

An Improved Approach for QoS Based Web Services Selection Using Clustering

Mourad Fariss*, Naoufal El Allali, Hakima Asaidi, Mohamed Bellouki

MASI Laboratory, FPN, UMP, Nador, 62000, Morocco

ARTICLE INFO

Article history:

Received: 22 December, 2020

Accepted: 11 March, 2021

Online: 20 March, 2021

Keywords:

Web service selection

K-Means clustering

Prefiltering

Skyline techniques

BBS algorithm

ABSTRACT

With the rising number of web services created to build complex business processes, selecting the appropriate web service from a large number of web services respond to the same client request with the same functionality are developed independently but with different quality of service (QoS) attributes. From this point, there are many approaches to web service selection. Nevertheless, this is still deficient due to a considerable number of discovered web services. The prefiltering is a solution to reduce the number of web services candidates. In this paper, the K-means clustering is applied to determine similar services based on QoS information. The results of this prefiltering are considered at the selection task using the Branch and Bound Skyline (BBS) algorithm. The experimental evaluation performed on real Dataset proves that our approach presents efficient results for web service selection.

1. Introduction

This paper is an extension of [1], where an advanced mechanism of prefiltering and selection of web services based on QoS is proposed.

Over the past decade, many researchers have developed a strong interest in web services, an important standard of Service Oriented Architecture (SOA). It is a novel paradigm to build the large-scale of distributed applications. Web service is defined as a software-system and identified using an URI, where its public interface and binding description use the XML language, can be discovered and invoked by other web services. This invocation requires a prescribed of resources using XML messages via such protocols of the Internet.

WSDL, SOAP, and UDDI are the series of technology criteria for web services [2], on which other technologies closer to the application problem can be specified and implemented. It presents standard web service protocols to implement /develop the interaction between applications (services) among diverse platforms. The web service architectures are based on the following three entities; (i) service provider, (ii) service registry, and (iii) service customer. The service provider corresponds to the proprietor of the service. It is required to depict the web service and publish it in the service registry (a central entity). The service registry possesses the technical details of web service and the service provider information to facilitate and find services for

customers. The customer is the application that is going to search for and invoke a service. The client application can itself be a web service.

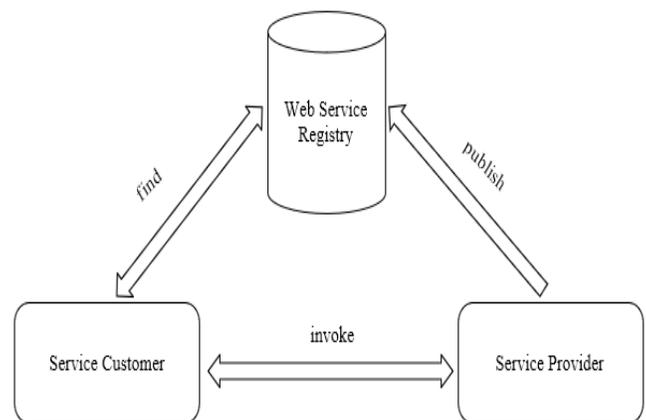


Figure 1: Web Service Model

The increased web applications usage for different fields, making service providers to respond to customers by releasing an enormous number of web services ; the customer finds a problem in choosing the web service that meets his request with this large number of published web services. QoS appears as a solution to help customers select an adequate web service that meets the

*Corresponding Author: Mourad Fariss, m.fariss@ump.ac.ma

customers' requirements. These requirements seek to benefit from more web service performances as cost, response-time, and other QoS properties [3].

One of the immense challenges of SOA is to attribute QoS to the description of web services to facilitate the choice for customers according to their requirements as well as to dynamically select the most efficient web services for each customer according to the criteria of the requirements given by the client or just the best web service among the web services found in the web services registry that have similar functionalities.

The rest of the paper is presented as follows: Section 2 gives the related works. Section 3 explains the background used in this paper as the QoS in web service, the K-Means clustering and the BBS algorithm. Section 4 contains our proposed approach. Section 5 discusses the experimental results. The final section concludes the paper.

2. Related works and motivation

The selection of web service is a hard process because various web services offer similar features. Applications in their consumption of services struggle to employ the optimal QoS; however, the selection phase faces many hardships since the QoS is at the same time influences by several inconsistent QoS features. Present solutions have a shortage in performance for the reason that they take in consideration the potential web services to find QoS features. In case, customer needs are taking place in the selection process. We can use this bit of information in order to distinguish between web services that possess similar QoS as end-user QoS features. For the sake of gathering similar web services together, it is useful to use cluster technology with reference to QoS properties. The selection considers only web services, which is abided by the customer's QoS requirements.

Recently, to solve the selection web service problem, the skyline algorithms have been introduced by selecting service as the optimal candidate services [4–9]. The BBS approach is the most famous skyline algorithm suggested by [10]. As far as the large data spaces are concerned, it is the most efficacious algorithm.

The authors of [11] compared two algorithms: the BBS and the SFS algorithms on the service web selection. The service selection system was performed efficaciously and reliably by the BBS algorithm as the experimental outcomes show.

Authors in [12] were the first to propose the filtering of web services system named “*F-WebS system*”. The system builds the performance on the description and discovery area of web services [11,13–15] as semantics-based web service filtering and utilizes a variety of matching algorithms such as those in the discovery task.

In [7], the authors proposed a framework named “*KRSWS*” to reduce the web services candidates based on QoS attributes and the customer requirements, the proposed method use the Fuzzy AHP method and a new version of Promethee [16].

The clustering mechanism for generating services clusters according to the same QoS is useful for determining the relevant web service [17]. Nonetheless, to cluster the generic type of QoS properties in one group can negatively effect the efficiency of the selection of web service [18].

The authors of [19] proposed a cluster and filtering system architecture model, and have demonstrated that the use of the clustering technique does not effect on the system's accuracy and pertinence, yet it increases the speed of the process of simple service processing.

The proposed method for selecting web services uses a cluster approach based on QoS parameters to pre-filter web services, once we filter the web services with a pre-filter based on K-Means clustering, we obtain a decreased set that contains the web services filtered, after that we select the dominate web service with the skyline technique. The proposed solution presents a significant precision and performance of selecting web services regarding other approaches cited in the literature.

3. Preliminaries

3.1. QoS in Web Service

Several works that have been conducted in web service discovery focus only on the functional features (content requirements) of a web service, but this phase remains insufficient to meet the customer's requirements because of the many similar web services that offer the same functionalities for the customer. However, the web service selection phase introduces the notion of non-functional features (context requirements), namely QoS, to determine the most efficient web service that meets customer requirements.

QoS is a set of features and characteristics of an entity or a service that gives it the ability to meet stated or implicit needs. The needs can be linked to parameters such as accessibility, availability, response time, reliability, cost, etc. The parameters can help to select from the candidate web services and reduce the consumed time.

We can distinguish between customer-independent QoS attributes such as (cost, reputation, accuracy) and customer-dependent QoS attributes such as (throughput, scalability, availability) [2], [3].

QoS has the capacity to satisfy its significance by:

- Defining the operational measurements for the web service.
- Distinguishing between providers and services.
- Filtering and ranking the web services.
- Selecting the efficient and appropriate service that achieves the whole customer needs.

Those QoS attributes can be classified into six categories as shown in Table 1.

Table 1: QoS Categories and Attributes [20].

QoS categories and attributes	1. Service Provider	Service provider reputation, Accountability, Throughput, Scalability, Availability.
	2. Service Customer	Response time, Reliability,

		Usability, Cost, Discoverability.
	3. Service Developer	Maintainability, Interoperability, Composability, Reusability, Stability, Traceability, Testability.
	4. Service runtime management	Exception handling, Completeness, Robustness.
	5. Security	Authentication, Confidentiality, Authorization, Non repudiation, Auditability, Encryption, Integrity.
	6. Network infrastructure	Server failure, Guaranteed messaging, Bandwidth, Delay time, Packet loss ratio.
	7. Network infrastructure	Server failure, Guaranteed messaging, Bandwidth, Delay time, Packet loss ratio.

3.2. K-Means Clustering

The clustering algorithms are classified into hierarchical clustering, exclusive clustering, probabilistic clustering, and superimposed clustering [21]. The K-Means clustering can be considered the most known to resolve many clustering problems. Our use for this algorithm is to rank web service applications according to the QoS attributes.

The advantages of the K-Means clustering are as follows [22]:

- The larger number of the variables are, the smaller the number of the clusters, and the smaller the speed of calculation than the hierarchical clustering algorithms.
- If the clusters are globular, K-Means will produce tighter clusters, which will be tighter than hierarchical clusters.

Despite all of these advantages, K-Means has also some limitations, but these later ones do not influence our approach to study the QoS between queries and published web services.

The goal of applying the K-Means clustering algorithm is to classify the database of QoS attributes offered by the list of discovered web services. This is done in multiple steps. The first is to randomly initialize the number K of centroids. The centroids represent the centers of the clusters. The following process takes place in two stages called expectation and maximization; these two stages involve assigning each data element to its nearest centroid. Next, the algorithm calculates the new centroid of all the points of

each cluster and define the new centroid. The following algorithm describes the K-Means algorithm's steps:

Algorithm 1: K-Means algorithm

- 1: Define K // The number of clusters
- 2: Initialize K centroids (Randomly)
- 3: repeat
- 4: **expectation:** Assign the points to their closest centroid
- 5: **maximization:** Calculate the new centroid of clusters
- 6: **until:** The position of centroid does not change

3.3. Branch and Bound Skyline Algorithm

The BBS algorithm is considered as an enhancement of the K Nearest Neighbors (KNN) algorithm with a difference that the BBS algorithm crosses the R-tree only once. The algorithm uses a priority queue, where data points are organized according to their minimum distances (*mindist*) or minimum bounding rectangles (*MBR*) from an origin point. A minimal bounding rectangle is used to evaluate a complex shape. It is a rectangle with parallel sides to the x and y-axis and minimally surrounds the utmost complex shape [23].

The algorithm chooses at each step, among all the unvisited points, the closest tree points to the origin. In addition, it keeps these discovered points in a set S for the validation step of dominance.

The description of the BBS algorithm is given as follows:

Algorithm 2: BBS Algorithm

- 01: $P = \emptyset$ // P is a set of dominant points
- 02: fill the root R by all entries in the heap
- 03: while R not empty yet do
- 04: c is removed // c is the top entry from R
- 05: if c is dominated by other points in P remove c
- 06: else
- 07: if c is an intermediate entry
- 08: for each child c_i of c
- 09: if c_i is not dominated by some point in P
- 10: insert c_i into heap
- 11: else // c is a data point
- 12: | insert c_i into P
- 13: end
- 14: end
- 15: end
- 16: end

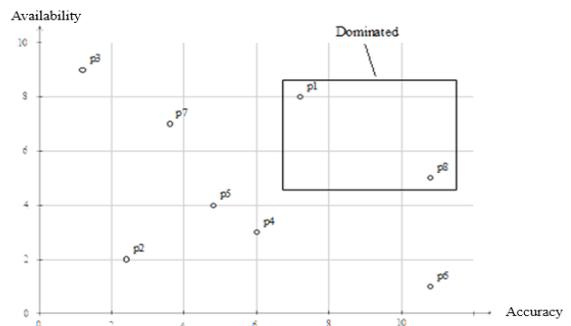


Figure 2: Example of BBS Algorithm

The BBS algorithm performs better than other skyline algorithms, ensuring a minimum cost of input/output, and the number of R-tree node access, and processing time. However, since the number of attributes is increased, the number of points in the skyline is increased substantially [24]. Hence, the idea of using a filter is to decrease the number of candidate points.

Figure 2 provides an example of domination point using two attributes of QoS.

4. Proposed Approach

The QoS-based selection consists to choose the best web service from the candidate (discovered) web services to satisfy the customer's non-functional requirements as QoS needs. This selection depends on the specification adopted when defining the QoS criteria and the QoS profile of the web service.

For solving this problem, we propose to add a filter with K-Means clustering as a first step to generate the clusters of web services based on the QoS properties assigned to each discovered web service. This technique determines the number K of classes as an input and generates the K clusters. When the process starts, it chooses centers randomly. Then at each step, it recalculates the new cluster centers as the mean of the QoS for this cluster. The criterion function used in this step is expressed in equation (1) [25] below:

$$E = \sum_{i=1}^k \sum_{ws \in C_i} |ws - Mi|^2 \quad (1)$$

where E is the error value between the consumer 'constraints and the candidate web services, ws refers to the candidate web service, and Mi is the mean of the cluster C_i which contains ws . Each cluster created by the K-Means algorithm contains web services with similar QoS attributes. The algorithm serves as a pre-filter for the discovered web services. The cluster obtained from this phase has the most efficient centroid to meet customer requirements while all web services in this cluster can also meet these requirements, and they are affected to step two.

The second step is to exploit the filter results and apply the BBS algorithm on the filtered cluster web services. The objective of this step is to determine the dominant web services among the obtained cluster web services using their QoS properties. Eliminating inappropriate web services in the pre-filter phase with K-Means clustering makes it easier for the BBS algorithm to find the most appropriate web service and meets customer requirements.

Figure 3 summarizes the model of our proposed approach. The customer submits a request for a service that meets their needs and requirements. In the discovery stage, the web service registry determines a set of candidate web services that can meet customer needs. In the web service selection stage, we propose to add a pre-filter using K-Means algorithm which minimizes the candidate web services. This step takes place by creating clusters that contain web services including similar QoS properties and determining which cluster meets the customer requirements, thereby the BBS algorithm is applied to find the appropriate web service for customer needs and requirements.

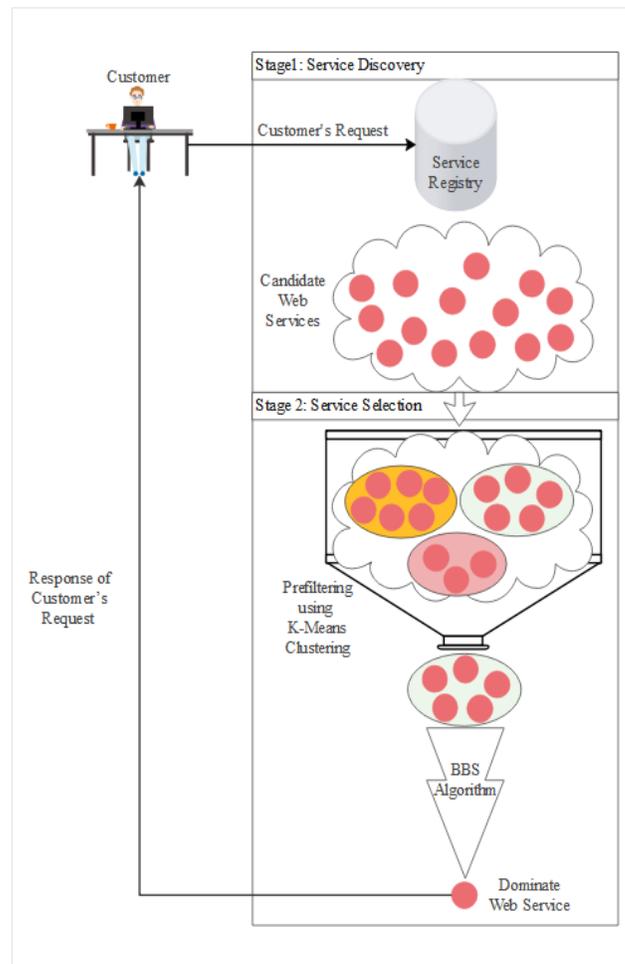


Figure 3: Web Service Selection Model

5. Experimental Results

The evaluation of the proposed approach aims to show the interest of adding a pre-filter to a web service selection system. The pre-filter proposed in our approach is based on K-Means clustering.

A real-world dataset of QoS attributes named QWSDataset [26] is used for experimentation. The dataset contains 9 QoS attributes per service, downloaded by a web service Crawler Engine [27]. The used version contains 2507 web services. This dataset contains 11 segments representing 9 QoS attributes, the URL of WSDL file and the service name. The database is used in experiments to prove the efficiency and performance of our proposed approach. Table 2 represents some values of QWSDataset used in the experimentation.

To reduce the search domain, we apply the K-Means algorithm to determine the web services that have similar QoS parameters. The selection process is conducted using the BBS algorithm. The performance and efficiency of our proposed approach are verified using the evaluation metrics: Success rate and execution time for a selection process with different approaches.

5.1. Success Rate

The success rate (SR) of all selected web services is the proportion of customers' QoS requirements (C_i) to the QoS values

Table 2: QoS Values of QWSDataset.

Service Name	Response Time	Availability	Throughput	Successability	Reliability	Compliance	Best Practices	Latency	Documentatio
MAPPMatching	302.75	89	7.1	90	73	78	80	187.75	32
Compound2	482	85	16	95	73	100	84	1	2
USDADData	3321.4	89	1.4	96	73	78	80	2.6	96
GBNIRHoliday Dates	126.17	98	12	100	67	78	82	22.77	89
CasUsers	107	87	1.9	95	73	89	62	58.33	93

of these web services ($\bar{U}_i(S)$). The success rate of a web service (SRn) equals 1 if its value is greater than a threshold value (ts).

$$SR = \frac{SRn}{n} \times 100\% \tag{2}$$

and

$$SRn = \sum_{i=1}^n \begin{cases} 1, \cap_{i=1}^n \frac{c_i}{\bar{U}_i(S)} \geq ts \\ 0, otherwise \end{cases} \tag{3}$$

where SRn is the success rate for n web services. Based on [11], the values of parameters used in the experimentation are $ts=0.86$ and $n=200$.

Figure 4 presents the success rate of our approach compared to other approaches depending on the number of candidate web services. The success rate is increased for our approach compared to other approaches. Furthermore, the more the number of candidate web services is increased, the more the success rate of our approach is increased, which means that the proposed approach is scalable for the large dataset of web services. However, the other approaches have a stable success rate or a decreased success rate when the number of candidate web services is increased.

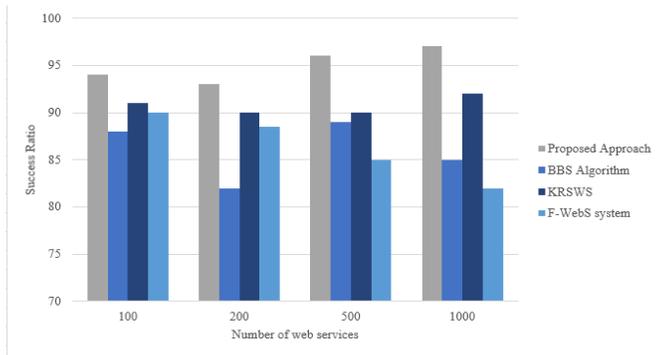


Figure 4: Success Rate Comparison with QWSDataset.

5.2. Computation Time

To evaluate the efficiency of our approach in comparison with other ones, the execution time is computed as shown in Figure 5. The execution time for our approach is minimized than other approaches depending on the number of web services. As a consequence, our approach has a high performance considering the execution time even we deal with a large dataset.

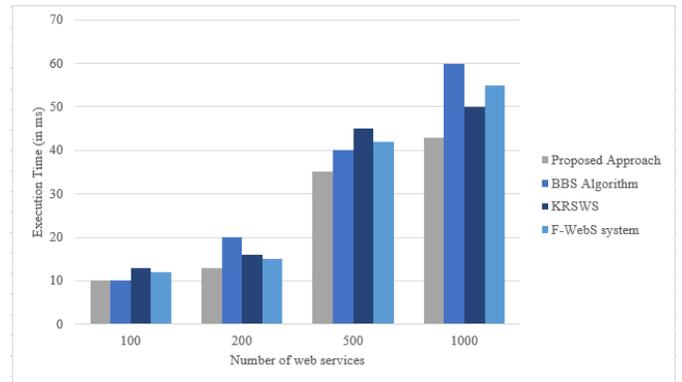


Figure 5: Time Execution Comparison.

After that, we have calculated the execution time by modifying the number of QoS attributes from 3 to 9. This modification is performed to examine the impact of QoS attributes' number to select the adequate web service. Moreover, the comparison between the proposed approach with other approaches is performed in terms of the execution time. As illustrated in Table 3, the execution time for the compared approaches shows a high performance of our approach, which uses the K-Means clustering as a pre-filter. The effectiveness of our approach is proved by increasing the number of QoS attributes.

Table 3: Execution time according to QoS attributes.

Number of QoS attributes	Execution Time (ms)			
	BBS Algorithm	F-WebS system	KRSWS	Proposed Approach
3	5	7	6	4
4	7	7	8	6
5	12	10	11	10
6	15	16	17	13
7	17	19	17	15
8	34	30	32	28
9	60	55	53	43

Adding a pre-filter to the web service selection process has allowed us to improve the success rate up to 97% and minimize the execution time of web service selection compared to other approaches. This is due to the elimination of the inappropriate

web services using a new pre-filer mechanism that is based on the K-Means clustering.

6. Conclusion

The selection web service problem consists to find the most adequate web service from a large dataset with a short period of time based on the QoS proprieties. To resolve this problem, we proposed a new mechanism using the K-Means clustering as a pre-filter to eliminate the inappropriate web services, which leads to minimizing the search space. Then the BBS algorithm is applied to select the dominant web service for increasing the precision rate. The experimentation results show a high performance of our approach compared to other ones in terms of the scalability, the execution time, and the precision rate.

The current work will provide many benefits and advantages to end-users, practitioners, and researchers who deal with a large data set of web services. It allows to prefilter an enormous number of web services that are generated from different systems such as smart health, smart agriculture, smart city, etc.

In the future, we can use this approach to resolve the web service composition problems and minimized the composition time, and we try to integrate uncertain QoS parameters in our approach.

References

- [1] M. Fariss, N. El Allali, H. Asaidi, M. Bellouki, "Prefiltering Approach for Web Service Selection Based on QoS," Proceedings - 2019 4th International Conference on Systems of Collaboration, Big Data, Internet of Things and Security, SysCoBioTS 2019, 1-5, 2019, doi:10.1109/SysCoBioTS48768.2019.9028043.
- [2] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana, "Unraveling the Web services Web: An introduction to SOAP, WSDL, and UDDI," IEEE Internet Computing, 6(2), 86-93, 2002, doi:10.1109/4236.991449.
- [3] S.Y. Hwang, C.C. Hsu, C.H. Lee, "Service selection for web services with probabilistic QoS," IEEE Transactions on Services Computing, 8(3), 467-480, 2015, doi:10.1109/TSC.2014.2338851.
- [4] M. Alrifai, D. Skoutas, T. Risse, "Selecting skyline services for QoS-based web service composition," in Proceedings of the 19th international conference on World wide web - WWW '10, ACM Press, New York, New York, USA: 11, 2010, doi:10.1145/1772690.1772693.
- [5] H.T. Kung, F. Luccio, F.P. Preparata, "On Finding the Maxima of a Set of Vectors," Journal of the ACM (JACM), 22(4), 469-476, 1975, doi:10.1145/321906.321910.
- [6] C.H. Papadimitriou, M. Yannakakis, "Multiobjective query optimization," in Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Santa Barbara, California, USA: 52-58, 2001, doi:10.1145/375551.375560.
- [7] A. Ouadah, A. Hadjali, F. Nader, "A Hybrid MCDM Framework for Efficient Web Services Selection Based on QoS," in 2018 International Conference on Applied Smart Systems (ICASS), IEEE: 1-6, 2018, doi:10.1109/ICASS.2018.8652037.
- [8] S.S. Joshi, O.B.V. Ramanaiyah, "An integrated QoE and QoS based approach for web service selection," in 2016 International Conference on ICT in Business Industry & Government (ICTBIG), IEEE: 1-7, 2016, doi:10.1109/ICTBIG.2016.7892671.
- [9] R. Karim, C. Ding, C.-H. Chi, "An Enhanced PROMETHEE Model for QoS-Based Web Service Selection," in 2011 IEEE International Conference on Services Computing, IEEE: 536-543, 2011, doi:10.1109/SCC.2011.81.
- [10] Marios Kokkodis., "Implementation Of Skyline Query Algorithms," 2003.
- [11] M. Fariss, H. Asaidi, M. Bellouki, "Comparative study of skyline algorithms for selecting Web Services based on QoS," Procedia Computer Science, 127, 408-415, 2018, doi:10.1016/j.procs.2018.01.138.
- [12] W. Abramowicz, A. Godlewska, J. Gwizdala, M. Kaczmarek, D. Zyskowski, "Application-Oriented Web Services Filtering," in: IEEE, ed., in International Conference on Next Generation Web Services Practices (NWEsp'05), IEEE, Los Alamitos: 63-68, 2005, doi:10.1109/NWESP.2005.17.
- [13] L. Li, I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology," International Journal of Electronic Commerce, 8(4), 39-60, 2004, doi:10.1080/10864415.2004.11044307.
- [14] K. Lynch, D., Keeney, J., Lewis, D., O'Sullivan, "A Proactive Approach to Semantically-Driven Service Discovery," in the Proceedings of 2nd Workshop on Innovations in Web Infrastructure, Edinburgh, 2006.
- [15] N. Srinivasan, M. Paolucci, K. Sycara, "Semantic Web Service Discovery in the OWL-S IDE," in Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), IEEE: 109b-109b, 2006, doi:10.1109/HICSS.2006.431.
- [16] E. Al-Masri, Q.H. Mahmoud, "QoS-based discovery and ranking of Web services," Proceedings - International Conference on Computer Communications and Networks, ICCCN, 529-534, 2007, doi:10.1109/ICCCN.2007.4317873.
- [17] Y. Xia, P. Chen, L. Bao, M. Wang, "A QoS-Aware Web Service Selection Algorithm Based On Clustering," in 2011 IEEE International Conference on Web Services, 428-435, 2011, doi:10.1109/ICWS.2011.36.
- [18] M. Ahmadi Oskooei, S. Mohd Daud, "Quality of service (QoS) model for web service selection," in 2014 International Conference on Computer, Communications, and Control Technology (I4CT), IEEE: 266-270, 2014, doi:10.1109/I4CT.2014.6914187.
- [19] W. Abramowicz, K. Haniewicz, M. Kaczmarek, D. Zyskowski, "Architecture for Web Services Filtering and Clustering," in Second International Conference on Internet and Web Applications and Services (ICIW'07), IEEE: 18-18, 2007, doi:10.1109/ICIW.2007.19.
- [20] K. Kaewwanjong, S. Intakosum, "QoS Attributes of Web Services: A Systematic Review and Classification," Journal of Advanced Management Science, 3(3), 194-202, 2015, doi:10.12720/joams.3.3.194-202.
- [21] J. Han, M. Kamber, "Data mining concepts and techniques San Francisco Moraga Kaufman," 2001.
- [22] N. Arunachalam, A. Amuthan, C. Kavya, M. Sharmilla, K. Ushanandhini, M. Shanmughapriya, "A SURVEY ON WEB SERVICE CLUSTERING," in 2017 International Conference on Computation of Power, Energy Information and Communication (ICCPEIC), IEEE, Melmaruvathur, India: 247-252, 2017, doi:10.1109/ICCPEIC.2017.8290371.
- [23] J. Wood, Minimum Bounding Rectangle, Springer US, Boston, MA: 660-661, 2008, doi:10.1007/978-0-387-35973-1_783.
- [24] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," 217-235, 1999, doi:10.1007/3-540-49257-7_15.
- [25] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques: Concepts and Techniques (3rd Edition), 3rd ed., San Francisco, CA, USA: Morgan Inc, Kaufmann Publishers, 2012.
- [26] E. Al-Masri, Q.H. Mahmoud, "Investigating web services on the world wide web," in Proceeding of the 17th international conference on World Wide Web - WWW '08, ACM Press, New York, New York, USA: 795, 2008, doi:10.1145/1367497.1367605.
- [27] E. Al-Masri, Q.H. Mahmoud, "Crawling multiple UDDI business registries," 16th International World Wide Web Conference, WWW2007, 1255-1256, 2007, doi:10.1145/1242572.1242794.