

## An effective method for clustering-based web service recommendation

Ha Huy Cuong Nguyen<sup>1</sup>, Bui Thanh Khiet<sup>2</sup>, Van Loi Nguyen<sup>1</sup>, Thanh Thuy Nguyen<sup>3</sup>

<sup>1</sup>Faculty of Computer Science, Vietnam - Korea University of Information and Communication Technology, The University of Danang, Danang, Vietnam

<sup>2</sup>Institute of Engineering and Technology, Thu Dau Mot University, Binhduong, Vietnam.

<sup>3</sup>Faculty of Computer Science, VNU University of Engineering and Technology, Hanoi, Vietnam

### Article Info

#### Article history:

Received Jun 28, 2021

Revised Sep 16, 2021

Accepted Oct 10, 2021

#### Keywords:

QoS prediction

Service-oriented computing performance

Web service recommendation clustering

### ABSTRACT

Normally web services are classified by the quality of services; however, the term quality is not absolute and defined relatively. The quality of web services is measured or derived using various parameters like reliability, scalability, flexibility, and availability. The limitation of the methods employing these parameters is that sometimes they are producing similar web services in recommendation lists. To address this research problem, the novel improved clustering-based web service recommendation method is proposed in this paper. This approach is mainly dealing with producing diversity in the results of web service recommendations. In this method, functional interest, quality of service (QoS) preference, and diversity features are combined to produce a unique recommendation list of web services to end-users. To produce the unique recommendation results, we propose a varied web service classification order that is clustering-based on web services' functional relevance such as non-useful pertinence, recorded client intrigue importance, and potential client intrigue significance. Additionally, to further improve the performance of this approach, we designed web service graph construction, an algorithm of various widths clustering. This approach serves to enhance the exceptional quality, that is, the accuracy of web service recommendation outcomes. The performance of this method was implemented and evaluated against existing systems for precision, and f-score performance metrics, using the research datasets.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



### Corresponding Author:

Bui Thanh Khiet

Institute of Engineering and Technology, Thu Dau Mot University

Binhduong, Vietnam

Email: khietbt@tdmu.edu.vn

## 1. INTRODUCTION

To expand the host pool, the client sends a solicitation and acquires a reaction from utilizing web hosting. Fundamentally, web hosting can be adopted in two distinct ways. They can be utilized as straightforward web hosting that give an interface to get information sources and return yields or they can be utilized as segments that can be incorporated into business forms. The first type of usage is called individual use and the second is referred to as process use. This research work deals with recommending services with respect to the individual case.

Though web service technologies and service-oriented computing (SOC) promise to provide loose coupling among parts and dexterity to react to changes in necessities with obviously conveyed registering and lesser progressing ventures, wavelength routed (WS) is not shared and reused as expected [1]. One of the

reasons that impede the usage of such technologies and SOC is that efficient WS discovery presents many challenges [2]-[5]. Recommender systems (RS) are one tool to help bridge this gap. There are various mechanisms being employed to create RS and the common systems include two main classes, namely, content basis and collaborative filtering schemes. Content basis RS does the matching between textual information of a particular product with the textual information representing the interests of a customer. Collaborative filtering methods use designs in customer grades to recommend. Both types of RS expect notable data resources under the order of user ranks and product features; hence they are not able to generate high-quality recommendations.

Web service is related to a web services description language (WSDL) description that includes the depiction of the service. Various studies have been performed to use WSDL archives [6]-[9]. Kumar *et al.* [4] suggested that WS index Google is reasonable for giving WS closeness search. In a few cases, their gadget does not adequately acknowledge data types, which for the most part uncover significant data regarding the operation of web services [8]. Elgazzar *et al.* [9], Liu *et al.* [10] exhibited a relative technique, which conveys WSDL reports to build the non-semantic website composition appearing. They collected singular segments in WSDL records as their segments and organized web services into worth-based social issues. The clustering impacts compartment was utilized to build up the idea of web service record data.

Lausen and Haselwanter [11] executed substance mining systems to concentrate highlights, for example, service content, setting, hostname, and name, from web service portrayal documents so as to bunch web services. They proposed a consolidated component burrowing and the clustering approach concerning web services as a predecessor to exposure in order to assist in constructing a web search contraction to the edge and push non-semantic web services [12]-[15]. Maximilien and Singh [16], Sofian *et al.* [17] proposed a multi-master based structure where managers help quality-based organization certification using an office to disperse reputation and support data. Every go-between service is independent, yet furthermore cooperates with various pros to accumulate diverse suppositions and along these lines intensifies its data to improve its fundamental pro. Lausen and Haselwanter [11] executed the calculation about how to solidify undeniable quality of service (QoS) estimations to get a sense as a rule rating for a web service. The proposed reputation can be portrayed as the standard given to the organization by the end client [18]-[20].

This work proposes methods for service discovery that are lighter than those based on semantics and can be a feasible way towards the realization of service-oriented applications. We attempt to overcome the difficulties of forecasting QoS values by combining Pearson similarity and the slope one method. Our simple enhanced algorithm for ranking services considering users' requirements is better than the existing complicated algorithms. The basic purposes of this research are: i) to propose an approach to build a semantic kernel consisting of semantically similar web hosting using the various widths clustering and merging method; ii) to design new efficient and scalable algorithms for various widths clustering based web service reliability for the recommendation systems; and iii) to design, implement, and evaluate the proposed technique. This paper is organized as follows: section 1 presents the introduction and the related work. Section 2 introduces the methodology. Section 3 shows results and discussions. Finally, conclusions are presented in section 4.

## 2. PROPOSED METHOD

We are contributing to the existing web service recommendation approaches with the proposed algorithm called clustering-based system to overcome the limitation of web service recommendation. The advanced strategy will be used to develop the production of the system. This system is shown in Figure 1. All the functionalities for web services on devices (WSD) are used by the proposed method CWBR with one extra functionality, that is, clustered data. Below we define all functionalities of the method.

### 2.1. Functional evaluation

The functional appraisal can be furthermore isolated into two sections; functional estimation and utilitarian estimation. Functional estimation considers the result of the client's chronicled expectation with web services controlled to a premise-based equivalence criterion. The substance-based identity is procured by object closeness. This process simply recognizes web services that are represented by the WSD. All things considered; it is anything but difficult to stretch out our work to deal with different sorts of Web services. The client's real interest can be mined from his/her very own affiliation use or requesting history. Utilitarian estimation predicts the client's potential interest and diagrams its congruity with web services by using shared isolating based on customer comparability. This comparability is estimated depending on the web hosting summon history of all web hosting clients.

### 2.2. Non-functional evaluation

Think about that  $m$  QoS structures are working for estimating the non-utilitarian quality of  $US_i$ , its QoS vector is meant by  $SW_i$ , i.e.,  $SW_i = (q_{i,1}, q_{i,2}, \dots, q_{i,m})$ , where  $q_{i,j}$  expresses the value of the  $ij$  quality

standard. For the most part, there are two types of QoS measures. A QoS model is seen as negative if the bigger the worth, the lower the quality (e.g., cost besides reaction time). Otherwise, the QoS measure is seen as positive (e.g., Accessibility and Unwavering quality). Evaluations of different QoS criteria should be built up to a tantamount arrangement for different assessment purposes. While having a previous uniformity, it is feasible to implement the measurable strategy (i.e., Pauta Paradigm methodology) before the procedure of QoS esteems ahead of time to expel the exceptions. Here, to change each QoS standard incentive to a genuine number somewhere in the range of 0 and 1, contrasting it and the base and most extreme estimations of the QoS basis among all accessible web hosting up-and-comers are needed. After such standardization preparation, the more noteworthy incentive for the quality is, the more a model implies excellent quality.

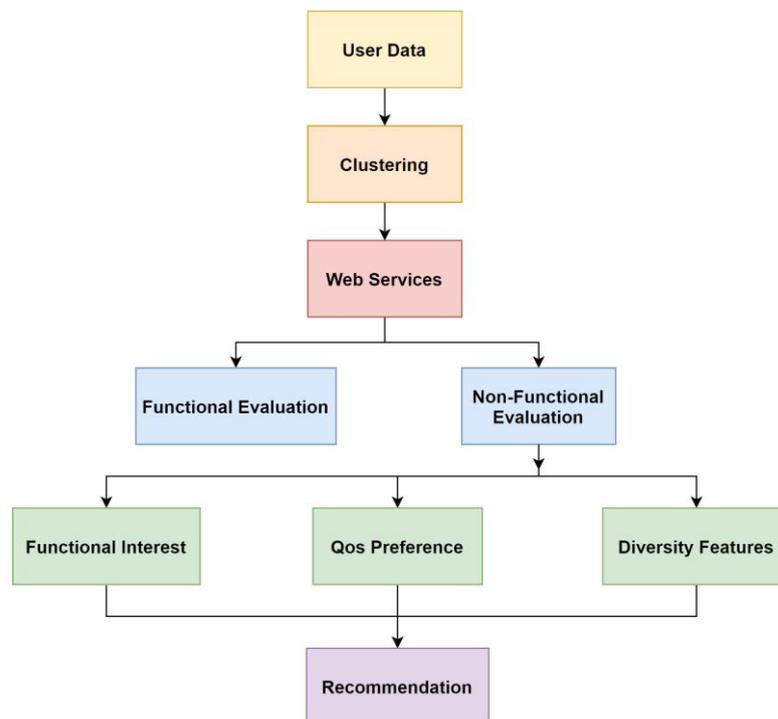


Figure 1. System architecture

Step 1. Input dataset: i) user set, ii) web service set, and iii) QoS matrix

Equation (1)  $userSim_{da,db} = 2 \times |US_{ab}| / (|S_{da}| + |S_{db}|)$ , where  $S_{da}$  and  $S_{db}$  are the numbers of web services appropriated by the user  $da$  and  $db$  respectively,  $US_{ab}$  denotes the collection of Web services utilized with both  $da$  and  $db$ , i.e.,  $US_{ab} = S_{da} \cap S_{db}$ . If  $US_{ab} = 0$ , then use  $(ua, ub) = 0$ .

Equation (2)  $(US_i, US_j) = \phi_{texSim} + \phi_{opSim}$ , where  $texSim = \cos(\mathbf{w}_i, \mathbf{w}_j) = \mathbf{w}_i \cdot \mathbf{w}_j / (|\mathbf{w}_i| \times |\mathbf{w}_j|)$  where  $|\mathbf{w}_i|$  and  $|\mathbf{w}_j|$  signify the euclidean length of the vector  $\mathbf{w}_i$  and  $\mathbf{w}_j$  respectively. Moreover, the numerator is the dot outcome of  $\mathbf{w}_i$  with  $\mathbf{w}_j$ . Where,  $opSim =$

Step 2. Various widths clustering

In this subsection, the various widths clustering part is explained, where an instructive file is appropriated into a few packages whose size is required by the client characterized limit. Three procedures are engaged with this activity: getting the group width, dividing, and blending. They are executed successively until the criteria are met. That is, the procedures of isolating and blending are ceased when the size of the greatest gathering isn't actually a customer described edge  $b$ , or when the quantity of packs with the accomplishment of the referenced methods is comparable to  $b$ . Algorithm 2 demonstrates the midpoints of the referenced plans, with the factors, information structures, and capacities utilized by the calculation.

Step 3. Cluster-width learning

Given  $D$ , an information collection to be clustered, including  $NN_k(H_i)$  be the method of  $k$ -nearest neighbors for the target  $H_i$ , and  $clsWidth$  be the method calculating the width (radius) of  $NN_k(H_i)$ , to find the appropriate worldwide width, two or three entities from  $D, H = \{H_1; H_2; \dots; H_r\}$  where  $r < |D|$  are randomly chosen, and for each entity, the span of its  $k$ -nearest neighbors is enlisted, and the ordinary is used as an overall width for  $D$  as sought after:

$$w = \frac{1}{r} \sum_{i=1}^r \text{clsWidth}(NN_K(H_i)) \quad (1)$$

This procedure segments a dataset into various clusters utilizing a huge width to arrange the result of clustering the meagerly distributed items in the n-dimensional range. Be that as it may, large clusters from thick territories will be made, for example, clusters C2 and C3. Along these lines, every huge bunch whose size surpasses a client characterized limit (greatest group size) will be separated into various clusters utilizing a width that changes the depth of that collection. This method progresses continuously until all clusters are not greater than the client characterized limit. The delivered clusters utilize various-widths clustering web hosting suggestion framework, where large clusters are parceled into various smaller clusters. The primary steps of this technique are condensed in the framework for coursing in algorithm 1. This method has two elements: Gathering additionally b. The past is a look at class objects, where everything contains characters similar properties of a gathering. In the fundamental development, the entire informative record is perceived as a gathering and it's driven whose width is fixed with zeros (Stage 4). The last factor is the division to the best packs size. At the next stage, the capacity biggest group restores the biggest bunch U, which isn't allocated as non-divisional, from clusters (Stage 14). On the off chance that the size regarding U is more remarkable than (or grows to) b, State (1) implies appropriation to figure a suitable width w for apportioning U. In the event when the estimation of w is equal to zero, U is assigned as non-disseminated (Stages 15-20). This occurs when the items in U have similarities regarding the separation work, and in this manner, they can't be apportioned. Otherwise, Algorithm 1 is charged into division U (Stage 21). On the off chance that the quantity of delivered clusters is only one, the estimation of w is huge and it ought to be limited by 10% and utilized once more (Stage 27). Otherwise, the new groups conveyed from U are added to bunches as opposed to U, and the greatest pack again is pulled from bunches (Stages 22-25). The processes (Stages 15-27) are repeated until the partition of the biggest bundle in Groups is less than b.

### 2.3. Algorithms

Algorithm: 1 Various-widths clustering

Input: Data;  $US_{u,1}, US_{u,2}, \dots, US_{u,M}; P_{u,1}, P_{u,2}, \dots, P_{u,M}; \varepsilon; US_1, US_2, \dots, US_N; SW_1, SW_2, \dots, SW_N; \alpha$

Output: Clusters;  $U_{u,1}, U_{u,2}, \dots, U_{u,N}$

1. Clusters ← add (Clusters; Data; zeros; 0); finished ← 0;
2. while finished == 0 do ClsSize ← Clusters: getSize /\* The product of clusters \*/
3. Partitioning (Clusters; α); Merging (Clusters; α);
4. if |LargestCluster (Clusters)| ≤ α or Clusters.getSize == ClsSize then finished ← 1
5. return [Clusters];
6. Procedure Partitioning (Clusters; α)
7. U LargestCluster (Clusters); while |U: objects| > α do w ← using eq.(1);
8. if (w == 0) then U.nonPartitioned (1); upgrade (Clusters; U);
9. resume; < tmpClusters > ← Algorithm 1(U, w);
10. if ClusterNum (tmpClusters) > 1 then remove (Clusters; U);
11. attach (Clusters; tmpClusters); U LargestCluster (Clusters);
12. else w ← w - (w \* 0.1); pass to step 21
13. Procedure Merging (Clusters; α)
14. MergingList /\* list of tuples < /\* childClusterID, parentClusterID > \*/
15. for each U in Clusters do j ← using eq. (2) and eq. (3); /\* ID of cluster contained U \*/
16. if j not equal 0 then put < U: getID; j > in MergingList;
17. while MergingList not similar to f produce
18. for each tuple in MergingList do
19. < i; j > ← tuple; if !isParent (MergingList; i) then MergeClus (Clusters; i; j);
20. delete tuple from MergingList;
21. for i = 1 to N do SWi = norm (SWi); Ssim = ∅;
22. for j = 1 to M do Si, ws = wsSimWSi, , ;
23. if Si, ws > ε and Pu, ≠ ∅ then add USu, into Ssim;
24. end if
25. end for
26. if < Num then // Num is a threshold number Find the top-10 similar users Usim that have used USC,;
27.  $P_{u,i} = \omega \frac{\sum_{US_{u,j} \in S_{sim}} S_{i,j}^{WS} \times P_{u,j}}{\sum_{US_{u,j} \in S_{sim}} S_{i,j}^{WS}} + \omega \frac{\sum_{uk \in U_{sim}} S_{u,uk}^{USER} \times P_{u,ki}}{\sum_{uk \in U_{sim}} S_{u,uk}^{USER}}$
28. else
29.  $P_{u,i} = \omega \frac{\sum_{US_{u,j} \in S_{sim}} S_{i,j}^{WS} \times P_{u,j}}{\sum_{US_{u,j} \in S_{sim}} S_{i,j}^{WS}}$

```

30. end if  $Uu_i = S_{wi} \times Pu_i$ ;
31. end for
32. return  $Uu_1, Uu_2, \dots, Uu_N$ ;

```

#### Algorithm 2: Web service graph construction

Input:  $S1_H, S2_H, \dots, SN_H$ ;  $S1_P, S2_P, \dots, SN_P$ ;  $Uu_1, Uu_2, \dots, Uu_N$ ;  $\theta_H, \theta_P, \alpha, \beta, \gamma$ ; Web service Graph  $G=(V, E)$ , parameter  $\lambda$ , adjacency matrix  $M$

Output: Web service Graph  $G=(V, E)$ ;  $M$  set a of  $b$  ranked Web services

```

1.  $V=\emptyset, E=\emptyset$ ;
2. for  $i=1$  to  $N$  do
3. if  $SiH \geq \theta_H$  or  $SiP \geq \theta_P$  then
4. add  $vi$  to  $V$ ;
5. end if
6. end for
7. or each node in  $V$  do
8.  $Score_{vi} = \alpha SiH + \beta SiP + \gamma Uu_i$ ;
9. end for
10. for each pair of nodes  $vi$  and  $vj$  in  $V$  do
11. if  $(USi, USj) \geq r$  then
12. add edge  $(vi, vj)$  to  $E$ ;
13. end if
14. end for
15. return  $G=(V, E)$ ;
16.  $a=\emptyset$ ;
17. while  $|a| \leq b$  do
18. find  $vmax = \arg \max_{v \in (V-A)} (1-\lambda)Score_v + \lambda b |Nv - N(A)|$ ;
19.  $a = a \cup \{vmax\}$ ;
20. end while
21. return  $A$ ;

```

An internet service diagram,  $G=V$ , is an undirected weighted graph comprising of a lot of nodes  $V$  and a lot of edges  $E$ , wherein a hub indicates an Internet service competitor, i.e.,  $vi=USi$ , and an edge signifies that the associated nodes are comparative.  $V=K$  is the number of nodes (i.e., Web services) that appear in the diagram. Be that as it may, here not all the internet services in the Internet service pool are utilized for developing the Internet service diagram. Just the Internet services with a specific pertinence to client interests are utilized.

#### 2.4. Clustering quality

Clustering algorithms can be difficult to determine. False-positive and false-negative decisions are penalized by external evaluation measures such as the rand index. A high number of output classes and output clusters will compromise the quality of the output. Particularly concerning is the lack of a predetermined  $K$ -value in the tested graph-based clustering algorithms. Pure measures penalize when the number of output clusters exceeds the number of class labels, so it is preferable over other evaluation measures for reducing quality trade-offs. The purity was calculated for each of the tested corpora and algorithms by

$$purity(C, M) = \frac{1}{N} \sum_k \max_j |c_k \cap m_j| \quad (2)$$

with  $N$  as the total number of documents, the set of clusters  $C$  and  $M$  as the set of classes.

### 3. RESULTS AND DISCUSSION

To perform reliable examinations, it is required to utilize huge scale true web services. To overcome the tedious job of gathering and getting ready such data, Lausen and Haselwanter [11] share an enormous scale genuine web services dataset gathered throughout their Distributed Reliability Assessment Mechanism for Web services (WS-DREAM) test. WS-DREAM exists in a web crawling motor that shakes the openly accessible Web Services Description Language (WSDL) document path of the web. It furthermore assumes non-functional traits (e.g., QoS) of these web services, considered by 640 appropriated PCs situated in 25 unique nations, from Planet-Lab4. As a result, we get the top  $k$ -web hosting list which is recommended by the system [21]-[25]. The following figures show exceptionally practical results for the proposed work clustering-based web service recommendation (CWSR). Figure 2 shows a comparison of precision of Web service Discovery (WSD) and CWSR approach and Figure 3 shows an F-score comparison between the existing systems and the proposed one.

In this test, we executed 500 experiments to evaluate the pick time of our technique. The plans change long; the range in our examination begins with plans comprising of 40 occupations to plans to contain 400 employments. Figure 4 demonstrates the determination time for the analysis of the clustering-based methodology. The time to choose web hosting for each activity inside an arrangement is somewhere in the range of 0.5 and 1.3 seconds.

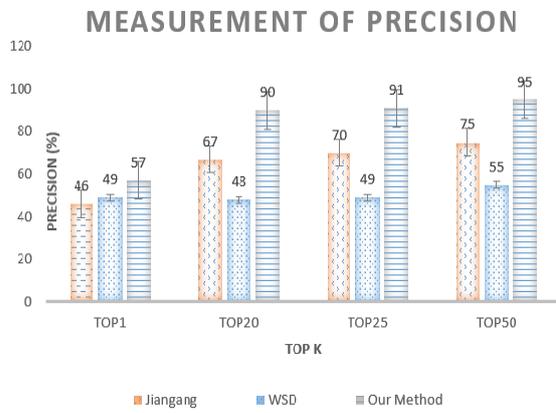


Figure 2. Measurement of precision of WSD and our method

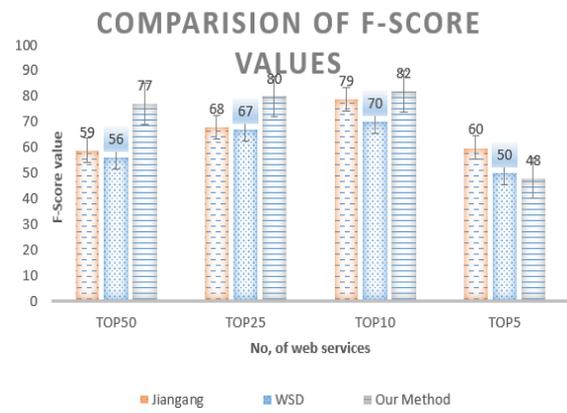


Figure 3. Comparison of F-Score values for the proposed system

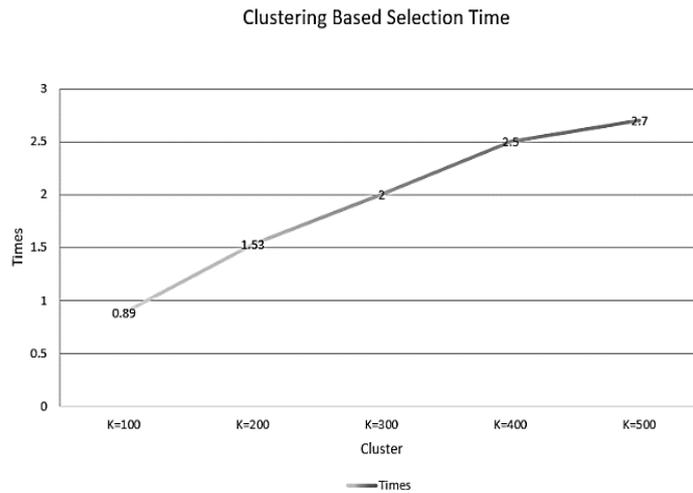


Figure 4. Selection time in clustering-based approach

#### 4. CONCLUSION

In the paper, we present an effective cluster-based novel technical solution to improve web services, by building up the exhibition of serving suggestions. We have identified the use of clustering to investigate a set of data in the real world. Data clustering can be used in the real world to investigate a set of data. Comparing novel graph-based clustering algorithms with well-known vector-based algorithms is presented here. Our study looked at how well each algorithm clustered and how it performed in general. In contrast to the graph-based clustering algorithm, the vector-based algorithms performed more efficiently. The classical approach of using the k-means algorithm, however, requires the user's intervention a priori, which eliminates use cases where the user can investigate the input data before that is applied to the clustering. However, the graph-based clustering algorithm demonstrates that a good categorization can be achieved even without requiring the pre-requisite k-value. The distance between search points is reduced by our method. Furthermore, our suggested strategy offers greater accuracy than existing systems, as shown by the results of the tests. A small number of clusters (k=6) results in a large memory cost and a long run time for the k-means algorithm. The rough clustering algorithm

improves all three criteria concerning rough clustering. As a result of the need to estimate upper and lower approximations during the process of updating a new focus, there is the creation of a high average error cost.

## ACKNOWLEDGMENT

The authors wish to express their appreciation to the Ministry of Education and Training for supporting this research project as part of the Ministerial Program of Science and Technology CTB.2021.DNA. "Research on applying deep learning model to recognize ripe pineapple period in Quang Nam-Da Nang".

## REFERENCES

- [1] H. Wang, J. Z. Huang, Y. Qu, and J. Xie, "Web services: Problems and future directions," *Journal of Web Semantics*, vol. 1, no. 3, pp. 309–320, Apr. 2004, doi: 10.1016/j.websem.2004.02.001.
- [2] M. J. Mokarrama and M. S. Arefin, "RSF: A recommendation system for farmers," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2017, pp. 843–850, doi: 10.1109/R10-HTC.2017.8289086.
- [3] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 325–341.
- [4] A. Kumar, S. Sarkar, and C. Pradhan, "Recommendation system for crop identification and pest control technique in agriculture," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2019, pp. 0185–0189, doi: 10.1109/ICCSP.2019.8698099.
- [5] C. Atkinson, P. Bostan, O. Hummel, and D. Stoll, "A practical approach to web service discovery and retrieval," in *IEEE International Conference on Web Services (ICWS 2007)*, Jul. 2007, pp. 241–248, doi: 10.1109/ICWS.2007.12.
- [6] S. K. S. Raja, R. Rishi, E. Sundaresan, and V. Srijit, "Demand based crop recommender system for farmers," in *2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, Apr. 2017, pp. 194–199, doi: 10.1109/TIAR.2017.8273714.
- [7] C. Platzter, F. Rosenberg, and S. Dustdar, "Web service clustering using multidimensional angles as proximity measures," *ACM Transactions on Internet Technology*, vol. 9, no. 3, pp. 1–26, Jul. 2009, doi: 10.1145/1552291.1552294.
- [8] N. Kokash, "A comparison of web service interface similarity measures," *Frontiers in Artificial Intelligence and Applications*, vol. 142, 2006.
- [9] K. Elgazzar, A. E. Hassan, and P. Martin, "Clustering WSDL documents to bootstrap the discovery of web services," in *2010 IEEE International Conference on Web Services*, Jul. 2010, pp. 147–154, doi: 10.1109/ICWS.2010.31.
- [10] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic web service selection," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters-WWW Alt. '04*, 2004, Art. no. 66, doi: 10.1145/1013367.1013379.
- [11] H. Lausen and T. Haselwanter, "Finding web services," *The 1st European Semantic Technology Conference*, 2007.
- [12] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval-SIGIR '07*, 2007, Art. no. 39, doi: 10.1145/1277741.1277751.
- [13] Y. Wang and J. Vassileva, "Towards trust and reputation based web service selection," in *Multi-Agent and Grid Systems (MAGS) Journal*, 2007.
- [14] B. T. G. S. Kumara, I. Paik, W. Chen, and K. H. Ryu, "Web service clustering using a hybrid term-similarity measure with ontology learning," *International Journal of Web Services Research*, vol. 11, no. 2, pp. 24–45, Apr. 2014, doi: 10.4018/ijwsr.2014040102.
- [15] E. M. Maximilien and M. P. Singh, "Multi agent system for dynamic web services selection," in *Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS)*, 2005, pp. 25–29.
- [16] E. M. Maximilien and M. P. Singh, "Reputation and endorsement for web services," *ACM SIGecom Exchanges*, vol. 3, no. 1, pp. 24–31, Dec. 2001, doi: 10.1145/844331.844335.
- [17] H. Sofian, J. T. Chia Ming, S. Muhammad, and N. Mohd Noor, "Calcification detection using convolutional neural network architectures in intravascular ultrasound images," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1313–1321, Mar. 2020, doi: 10.11591/ijeecs.v17.i3.pp1313-1321.
- [18] O. Feddaoui, R. Toufouti, L. Jamel, and S. Meziane, "Fuzzy logic control of hybrid systems including renewable energy in microgrids," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5559–5569, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5559-5569.
- [19] C. S. Baidal, N. X. Arreaga, and V. S. Padilla, "Design and testing of a dynamic reactive signage network towards fire emergency evacuations," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5853–5860, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5853-5860.
- [20] B. E. Elnaghi, M. E. Dessouki, M. N. Abd-Alwahab, and E. E. Elkholy, "Development and implementation of two-stage boost converter for single-phase inverter without transformer for PV systems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 660–669, Feb. 2020, doi: 10.11591/ijece.v10i1.pp660-669.
- [21] R. Burke, "Hybrid web recommender systems," in *The Adaptive Web*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 377–408.
- [22] H. H. C. Nguyen, A. T. Luong, T. H. Trinh, P. H. Ho, P. Meesad, and T. T. Nguyen, "Intelligent fruit recognition system using deep learning," in *Lecture Notes in Networks and Systems*, Springer International Publishing, pp. 13–22, 2021.
- [23] H. H. C. Nguyen, D. H. Nguyen, V. L. Nguyen, and T. T. Nguyen, "Smart solution to detect images in limited visibility conditions based convolutional neural networks," in *Advances in Computational Collective Intelligence*, Springer International Publishing, pp. 641–650, 2020.
- [24] Papaioannou, I., Tsesmetzis, D., Roussaki, I., and Anagnostou, M. (2006). "A QoS Ontology Language for Web-Services" In Proc. of 20th Intl. conf. on Advanced Information Networking and Applications (AINA), vol. 1, Vienna, Austria.
- [25] Y. Lai and J. Zeng, "A cross-language personalized recommendation model in digital libraries," *The Electronic Library*, vol. 31, no. 3, pp. 264–277, May 2013, doi: 10.1108/EL-08-2011-0126.

**BIOGRAPHIES OF AUTHORS**

**Ha Huy Cuong Nguyen**    obtained his doctorate in Computer Science/Resource Allocation Cloud Computing in 2017 from the University of Danang. He has published over 50 research papers. His main research interests include the resource allocation, detection, prevention, and avoidance of cloud computing and distributed systems. He serves as a technical committee program member, track chair, session chair and reviewer of many international conferences and journals. He is a guest editor of “International Journal of Information Technology Project Management (IJITPM)” with Special Issue On: Recent Works on Management and Technological Advancement. Currently, he is working at Software Development Centre, The University of Danang. He can be contacted at email: [nhhcuong@vku.udn.vn](mailto:nhhcuong@vku.udn.vn).



**Bui Thanh Khiet**    acquired his Master's degree from Posts and Telecommunications Institute of Technology in Ho Chi Minh in 2012. He is working at Institute of Engineering Technology, Thu Dau Mot University as a lecture. At present, he is a Ph.D. student at Computer Science, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), VNUHCM. His research focuses on Cloud computing. He can be contacted at email: [khietbt@tdmu.edu.vn](mailto:khietbt@tdmu.edu.vn).



**Van Loi Nguyen**    received his Master of Engineering in Computer Science from the University of Danang, Vietnam in 2010 and Ph.D. degree from Soongsil University, Korea in 2017. He is currently a lecturer at Vietnam-Korea University of Information and Communication Technology, the University of Danang. His research interests include multimedia, information retrieval, artificial intelligence, database, and IoT. He can be contacted at email: [nvloi@vku.udn.vn](mailto:nvloi@vku.udn.vn).



**Thanh Thuy Nguyen**    received the Engineer degree of the computing in 1982 and the Ph.D. degree of the computer science in 1987 from the Hanoi University of Science and Technology, Hanoi, Vietnam. He had been a Professor of computer science with the Hanoi University of Science and Technology until 2011 and since then has been with the VNU University of Engineering and Technology. Prof. Nguyen Thanh Thuy is now Head of the Key Laboratory for Artificial Intelligence (AI) at VNU UET. He is also Deputy Director of the National R&D Program KC4.0/2019-2025 (Ministry of Science and Technology). His AI research interests are mainly in Knowledge systems, Soft-computing, Data mining, Machine learning, and Hybrid Intelligent systems. He can be contacted at email: [nguyenthanhthuy@vnu.edu.vn](mailto:nguyenthanhthuy@vnu.edu.vn).