DOI: 10.21917/ijsc.2019.0269

IMPLEMENTATION OF QOS BASED INNOVATIVE FRAMEWORK FOR WEB SERVICES SELECTION AND COMPOSITION

Khozema Shabbar[1], Tarun Shrimali[2] and Ajay Chaudhary[3]
[1]*Department of Computer Science and Engineering, Career Point University, India*
[2]*Department of Computer Science and Information Technology, Janardan Rai Nagar Rajasthan Vidyapeeth, India*
[3]*Department of Psychology, Government Meera Girls College, India*

Abstract

Web Services in ample amount lying over the net are found similar in functionality varying in QoS (Quality of Services). Various methods and techniques have been proposed by researchers in order to get the most favoured and best-suited candidate services to be able to become the part of the final outcome which is called composite web service. Quality of Service has emerged as an important tool playing decisive role while selecting best candidate web Services as well as Composite Web Services. Undoubtedly, the technique of web services composition is the key property of service-orientation helps create new as well as advanced level services reusing the existing ones. In a distributed environment, services with no quality guarantees, have adverse impact on the final composition outcome. In their efforts to select the most favoured candidate services to be used for the composition, scholars while proposing new composition techniques relied on either Users' provided feedbacks or Providers' provided information. None of them took into account both of the information altogether. In our novel approach, we took into account both of the information (Users' provided feedbacks and Providers' provided information) altogether. We have designed an innovative framework incorporating an entity called Information Verification Engine whose job is to take into account both of the information, filtering out malicious feedback ratings and cross verifying both of the information in order to find the most suitable candidate web services. We have implemented this newly added Engine taking into account total of 450 Web Services and applying Statistical Computations using SPSS version 21.0. This has helped achieve a set of 279 web services, purely rated by Unbiased Users. Then We Cross Verified both of the ratings which further reduced no. of Web Services to 83. Experimental results confirmed the efficiency and reliability of our proposed framework. Detailed methodology and its implementation has been presented in the later part of this paper.

Keywords:

Web Services Composition, SPSS, Mean, Mode, Median, SD, Skewness

1. INTRODUCTION

Carrying out service composition by clubbing together either existing composite web services or atomic web services, satisfying both functional and non-functional requirements of the users and achieving overall QoS optimal value seemed to be a challenging and difficult task. This task of Composition gets more complicated when users' requirements are changeable and real-time. Since, web service composition is full of uncertainty, hence auto-dynamic composition of web services appears to be the most effective and the only solution. Although, auto-dynamic composition of web services is taken as a complicated technique since it is executed at run time based on the users' requirements. But the key advantage of this technique is dynamic invocation and discovery of web services upon users' request, which is vital and novel. Further, this technique (auto-dynamic composition of web services) provides an effective solution ensuring service quality and reliability. Most of the auto-dynamic compositions are based on workflow and/or Artificial Intelligence techniques. Different researchers have adopted different methods. Examples include methods based on multi-agents, based on work flow, based on Multi-Dimension QoS, based on Improved Genetic Algorithm (IGA), based on Fuzzy Logic, based on Ant Colony Optimization, based on Intelligent Algorithm, based on Graph, based on Decentralized Dataflow, based on Semantic Web etc.

1.1 RATING BASED TECHNIQUES

While publishing, Providers usually publish QoS related information of the web services including ratings of the web services. Users' upon using web services furnish feedbacks in terms of ratings based on their experience. In other words, Feedback rating is the evaluation by the users about the services invoked by them. In rating based methods, researchers in their work considered either QoS related information furnished by Service Providers in terms of published information or by Service Users in terms of the feedbacks for the said web services. In either of the case, it causes wrong selection of candidate web services adversely affecting the final outcome i.e. composite service. This happens because providers as well as users may not always be genuine and honest. In others words, they may be dishonest or biased. Henceforth, we have considered both of the information i.e. information furnished by the providers and feedbacks provided by the users. We have proposed a composition framework incorporating a novel engine named IVE that cross verifies both of the information helps achieve a set of best candidate services for the composition. We have successfully implemented IVE and achieved desired goal.

The paper is organized in the following manner. Related work is illustrated in section 2. Section 3 is all about preliminaries. Section 4 details about the proposed framework. Section 5 discusses composition execution method. Section 6 presents implementation of IVE. Section 7 narrates conclusion and future work.

2. RELATED WORK

Some of the works related to web services dynamic composition are summarized here under.

Jatoth et al [1] proposed a distinctive MR (Map Reduce) based algorithm using Guided Mutation for an effective composition of Big Services enhancing performance by reducing execution time. Additionally, they included a MapReduce skyline operator that plays vital role in QoS based service pre-selection process

improving the quality of results and the process of convergence. Authors conducted different tests including: Wilcoxon signed rank test and T-test and found outstanding result of the proposed method.

Zou et al. [2] developed a novel technique for auto-dynamic composition of Web services taking into account state-of-the-art AI (Artificial Intelligence) planners. This technique converts a Web service repository into a planning domain in PDDL just once regenerating it only when the Web service repository changes. This technique is novel in the sense that it avoids revisiting Web service repository each time a new request is submitted, thus improving the scalability of composition method effectively reducing the response time. Authors on conducting extensive experiments on the aforementioned composition method found outperforming results.

Lin et al. [3] proposed a QoS-based service selection (RQSS) algorithm carrying out composite web application development to get higher system availability and reliability. Proposed algorithm not only helped finding web services fulfilling users' functional as well as non-functional QoS requirements but also recommended solutions that could satisfactorily meet non-relaxable QoS requirements by relaxing the relaxable QoS requirements. Also, authors designed a generic framework evaluating system performance for service selection algorithms. Experimental outcome endorses better performance of RQSS. It found much lower failure rate of finding a feasible solution in RQSS than to other approaches like - RWSCS_KP and WFlow. RQSS could be able to perform well because of two things – 1. Low computation complexity of itself and 2. Idea of relaxing of QoS constraints.

Parejo et al. [4] presented a meta-heuristic algorithm 'QoS-Gasp' to carry out QoS-based dynamic web service composition. Proposed algorithm 'QoS-Gasp' is a combination of two meta-heuristic techniques/ approaches namely GRASP and Path Relinking. Authors evaluated their approach comparing it with other meta-heuristic algorithms. Experimental results showed that QoS-Gasp outperformed previous meta-heuristic proposals in rebinding scenarios, implying cost savings, increasing availability and reducing execution times.

Lee et al. [5] introduced an efficient and scalable composition search system for web services named PSR i.e. Pre-computing Solutions for web service composition in a RDBMS. Authors used relational database in place of in-memory algorithms. PSR system being a different one, favours semantic matching of web services composition and uses joins and indices for pre-computation of web services composition. This pre-computing algorithm works on a directed, weighted graph formed based on ontology information and web services. Authors used composed web services as intermediate vertices of paths between web services in the composition graph. The PSR system stores the graph into tables and joins these tables in order to compute answers for semantic web services composition search in advance. Proposed PSR system proved to be scalable one while handling greater number of user queries and web services yielding lower execution time in processing user queries.

Wang et al. [6] introduced a feasibility-enhanced approach for QoS-aware Web service composition in cloud environment proposing service selection algorithm and service evaluation method taking into account SLA satisfaction. SLA violation problem is handled using SLA constraint. Authors designed service selection algorithm based on the service evaluation method, simulated annealing and genetic algorithm that attains better feasibility not affecting solution's optimality. Experimental outcomes confirmed performance of the proposed approach with regard to the feasibility and the optimality of the composite services.

Yilmaz et al. [7] proposed a method based on two improved genetic algorithms to get optimized the overall QoS value of the given execution plan. Authors introduced smart mutation operator in such a way, that doesn't check the fitness value of mutated individuals in the former generations. It rather accepts mutations in the later generations in case if it finds higher fitness values for them. Authors first introduced an algorithm named '*GA_SA',* used 'Simulated Annealing' as the mutation operator, made the population have higher fitness values than the original GA. Authors introduced second algorithm named '*GA_HS',* used 'Harmony Search' as the smart mutation operator, was found better than ACO enabled GA and pure GA. Compare to pure GA, *GA_HS* convergence rate was far-far better. Also, fittest individual of the said algorithms were found to be quite better than the best of pure GA. Contrary to ACO enabled GA that operates on serial execution plans only, proposed method (based on GA_SA, GA_HS) proved to be suitable for all type of compositions.

Brahmi et al. [8] proposed a novel and efficient Multi-Agents based web service composition method to solve the major limitations with regard to dynamicity, scalability and accuracy. Agents cooperate to each other in order to find the feasible and optimal composition. Proposed method works in two steps - (a) Self-organization of agents (named as social network agent) into dependency graph and (b) Distributed computation of the optimal web services composition using a cooperative protocol amongst agents. Experimental results confirmed scalability, accuracy and dynamicity of the proposed method.

Liu et al. [9] presented a QoS aware extended model of Web Services composition based on providers' benefit, describing QoS attributes of Web Services in expanded UDDI. Authors introduced a Service Proxy role to find, bind and invoke users' requested web services compositions and presented a Service Request Queue Scheduling algorithm maximizing services providers' benefit. Experimental outcomes endorsed performance of the model in successfully meeting user's QoS requirements, enhancing providers' benefit and assuring availability of Web Services composition.

Yu et al. [10] presented a genetic programming (GP) approach for QoS-based web services composition. Primary objective is to achieve required outputs against supplied inputs plus to ensure optimal QoS value of the composite service. Additionally, an adaptive form of genetic programming (AGP) is applied to avoid premature and low rate of convergence. Authors evaluated their proposed approach conducting a no. of experiments and found it better than the existing ones finding a valid solution with lower in search time.

Ayoub [13] proposed a technique to successfully select services for QoS based composition employing the concept of Skyline approach and agent oriented system that reduces the number of candidate services to be generated and combined. They used the paradigm of system agents to increase the speed and

performance of algorithm taking advantage of its collective and intelligence behaviours. Experimental outcomes confirmed the usefulness of the proposed technique especially with regard to big number of candidate combinations as well as significantly reducing execution time.

Sadouki [14] presented a meta-heuristic bio-inspired algorithm based on Elephant Herding Optimization (EHO) algorithm in order to address the QoS based service composition. On comparing with PSO, EHO was found to be more efficient as far as large dataset is concerned. Experimental results showed excellent performance of EHO over existing algorithms with regard to Scalability, Convergence Speed and fitness value.

3. PRELIMINARIES

3.1 MEAN, MODE, MEDIAN AND STANDARD DEVIATION

- **Mean** is calculated by adding all the existing elements of a data set and dividing this sum by total number of data elements. The resultant value what we get is called the Mean.

$$\bar{x} = \sum \frac{fx}{n} \tag{1}$$

where $\bar{x}$ = Mean, f = Frequency, x = Data Element, n = Total Frequency

- **Median** is computed by first arranging all the elements of a data set in an ascending order. Then, data element found exactly in the middle of the series is termed as Median. In case, if the total number of elements in a data set is found to be even then average computed of the two middle values is termed as Median.
- **Mode** is computed by finding the element from the data set that is occurring most frequently.
- **Standard Deviation** is a statistical measure to compute dispersion of a dataset from its mean value. It is calculated as the square root of variance by determining the variation between each data point relative to the mean.

$$\sigma = \sqrt{\frac{\sum f\left(x-\bar{x}\right)^2}{n}} \tag{2}$$

3.2 SKEWNESS AND KARL PEARSON'S FORMULA

Skewness is a method used to measure the asymmetry lying in the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative.

In case of skewed frequency distribution, values of the mean and the median lie at different points in the distribution and the centre of gravity is moved to one side or the other i.e. to the left or right.

Skewness Measures tell us about the extent and the direction of the skewness. In symmetrical distribution the mean, mode and median are found to be identical whereas in an asymmetrical distribution mean moves away from the mode.

Karl Pearson proposed two methods to measure skewness in a sample -

- Pearson's Coefficient of Skewness #1 used the mode and is calculated using formula:

$$Sk_1 = (Mean - Mode)/\sigma \tag{3}$$

where σ = standard deviation for the sample

- Pearson's Coefficient of Skewness #2 used the median and is calculated using formula:

$$Sk_2 = 3(Mean - Mode)/\sigma \tag{4}$$

SPSS stands for Statistical Package for Social Sciences, is a software package widely used for statistical analysis in social science. It is also used for market and health research, organizational surveys, educational research and data mining etc. It allows ordinary researchers to do their own statistical analysis.

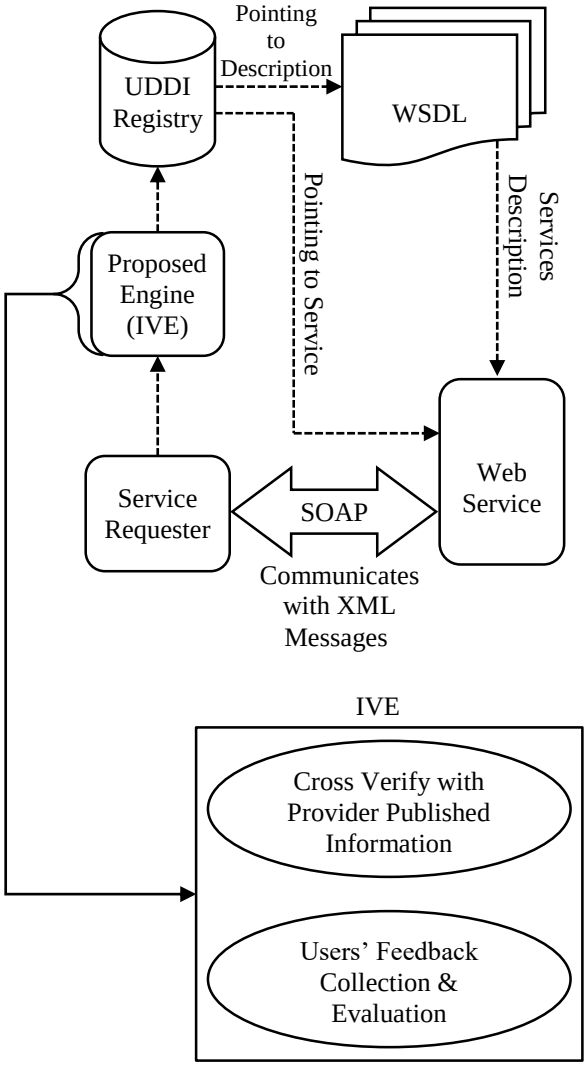

Fig.1. Enhanced Web Service Framework

3.3 ENHANCED WEB SERVICE FRAMEWORK

Here we are presenting an Enhanced Web Service Framework [11] by placing a new Engine named Information Verification Engine (IVE) between Service Requester and UDDI Registry. This Proposed Information Verification Engine (IVE) comprises

two sections: 1) Users' feedback collection and evaluation section and 2) Verification section.

4. PROPOSED COMPOSITION FRAMEWORK

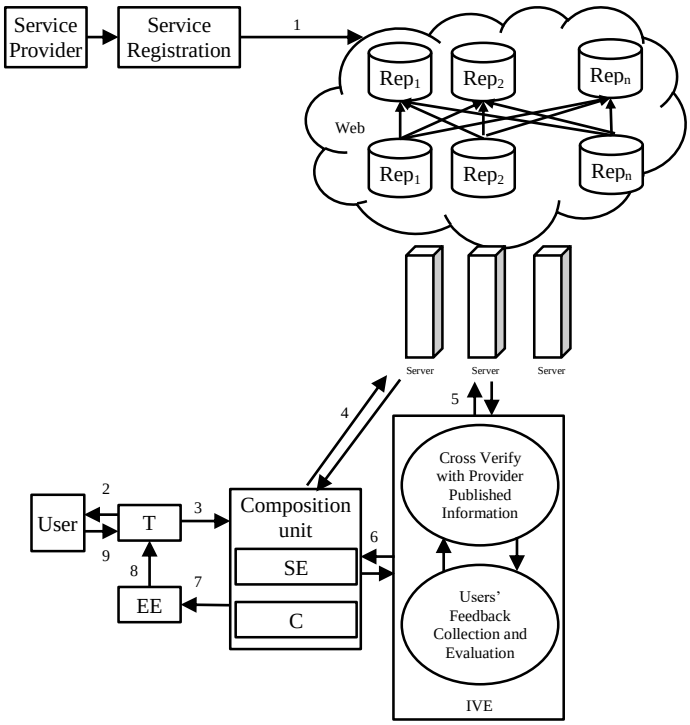

Fig.2. Proposed Web Services Composition Framework

Various components of the proposed Composition framework [11] are depicted in Fig.3 and are discussed here under -

- **Web Services Registration:** The method of specifying web services to the system is termed as service registration. New services are enrolled in the service-registry through this process. Service providers maintain various service registries and are synchronized regularly in a course of time.
- **Service Requester (SR):** Requesters through service request module send request for a particular service.
- **Translator (T):** Translator's job is to translate user's request from an external form to a system adaptable form and to translate response conversely.
- **Web Server:** Registries are hosted on server over the web.
- **Information Verification Engine (IVE):** IVE is an engine that cross verifies selected web services (meeting users' stated criteria) with the help of Users' feedback and Providers' published information before returning them for the final composition.
- **Web:** Here in the framework, Web stands for WWW network where service providers do register their web services in UDDI registries.
- **Composer (C):** The job of the composer is to perform composition of the selected candidate services. The resultant web service is then called as composite web service.
- **Service Engine (SE):** The job of service engine is to check for the web services from the web meeting users' request and returns those to the composer.
- **Service Registry:** Service providers' published web services are registered in the service registries and are used to make request for user's preferred web services. References of the web services being hosted on the service repositories are maintained by each registry.

5. COMPOSITION EXECUTION METHOD

5.1 PROCEDURE

Steps of the procedure of the proposed framework are discussed here under:

- Very first registration of Web services take place in various registries.
- User makes request for a particular service.
- Translator translates users' request into a system adaptable form.
- The request reaches to the Composition Unit where Service Engine checks for users' desired service from multiple registries lying on the web returning results to the IVE.
- IVE verifies these selected candidate web services in two stages. In the first stage, it sends the selected web services to seek Users' feedback, whereas in second stage it cross verifies Users' feedback with Provides' published information and vice versa. After verification, verified services are sent to composer.
- Composer performs composition of these verified services and sends result to Execution Engine (EE) which in turn executes these web services. Final outcomes are returned through translator back to the user.

5.2 PSEUDOCODE

Pseudocode of the procedure involved is outlined here:

Algorithmic Steps of Web Services Composition:

Input: Users' Request for specified service;

Output: Composite Web Service;

Step 1: Web services are registered in the web registries;

Step 2: Translator does translate users' request;

Step 3: User makes request for desired service;

Step 4: Request arrives at Service Engine;

Step 4.1: Service engine searches web UDDI registries for service;

Step 4.2: Selects matching services;

Step 4.3: IVE verifies these selected services based on users' feedback and providers' published information;

Step 4.4: Composer does composition of verified services;

Step 4.5: Composer returns outcome to Service Engine;

Step 4.6: Service engine in turn returns outcome to execution engine;

Step 4.7: Execution engine executes these services;

Step 4.8: Final outcomes are returned to the user through translator.

6. IMPLEMENTATION OF INFORMATION VERIFICATION ENGINE

Proposed Information Verification Engine (IVE) contains two modules - 1. Users' Feedback module and 2. Cross Verification Module (with Providers' Published Information). Users' Feedback Module has been implemented with the help of Statistical Methods. These include - computations of Mean, Mode, Median, Standard Deviation and Skewness. These statistical computations helped detect and filter out malicious feedback ratings. Resultant filtered users' feedbacks are finally cross verified with the Providers' published information in order to select the best candidate services for the composition.

6.1 IMPLEMENTATION FLOWCHART OF INFORMATION VERIFICATION ENGINE

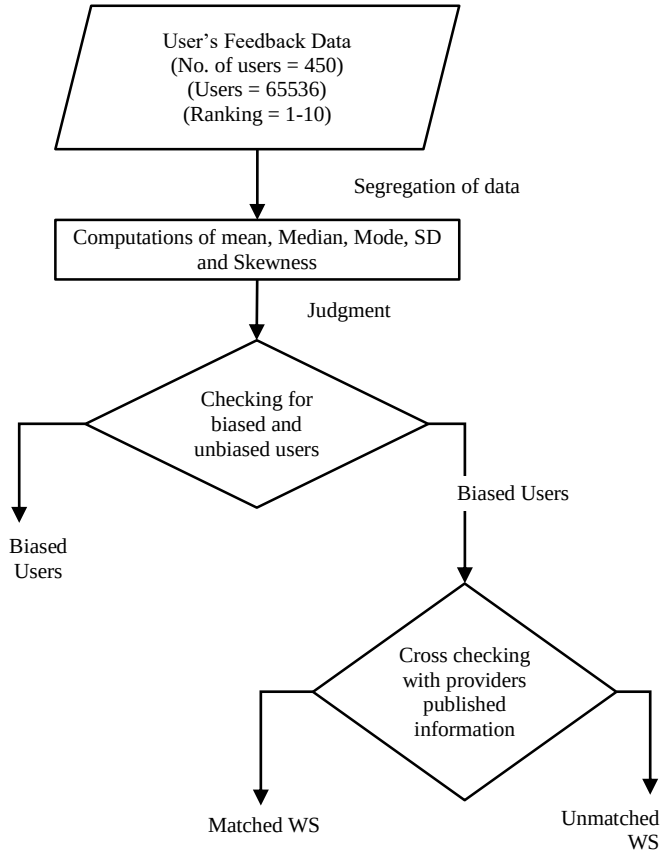

Fig.3. Flowchart of Information Verification Engine (IVE)

6.2 STATISTICAL COMPUTATIONS USING SPSS

First, we have computed Users' Mean Ranking, Mode, Median and Standard Deviation using SPSS version 21.0 and an actual feedback rating data set from a real online dating service (Libimseti) [12]. Then computed Skewness for the Web Services using these values that helped us detecting and segregating biased and unbiased Users or say helped us identifying and extracting web services with offending feedback ratings. Since our ultimate goal was to find out unbiased users. We segregated unbiased users from the biased ones. This is how we have successfully implemented Users' Feedback Module which is one of the components of our proposed Information Verification Engine (IVE). Further, to implement the second module of the IVE, we have cross verified values of the resultant filtered users' feedbacks with the Providers' published information. This technique offered amazing results and further filtered out web services helping us to achieve a set of more refined web services said to be the best candidate web services to be used for the composition. As stated, we have taken into account actual feedback rating data set from a real online dating service (Libimseti) [12]. We have considered following number of Data:

- Total no. of Web Services ~ 450
- Total no. of Users ~ 65536
- Ratings ranging between ~ 1 to10

Computation of Skewness values helped us finding biased and unbiased Users. Skewness Values ranging between [-0.5 to +0.5] were considered as Unbiased Users. Rest others as biased or highly biased.

- Out of 450 Web Services: Unbiased/Genuine - 279 and Biased/ highly biased - 171

Successful implementation of the first module of IVE resulted in achieving a set of unbiased users (segregating biased and unbiased users). This set of web services with unbiased Users is further scrutinized in the second step. We have cross verified Unbiased Users Rankings with the Providers' published information and computed the difference between the twos. Difference of values helped us achieve a further refined form of the set of web services. Web Services with the difference values ranging between [-0.1 to +0.1] were only taken into account and were assumed as the best web services to be recommended for the final composition resulting in optimized composition of the web services.

83	Web Services found close in Ranking provided by both (Unbiased Users and Providers)
79	Web Services ranked by Unbiased Users
171	Web Services ranked by Biased Users

We then plotted a graph using both of the values of Users and Providers depicted in the following table, highlighting closeness between the twos.

Table.1. Exclusive Data of Users and Providers (used in plotting a Graph (Fig.4) to show their relationship)

Web Services	Users' Mean Rank	Ranks by Providers	Web Services	Users' Mean Rank	Ranks by Providers
6	5.69	6	232	6.00	6
8	6.38	7	233	5.69	6
9	5.86	6	237	5.49	6
25	6.00	6	242	5.04	6
26	6.51	7	247	5.58	6
34	6.04	7	250	6.96	7
46	5.85	6	267	6.23	7
47	5.48	6	269	5.41	6
65	5.48	6	272	5.84	6
68	5.71	6	275	5.87	6
71	6.55	7	284	7.02	7
72	6.50	7	292	5.90	6
77	6.43	7	298	5.39	6

91	6.09	7	303	5.46	6
97	5.80	6	306	5.67	6
100	5.62	6	310	5.74	6
105	5.33	6	328	6.40	7
107	6.16	7	332	5.34	6
113	5.68	6	345	6.38	7
115	6.71	7	348	5.55	6
119	5.70	6	353	5.65	6
121	5.95	6	367	6.09	7
126	6.37	7	370	5.80	6
130	6.04	7	374	5.77	6
139	5.72	6	375	5.09	6
143	5.93	6	381	6.05	7
144	5.62	6	383	6.19	7
147	5.69	6	386	6.04	7
172	5.34	6	404	6.06	7
181	5.94	6	408	5.78	6
187	5.59	6	413	5.78	6
188	5.93	6	417	5.86	6
190	6.03	6	418	6.27	7
191	5.58	6	420	6.40	7
197	6.30	7	423	5.62	6
199	5.41	6	425	5.86	6
209	6.38	7	432	5.58	6
221	6.12	7	436	5.79	6
225	5.92	6	443	6.36	7
226	5.91	6	445	5.60	6
227	5.71	6	448	5.76	6
228	6.93	7	Total no. of Web Services = 83		

6.3 THEORETICAL ANALYSIS

We are presenting theoretic analysis here for our proposed Innovative Composition Framework. At the first, efficiency of the proposed Composition Framework is discussed then the Prevention Scheme pertaining to malicious feedback ratings is presented in detail.

6.3.1 Efficiency of the Innovative Composition Framework:

What has been observed in our study is that researchers while proposing a new framework or mechanism for the composition, relied on either Provider's published information or users' supplied feedbacks which led to wrong selection of component web services. This in turn adversely affected final composition outcome. To address this issue, we have come out with a novel composition mechanism.

The novelty of our research work lies in presenting an innovative framework (for the selection and composition of web services) taking into account both of the information altogether i.e. users' supplied feedback ratings and Provider's published information for web services.

The innovative idea here is the induction of IVE i.e. Information Verification Engine. Prior to sending selected web services to the Composer for the final composition, this IVE plays very important role by cross verifying QoS related information furnished by Service Providers in terms of published information and by service Users in terms of the feedbacks for the said web services. This move ascertains selection of the best web services for the composition, which is the sole motive of this research work. This feature undoubtedly makes our proposed framework more reliable and efficient one.

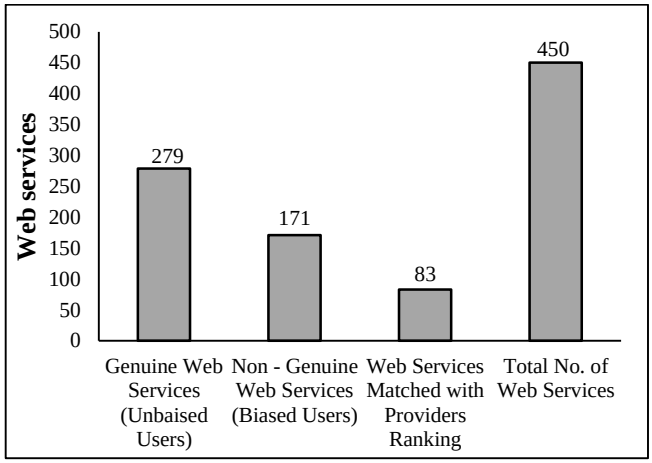

Fig.5. Summary Graph showing Data of Web Services

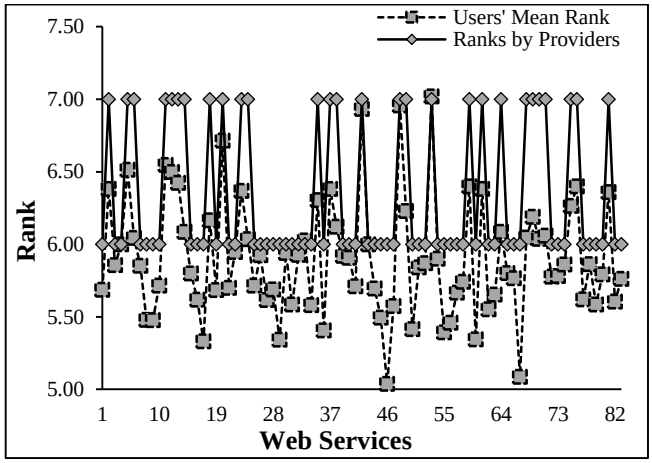

Fig.4. Graph showing Data of Web Services for whom Feedback Rankings provided by Users and Providers were found close

6.3.2 Proposed Prevention Scheme:

Moreover, we simply do not take into account Users' supplied feedbacks as it is, rather we apply statistical computations in order to discard and filter out malicious feedback ratings. We take into account only benign and genuine feedback ratings for the services. Carrying out statistical computations in order to ascertain selection of best web services comprises three important phases:

- Malicious Rating Detection,
- Discarding/ filtering out Malicious Feedback Ratings and
- Cross Verifying these resultant filtered (unbiased) users' feedbacks with the Providers' published information.

As stated, only Users' factual feedbacks are taken into account, feedback ratings associated with each of the Web services are computed identifying and discarding biased/malicious feedbacks rating. Karl Pearson's Coefficient method of Skewness is used to identify and segregate Users' biased and unbiased feedbacks rating. Hence, our proposed composition framework is found to be more efficient and reliable in the light of the above discussion.

7. CONCLUSION AND FUTURE WORK

This paper is an extension of our previous paper in which we had presented an innovative framework for automated and dynamic web service composition incorporating a novel Engine named as Information Verification Engine (IVE). This paper has nicely presented implementation of IVE achieving factual Users' feedbacks, carrying out various statistical computations using SPSS version 21.0. These statistical computations helped identify and extract web services with offending feedback ratings or say helped detect and segregate malicious/ biased users' feedbacks and unbiased/ genuine users' feedbacks. We started with 450 no. of web services and achieved a set of 279 web services rated by unbiased users (isolating those rated by malicious/ biased users). We then Cross verified ratings of these genuine web services with Provider's published information and extracted those web services having similar or near to similar ratings. This has further reduced no. of web services to 83. These resultant web services will then be used as component web services for the final composition. The said statistical computation was proved to be vital in achieving objectives of IVE helping select best web services which in turn leads to optimized composition of web services. Thus, we have successfully implemented the novel feature of our Innovative Framework i.e. IVE and hence, the defined goal is achieved. As far as future work is concerned, this includes implementation of our proposed Composition Framework as a whole which can be done with the help of the existing/ prevailing techniques of Web Services Composition. Proposing a robust and more reliable composition technique that best suits to our proposed framework would be the part of our future and on-going research work.

REFERENCES

[1] Chandrashekar Jatoth, G.R. Gangadharan, Ugo Fiore and Rajkumar Buyya, "QoS-Aware Big Service Composition using MapReduce based Evolutionary Algorithm with Guided Mutation", *Journal of Future Generation Computer Systems*, Vol. 86, pp. 1008-1018, 2018.

[2] Guobing Zou, Yanglan Gan, Yixin Chen, Bofeng Zhang, "Dynamic Composition of Web Services using Efficient Planners in Large-Scale Service Repository", *Journal of Knowledge-Based Systems*, Vol. 62, pp. 98-112, 2014.

[3] Chia-Feng Lin, Ruey-Kai Sheu, Yue-Shan Chang and Shyan-Ming Yuan, "A Relaxable Service Selection Algorithm for QoS-based Web Service Composition", *Journal of Information and Software Technology* , Vol. 53, No. 12, pp. 1370-1381, 2011

[4] Jose Antonio Parejo, Sergio Segura, Pablo Fernandez and Antonio Ruiz-Cortes, "QoS-Aware Web Services Composition using GRASP with Path Relinking", Journal of Expert Systems with Applications, Vol. 41, No. 9, pp. 4211-4223, 2014.

[5] Daewook Lee, Joonho Kwon, Sangjun Lee, Seog Park and Bonghee Hong, "Scalable and Efficient Web Services Composition based on a Relational Database", *Journal of Systems and Software*, Elsevier, Vol. 84, No. 12, pp. 2139-2155, 2011.

[6] Dandan Wang, Hao Ding, Yang Yang, Zhenqiang Mi and Zenggang Xiong, "A Feasibility-Enhanced Approach for QoS-Based Web Service Composition in Cloud Environment", *Proceedings of IEEE International Conference on Cloud Computing and Big Data*, pp. 67-72, 2015.

[7] A. Erdinc Yilmaz and Pinar Karagoz, "Improved Genetic Algorithm based Approach for QoS Aware Web Service Composition", *Proceedings of IEEE International Conference on Web Services*, pp. 463-470, 2014.

[8] Zaki Brahmi and M.M. Gammoudi, "QoS-Aware Automatic Web Service Composition based on Cooperative Agents", *Proceedings of IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 27-32, 2013.

[9] Zhenpeng Liu, Junbao Li, Jimin Li, Aiguo An and Jianmin Xu, "A Model for Web Services Composition Based on QoS and Providers' Benefit", *Proceedings of 5th* IEEE *International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 12-18, 2009.

[10] Yang Yu, Hui Ma and Mengjie Zhang, "An Adaptive Genetic Programming Approach to QoS-aware Web Services Composition", *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 1740-1747, 2013.

[11] Khozema Ali Shabbar, Tarun Shrimali and Mohammed Sha "Computation of QoS While Composing Web Services", *International Journal of Advanced Computer Science and Applications*, Vol. 8, No. 3, pp. 309-317, 2017.

[12] L. Brozovsky and V. Petricek, "Recommender System for Online Dating Service", *Proceedings of IEEE International Conference on Information Retrieval and Software Engineering*, pp. 1-12, 2007.

[13] El-Alami Ayoub and Hair Abdellatif, "An Agent Architecture for QoS-based Web Service Composition using the Skyline Algorithm", *International Journal of Informatics and Communication Technology*, Vol. 6, No. 3, pp. 179-188, 2017.

[14] Samia Sadouki Chibani and Abdelkamel Tari, "Elephant Herding Optimization for Service Selection in QoS-Aware Web Service Composition", *International Journal of Computer and Information Engineering*, Vol. 11, No. 10, pp. 1124-1128, 2017.