions. An illustrating the impact of the *m* selection determined by the fuzzy c-means is provided below.
Instead of belonging to only one cluster as is the case with classic k-means, each point has a probability of being a part of each cluster with fuzzy clustering. In Fuzzy-C Means clustering, each point has a weighting associated with a specific cluster, as a result, a point is not so much "in a cluster" as it is "affiliated" with the cluster, with the degree of association being decided by the inverse distance to the cluster's centre. It is actually performing more work than K means, fuzzy-C means will typically run more slowly. With each cluster, each point is reviewed, and each evaluation involves more processes. K-Means only needs to calculate distance, whereas fuzzy c-Means needs to perform a full inverse-distance weighting.
The process flow of fuzzy c-means is enumerated below:

1. Assume that there are k fixed clusters.
2. **Initialization:** Calculate the likelihood that each data point x_i belongs to a specific cluster k using a random initialization of the k-means μ_k and the formula P (point x_i has label $k|x_i, k$).
3. **Iteration**: Recalculate the cluster centroid as the weighted centroid given the membership probability of all the data points. x_i

$$\mu_k(n+1) = \frac{\sum_{x_i \epsilon k} x_i * P(\mu_k|x_i)^b}{\sum_{x_i \epsilon k} P(\mu_k|x_i)^b} \qquad (1)$$

4. **Termination**: Continue iterating until convergence or up to the amount of iterations that the user specifies (the iteration may be trapped at some local maxima or minima).

C. Genetic Programming

GP is an extension of GA that does away with the requirement that the chromosome representing the individual be a binary string of a specific length. In GP, the chromosome functions as a kind of programs that is performed to provide the desired consequences. A binary tree with operators and operands is one of the most basic types of programs, and it works for this application. As a result, each answer can be evaluated as an algebraic expression. According to Koza [23] "GP can be used to solve a huge number of seemingly dissimilar issues from many different domains," which is a startling and counterintuitive result.
The executional steps of Genetic Programming are as follows:

1. Generating an initial population (generation 0) of distinct computer programs at random from the available terminals and functions.
2. The next sub-steps (referred to as a generation) should be applied to the population iteratively until the termination criterion is met.

- Run each program in the population and determine (explicitly or implicitly) its fitness using the fitness measure for the problem.
- With reselection permitted, choose one or more individual program(s) from the population with a probability depending on fitness to take part in the genetic processes in (c).
- By conducting the following genetic operations with the following probability, create new individual program for the population:
- Copy the individual program you have chosen to the new population. - Crossover: By recombining two selected program ' randomly chosen portions, produce new offspring program(s) for the new population. - Mutation: By randomly changing a randomly selected portion of one selected program, one new offspring program will be produced for the new population. - Operations that change the architecture.

3. The best program in the population created throughout the run (the best-so-far person) is collected and declared as the run's outcome once the termination requirement has been met. If the test runs well, the outcome might be a solution or a close approximation to the issue.

D. Random Forest

Random Forests are an example of Ensemble methods that are used for the purpose of classification. If numerous decision trees are used with a random selection of attributes, the result is a random forest, which improves the performance of a great example. At each split, a fresh random sample from the features is selected for each tree [24, 25, 26]. For instance, the dataset can have a highly potent feature. Most trees would use that feature as the top split when using bagged trees. A group of trees with a lot of correlation would come from this. Such highly connected numbers cannot be averaged, hence the gap will remain. Random forests attempt to build trees by randomly deleting features from each split, which is advantageous [26].

Algorithm
Decision Tree (T, P, y)

- *Initialize a new Decision Tree(DT) with a single root node*
- *If one of the stopping criteria is met, then, label the root node in DT as a leaf with most prominent value of 'y' in DT.*
- *ELSE, search for a distinct function (f(p)) at the input attributes value such that splitting DT according to f(p)' outcomes(v_1, v_2, v_3,……v_n) gains the best splitting metric with highest accuracy.*
- *If best splitting metric>'threshold' THEN label't' with f(p).*

- *FOR each outcome v_i of f(p)*
  - *Set ST1=DecisionTree (BS(f(p))= (yDT, P,y)*

Where BS=best splitting metric.

- *Connect the root of t'_{DT} to sub tree 'e' with an edge that is labelled as v_i*
- *END FOR*
- *ELSE mark the root node in T as a leaf with the most common value of 'y' in S as label*
- *END IF*
- *END IF*
- *RETURN DT*

T= Training data, p= prediction, y= class label

IV. Results and Discussions

In this paper, Quality Web Services (QWS) dataset of version II is collected from literature. The dataset is having 10 attributes viz. Response time, Availability, Throughput, Successability, Reliability, Compliances, Best Practices, Latency, and Documentation and having no class labels [18]. The data set is having two thousand five hundred seven samples of records. Firstly, the dataset is normalized by using min- max method so that all data point within the range of 0 and1. As dataset is to be categorized into different labels as to find out the web services rank demanded by different software organizations claimed to be a particular category, two different clustering techniques are employed viz. K-means algorithm and Fuzzy C-means algorithm to the web services to categories into four labels. By employing Fuzzy C-means algorithm, it is observed that there are 766 samples of web services categorized into cluster level '1', 706 samples of web services in category '2' followed by 549 in category '3' and 485 web services comes under the category of class '4'. But in case of K-means algorithm, it is also observed that there are 369 samples of web services categorized into '1', 466 samples of dataset categorized into category '2', followed by 548 samples of web services in category '3' and finally 1123 web services categorized into category '4'.

As it is found that the two algorithm categorized the web services into different clusters, it is required to test their accuracy level by employing machine learning techniques. Accordingly, two machine learning techniques are employed i.e. Random Forest and Genetic Programming to find out the accuracy of different clusters based on their quality attributes.

TABLE I. WEB SERVICE RANKING BY FUZZY C-MEANS ALGORITHM AND ITS ACCURACY BY MACHINE LEARNING TECHNIQUES

Sl. No	**Techniques**	**Fold-1 (%)**	**Fold-2(%)**	**Fold-3 (%)**	**Fold-4 (%)**	**Fold-5 (%)**	**Fold-6 (%)**	**Fold-7 (%)**	**Fold-8(%)**	**Fold-9 (%)**	**Fold-10 (%)**	**Average (%)**
1	GP	99.9	99	98.9	99	98.9	99.9	98	99.9	98.9	98.9	99.23
2	Random Forest	97	96.8	97	97	98	97.7	97.8	97	97	98	97.33

It is learned from Table 1 that folds 1 and 6 had 99.9% accuracy for clustering. The accuracy of the folds 2, 4, and 8 is 99%, while the accuracy of the folds 3, 5, 9, and 10 is 98.9%, on average of 99.23% for results derived from GP simulation. The accuracy of folds 1, 3, 4, 8 and 9 is observed to be 97% using Random Forest. The accuracy of folds 5 and 6 is 98%. The accuracy for fold 2 is 96.8 %, fold 6 is 97.7%, fold 7 is 97.8%, and the average accuracy is 97.33%.

TABLE II. WEB SERVICE RANKING BY K-MEANS ALGORITHM AND ITS ACCURACY BY MACHINE LEARNING TECHNIQUES

Sl.No	Techniques	Fold-1(%)	Fold-2(%)	Fold-3 (%)	Fold-4 (%)	Fold-5 (%)	Fold-6 (%)	Fold-7 (%)	Fold-8(%)	Fold-9 (%)	Fold-10 (%)	Average (%)
1	GP	99.9	99	98.9	99	98	99	99.9	97	99	98	99
2	Random Forest	97.3	94.8	95.6	97	95.4	97.7	96.8	97	96.5	95.9	97.33

Web services are ranked using K-means clustering in Table 2, the GP and Random Forest are used to test the precision of the rankings. From the simulated study, it can be seen that folds 1 and 7 have accuracy of 99.9%, while folds 2, 4, 6, and 9 have accuracy of 99%. The maximum accuracy was determined to be 99.9% for folds 1

and 7, which is pretty, incredibly good. Similar to the random forest method, the average accuracy of web services ranking were determined to be 97.33%, which is also quite good.
Oveally, it is observed that both techniques are predicted high degree of accuracy as far as web service is concerned.

V CONCLUSION

With the ubiquitous use of web services, it is crucial to track down and evaluate their quality. The machine learning models (techniques) that have been employed to categorize these web services into four main categories based on quality criteria. In this paper, fuzzy C-means and K-means algorithms were used to rank various web services into four clusters. The QWS dataset has been subjected to analysis using a number of supervised models, including A, B, C, and D. Later, the accuracy label of the clustering approaches used by the K-means and Fuzzy C-means algorithms was checked the accuracies using Random Forest and Genetic Programming. In this paper, the scikit learn library has been used for simulation. While Random Forest approach has done quite well, genetic programming has performed exceptionally well.
To increase the mannequin's accuracy, feature scaling and hyper parameter tuning have been used. The phases of machine learning modelling, from data splitting into train and test sets (used exclusively for hyper parameter tuning), through computation of average accuracies for all 10 folds presented. Finally, we draw the conclusion that, given the collection of quality attributes, this study might be utilized to classify a new web service into one of the four major classifications.

REFERENCES

[1] J.-M. Jézéquel, B. Baudry, Y.-G. Guéhéneuc, B. J. Conseil, M. Nayrolles, [11]F. Palma and N. Moha, "Specification and Detection of SOA Antipatterns," in Service-Oriented Computing, Shanghai, Springer Berlin Heidelberg, 2012, pp. 1-16.

[2] **Ramakanta Mohanty**, V. Ravi and M. R.Patra, (2010) **'**Web Services Classification using intelligent Techniques'. Elsevier, Expert Systems with Applications 37, PP. 5484-5490.

[3] Y.-G. Guéhéneuc, G. Tremblay, N. Moha and F. Palma, "Specification and Detection of SOA Anti patterns in Web Services," in Software Architecture, Vienna, Springer International Publishing, 2014, pp. 58-73.

[4] AL-Masri, E., & Mahmood, Q." H.QoS based discovery and ranking of web services", In IEEE 16th international conference on computer communications and networks (ICCCN), pp. 529–534, 2007.

[5] AL- Masri, E., & Q. H. Mahmood, " Investigating web services on the World Wide Web", In 17th International conferences on World Wide Web, PP. pp. 795–804, 2008, Beijing.

[6] M. Fokaefs, R. Mikhaiel, N. Tsantalis, E. Stroulia and A. Lau, "An Empirical Study on Web Service Evolution," in IEEE International Conference on Web Services, pp. 49-56, 2011.

[7] D. Romano and M. Pinzger, "Analyzing the Evolution of Web Services Using Fine-Grained Changes," in IEEE 19th International Conference on Web Services (ICWS),pp. 392-399, Honolulu, 2012

[8] L. Aversano, M. Bruno, M. Di Penta, A. Falanga and R. Scognamiglio, "Visualizing the evolution of Web services using formal concept analysis," in IWPSE '05 Proceedings of the Eighth International Workshop on Principles of Software Evolution, pp. 57-60, Lisbon, Portugal, 2005.

[9] Z. Xing and E. Stroulia, "UMLDiff: an algorithm for object oriented design differencing," in ASE '05 Proceedings of the 20th IEEE/ACM international Conference on Automated software Engineering, pp. 54-65, Long Beach, California, 2005.

[10] L. Dinos, P. Vassiliadis and A. V. Zarras, "Keep Calm and Wait for the Spike! Insights on the Evolution of Amazon Services," in Advanced Information Systems Engineering, vol. 9694, Ljubljana,Springer International Publishing, 2016, pp. 444-458.

[11] J. Král and M. Žemlicka, "Popular SOA Antipatterns," Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, pp. 271-276, Athens, Greece, 2009.

[12] J.-M. Jézéquel, B. Baudry, Y.-G. Guéhéneuc, B. J. Conseil, M. Nayrolles, [11]F. Palma and N. Moha, "Specification and Detection of SOA Antipatterns," in Service-Oriented Computing, Shanghai, Springer Berlin Heidelberg, 2012, pp. 1-16.

[13] J. M. Rodriguez, M. Crasso, A. Zunino and M. Campo, "Automatically Detecting Opportunities for Web Service Descriptions Improvement," in Software Services for eWorld, pp.139-150, Buenos Aires, Argentina, 2010.

[14] C. Mateos, J. M. Rodriguez and A. Zunino, "A tool to improve code-first Web services discoverability through text mining techniques," Software: Practice and Experience, vol. 45, no. 7, p. 925–948, 2014.

[15] A. Ouni, R. G. Kula, M. Kessentini and K. Inoue, "Web Service Antipatterns Detection Using Genetic Programming," in GECCO '15 Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation,PP. 1351-1358, Madrid, Spain, 2015.

[16] V. Ravi, **Ramakanta Mohanty**, M. R. Patra, (2009) 'Application of Data Envelopment Analysis to Rank Web Services', The 3rd International conferences on Global Interdependence and Decision Sciences (ICGIDS '09), pp. 287 – 296, McMillan, India. ASCI, Hyderabad.
[17] **Ramakanta Mohanty**, V. Ravi, M R Patra, "Application of Fuzzy Multi Attribute Decision making Analysis to Rank Web Services", The 6th **IEEE** International Conference on next Generation web services Practice (NWeSP, 2010), pp.52-57, 2010, Gwalior, India.
[18] R. Agrawal, A. Gupta, Y. Prabhu, M. Varma, "Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages", WWW '13 Proceedings of the 22nd international conference on World Wide Web, pp. 13 – 24, 2013.
[19] P. B Callahan and S. R. Kosaraju, "A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields", In Proc. 24th Ann. ACM Sympos. Theory Comput, pp. 546–556, 1992.
[20] K. L. Clarkson, K. L. "Fast algorithms for the all nearest neighbor's problem", In Proc. 24th Ann. IEEE Symposia on the Found. Comput. Sci., pp. 226–232, 1983
[21] J. G. Cleary, "Analysis of an algorithm for finding nearest neighbors in Euclidean space", ACM Transactions on Mathematical Software 5, 2, 183–192, 1979.
[22] Scikit-learn:Machine Learning in Python, Pedregosa *et al.*, JMLR12, pp. 2825-2830, 2011
[23] J. R.Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection". Cambridge, MA: The MIT Press, 1992.
[24] Andrew Colin, "Building Decision Trees with the ID3 Algorithm", Dr. Dobbs Journal, 1996
[25] R. Quinlan, "Induction of decision trees," Machine Learning, vol.1, No.1, pp. 81-106, 1986
[26] L. Breiman, "Random forests. Machine Learning", 45(1), 5- 32, 2001.