Static Hilbert convex set clustering for web services aggregation

Nawras A. Al-Musawi, Dhiah Al-Shammary
College of Computer Science and Information Technology, University of Al-Qadisiyah, Diwaniyah, Iraq

Article Info

Article history:

Received Aug 12, 2022
Revised Jun 21, 2023
Accepted Jul 2, 2023

Keywords:

Aggregation
Clustering
Hilbert
Simple object access protocol
Web services

ABSTRACT

Web services' high levels of duplicate textual structures have caused network bottlenecks and congestion. Clustering and then aggregating similar web services as one compressed message can potentially achieve network traffic reduction. In this paper, a static Hilbert clustering as new model for clustering web services based on convex set similarity is proposed. Mathematically, the proposed model calculates similarity among simple object access protocol (SOAP) messages and then cluster them based on higher similarity values. Next, each cluster is aggregated as a compact message. The experiments have explained the proposed model performance as it has outperformed the convention clustering strategies in both compression ratio and clustering time. The best results have been achievable by the proposed model has reached up to (15) with fixed-length and up to (21) with Huffman.

Corresponding Author:

Nawras A. Al-Musawi
College of Computer Science and Information Technology, University of Al-Qadisiyah
Diwaniyah, Iraq
Email: nawras.kadhim@qu.edu.iq

1. INTRODUCTION

A web service can be defined as a web application that communicates with other systems using standard internet mechanisms such as hyper text transfer protocol (HTTP) and extensible markup language (XML) in order to provide the required service to the user [1]–[3]. Web servers offer dynamic replies for all requests made via the internet [4]–[6]. Web service architecture refers to the software system design for supporting interoperable communication from machine to machine via a network [1], [7]. Web services communication is based on widely used technologies such as simple object access protocol (SOAP), web service description language (WSDL), and iniversal description discovery and integration (UDDI) [8], [9]. SOAP protocol is an XML Messaging protocol that supports encoding and decoding operations on XML-formatted data [10], [11]. SOAP is a protocol for allowing different distributed computing systems to communicate with each other [12]. Web services' technological benefits, such as interoperability and dynamic scalability, are causing organizations increasingly use them [10], [13].

Web services have acquired XML's drawbacks in creating large payload of web messages across the internet [14]. As a result, high network traffic is generated on the network. Additionally, the increased demand for web services as a means of information sharing around the world has caused congestion and bottleneck in performance [15], causing significant delay or completely halting the service. In other word, SOAP messages are inherently large, necessitating greater bandwidth for web service requests and responses [2], [16].

Web services aggregation can reduce the high network traffic by aggregating web messages along their shared network route as aggregated messages have removed data redundancy. Efficient web aggregation can be obtained for highly similar messages. Thus, similarity-based clustering and aggregation approaches have been used in a number of studies with the aim of lowering network bandwidth by delivering highly aggregated SOAP messages [17], [18]. Moreover, it would improve performance by reducing the size of the transmitted data.

In order to improve clustering approaches, a number of researches have presented different clustering methods for textual and XML documents. Some of these methods have concentrated on measuring distance for xml trees based on utilizing XML documents structural similarities. On the other hand, other methods take into account the xml documents content.

Flesca *et al.* [19] has addressed structural similarity detection problem between XML texts. New method has been introduced based on similarity for xml documents structures in a time series created. The suggested technique's fundamental idea is to transform the XML document to numeric form after linearizing the XML document's structure by encoding XML tags into signal pulses. Then, based on the numerical sequences analysis of their, XML documents are divided into clusters. The discrete Fourier transform is applied for comparing the encoded xml texts efficiently in frequency domain. Real and synthesized dataset have been used for experiment and evaluation. The experiments have shown the proposed model outperformed others.

Principal component analysis (PCA) is applied by Liu *et al.* [20] in order to create a novel XML clustering strategy. The suggested technique initially extracts features from documents that represented by ordered and labeled trees, then convert the extracted features into vectors based on features frequencies in documents. PCA has been used to minimize number of features in vectors and clustered them by K-means algorithm. Two datasets of XML documents have been used for testing and analysis. Datasets are obtained from Wisconsin's XML data bank. The experiments have shown improvement of clustering accuracy with PCA technique.

Yongming *et al.* [21] has introduced the extended vector space model (EVSM) as a new model for clustering XML documents based on structure and content by computing similarities between them. Moreover, this strategy has applied hierarchical clustering technique. Two metrics have been implemented in order to evaluate the proposed clustering model. These metrics include purity and entropy. Experimentally, two datasets (INEX IEEE corpus, Wikipedia collection) have been applied for testing their model. The results have indicated that this strategy would improve clustering efficiency.

This paper proposed Hilbert space and measurements as the based for static clustering for web messages to provide technical support for web services aggregation. The evaluation strategy for the proposed model is completely based on certain metrics like compression ratio, processing time and compressed size file. In this paper, Hilbert clustering model has been designed for clustering SOAP messages and then applied aggregator model that has developed by [15], [22]. The aggregator model technically aggregates web messages that have been clustered by the proposed model and both Jaccard [23] and vector space [24]. Evidently, experiments have shown that web aggregator can reduce messages size more effectively when using Hilbert clustering in comparison to other methods. Furthermore, compression ratio for different cluster size ranging from 2 to 10 has been computed and compared with other clustering techniques. Additionally, processing time of the proposed model has been examined and it has shown to be substantially faster than other models. Finally, it is compared to other research in the field that applied the same SOAP messages dataset.

The reset of this paper is organized as the follows: section 2 explains the proposed method. Section 3 shows the Hilbert convex set as similarity measurements. Section 4 illustrates web service aggregation. The proposed model is illustrated in section 5. Section 6 discusses the experiments and results. Finally, section 7 presents the conclusion and future work.

2. PROPOSED METHOD

Technically, the proposed model has provided several major contributions in regard to the web service performance over the network. Firstly, it would improve web based communication over network by significantly reducing the required network volume. This fact would solve the network congestion and bottleneck significantly. Secondly, an efficient web messages clustering has been achieved based on Hilbert similarity measurements. Finally, the proposed model would potentially improve web service response time as a result of the required low processing time.

3. HILBERT CONVEX SET AS SIMILARITY MEASUREMENTS

In this research, we have proposed Hilbert based on convex set measurements for SOAP messages clustering. We have considered each message as convex (set) and each feature in message as point. Technically, convex set measurements are applied to calculate similarity between two messages. As shown in (1) illustrates the Hilbert convex set measurements including their parameters:

$$Sim\ (x, y) = \frac{\sum_{i=1}^{n}(X_i * Y_i) - Max\ (x)}{\|x\| * \|y\|} \tag{1}$$

$$\|x\| = \sqrt{x_1^2 + x_2^2 \ + x_3^2 + \cdots} \quad (2)$$

$$\|y\| = \sqrt{y_1^2 + y_2^2 \ + y_3^2 + \cdots} \quad (3)$$

where:
- X and Y are vectors (X and Y are represented by SOAP messages).
- n is number of attributes in vector.
- Max (X) is maximum value in vector X.

4. WEB SERVICE AGGREGATION

Aggregation model based on compression [15], [22] is applied in this paper to evaluate the performance of the proposed model. Aggregation model has included three stages leading to a compact aggregated message. Firstly, XML messages are converted into web matrix (matrix form) and then represented by XML tree. Secondly, XML trees are converted into vectors by either traverse techniques depth or breadth first tree traverse. Finally, these text expressions are encoded by either fixed-length or Huffman lossless compression techniques. They have shaped the proposed model into two main approaches named one-bit and tow-bit aggregators. Both aggregators provide two versions for fixed-length and Huffman encoding methods.

5. STATIC HILBERT CLUSTERING FOR WEB SERVICE

A novel Hilbert clustering technique is proposed to provide static clustering for web service. Hilbert space and distance measurements are introduced as a new mathematical computations for similarity between XML messages. The proposed model first converts web messages into numeric form developed by [15], [25]. The XML message conversion into numeric form starts by building XML tree structure and matrix form then into unified vectors and applying term frequency–inverse document frequency (TD-IDF) to generate the final numeric representations. Next, Hilbert convex set measurements are achieved on the numeric representations of XML messages. Technically, Hilbert similarity measurements are applied in order cluster XML messages based on the maximum similarity values. The final Hilbert clusters would be the input to the We aggregators to aggregate each cluster messages together. Figure 1 explains the main steps of proposed Hilbert clustering model. Figure 2 explain the proposed model in details.

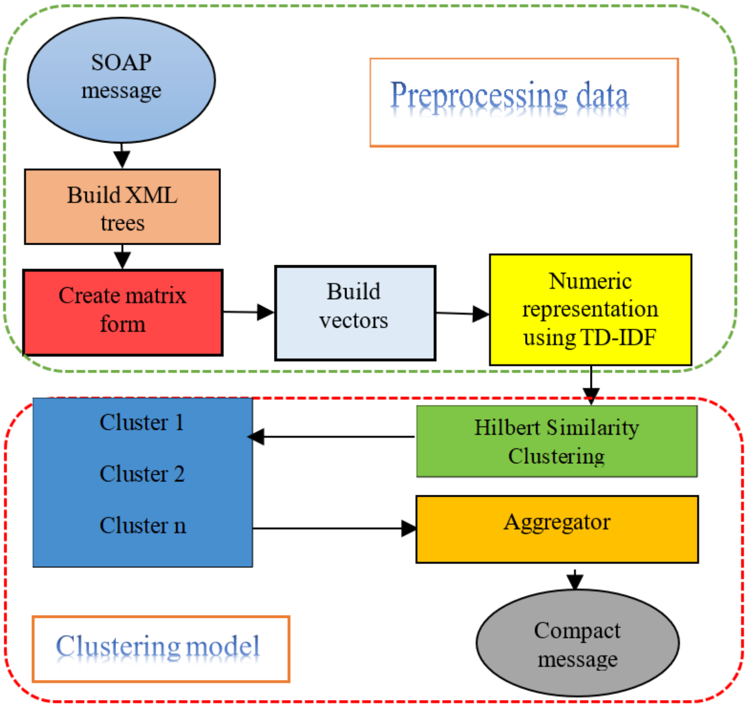


Figure 1. Main steps of Hilbert clustering

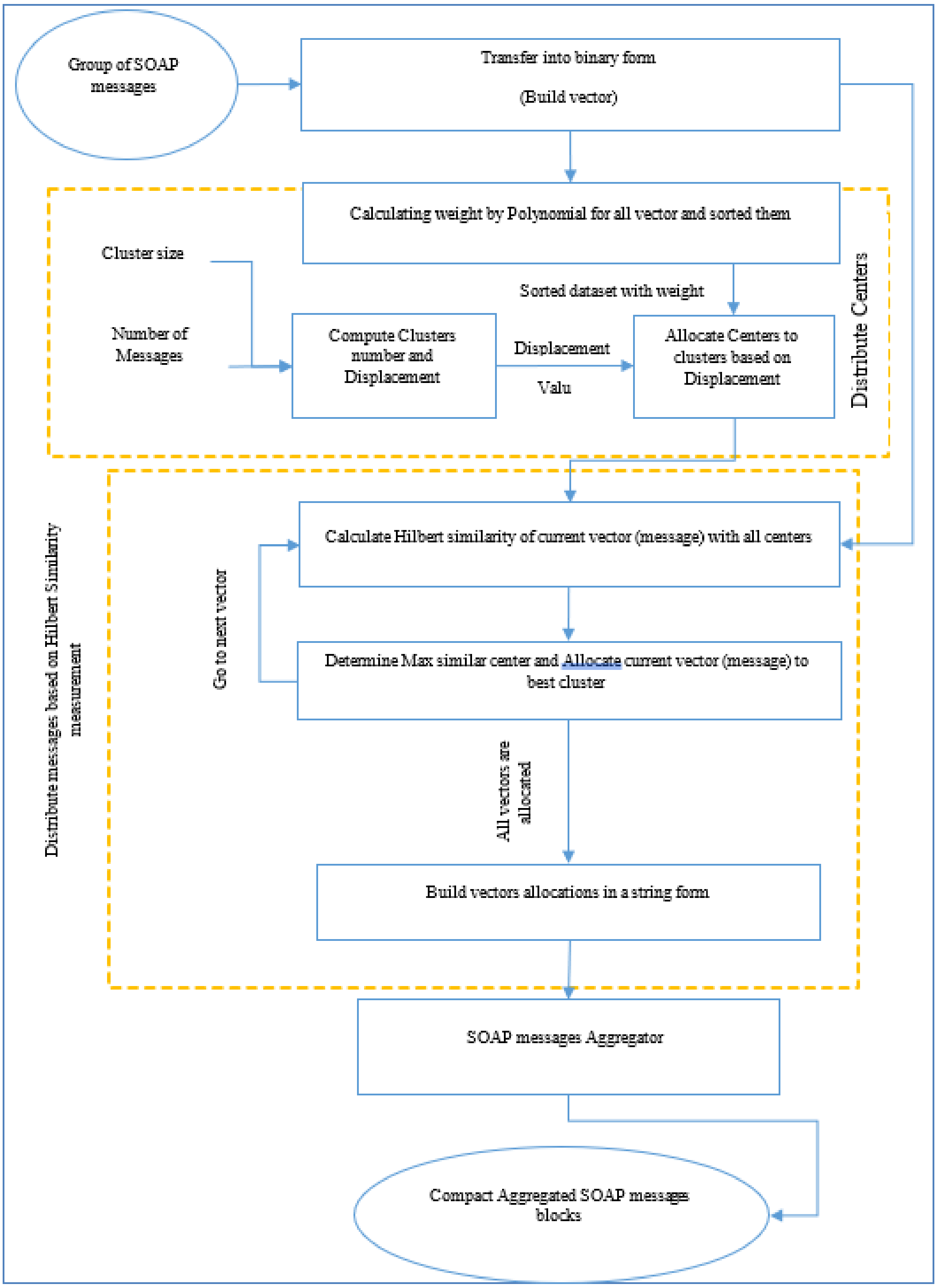

Group of SOAP messages
Transfer into binary form
(Build vector)
Calculating weight by Polynomial for all vector and sorted them
Cluster size
Number of Messages
Sorted dataset with weight
Compute Clusters number and Displacement
Displacement
Valu
Allocate Centers to clusters based on Displacement
Distribute Centers
Distribute messages based on Hilbert Similarity measurement
Go to next vector
Calculate Hilbert similarity of current vector (message) with all centers
Determine Max similar center and Allocate current vector (message) to best cluster
All vectors are allocated
Build vectors allocations in a string form
SOAP messages Aggregator
Compact Aggregated SOAP message blocks

Figure 2. Shown the main function of static Hilbert clustering model

Technically, our model starts with transform SOAP messages into numeric form and generate vectors matrix. Then, the proposed model computes vector weight by using polynomial shown in (4) and sorting them in ascending order to determine the centers points. Then the displacement is calculated between centers in order to achieve sufficient spacing and thus obtain an efficient clustering. Next, the remaining XML messages are allocated based on Hilbert convex measurements. Hilbert similarity measurements are calculated for each (current) message with all centers and compared them then allocate message to cluster with maximum similarity value. Finally, it is built the resultant clusters in a string form and put them into the aggregator tool that aggregate each cluster of similar messages into one compact message. Algorithm 1 has explained the process of messages distribution based on Hilbert convex set measurements.

$$W = \sum_{k=1}^{N} \mathrm{k} * x_k \quad (4)$$

Where:
- N: number of vector items.
- X: vector items.

Algorithm 1. Hilbert clustering (messages distribution)

```
 // Definitions
Sn // Number of message in dataset
Cn // Number of clusters
Vs // Vector items
Vflag (Sn) // Array of flag to determine the allocated centers


For i= 1 to Sn // compute sum weight and find norm for all messages
 Norm (i) = $\sqrt{\sum_{f=1}^{N} X_f^2}$
End for
For i= 1 to Vs // All vector items
 // find Maximum value (Max)
End for
For i=1 to Sn // Compute Hilbert similarity
 If Vflag (i) then
 For j = 1 to Cn // All centers

 Dot = $\sum_{r=1}^{N} [Wj(r) \times Wi(r)]$

 HS = $\frac{Dot-Max(i)}{Norm(i)\times Norm(j)}$
 // Find maximum similarity (closest message)
 End for
 // Add message to the cluster
 // Exclude message
 End if
End for
For i= 1 to Cn
 // generate string form for all allocation vectors (clusters)
End for
```

6. EXPERIMENTS AND RESULTS

In order to evaluate the performance of the proposed Hilbert clustering model, a wide range of XML message sizes have been considered in the experimental analysis. The dataset was formed using the WSDL (web service description language) at [26]. It contains 160 messages divided into four groups based on their size. Which categories: small, medium, large, very large, with sizes ranging from 140 to 5,500 bytes. Each group has 40 messages [15]. We have used the compression-based aggregation model [22] as a tool to investigate the efficiency of the proposed model compared to other standard clustering methods. Both vector space model and Jaccard techniques have been applied on the same dataset and compared with the proposed model. The main evaluation metrics include compression ratio, clustering time and compressed size file that are computed by aggregator tool on the resultant clusters. The highest CR achieved by these methods means that this is the best clustering. Furthermore, other comparisons are applied with K-means and PCA+K-means from previous studies [15], [20].

In order to achieve potentially large compression ratios on the clustered XML messages, all strategies demonstrated substantial results. Table 1 illustrates the average compression ratio results for Hilbert clustering based on convex compared to VSM and Jaccard with 40 messages size for all groups. The best achieved results of CR for proposed model started from (2.3818) for small messages to (21,5178) for very large messages with Huffman technique. Moreover, CR increases with the size of the message. On the other hand, Table 2 explains the average clustering time results of these techniques for all groups with 40 message size. Evidently, the proposed model has stated less processing time for small to very large SOAP messages than others. Table 3 presents the resultant average CR and clustering time for the proposed model compared with K-means, PCA combined with K-means, vector space and Jaccard clustering techniques for all messages groups. It has shown the performance of Hilbert clustering model as better than other models in terms of CR except Jaccard method. However, it is better than Jaccard in terms of processing time.

Several illustrative figures have been computed in order to show the outcome in a clear manner. Figures 3-6 depicts the ability of minimizing the overall size of the aggregated messages with variety original sizes for 10 messages in both fixed-length (Figure 3(a)) and Huffman (Figure 3(b)) encoding, 20 messages in both fixed-length (Figure 4(a)) and Huffman (Figure 4(b)) encoding, 30 messages in both fixed-length (Figure 5(a)) and Huffman (Figure 5(b)) encoding, and 40 messages in both fixed-length (Figure 6(a)) and Huffman (Figure 6(b)) encoding. It is clear that, all clustering methods have the same performance in compressed size of cluster size (10) for each group.

Table 1. Average compression ratio of VSM, Jaccard and Hilbert clustering models for small, medium, large and very large groups with 40 messages in both fixed-length and variable length (Huffman) compression techniques

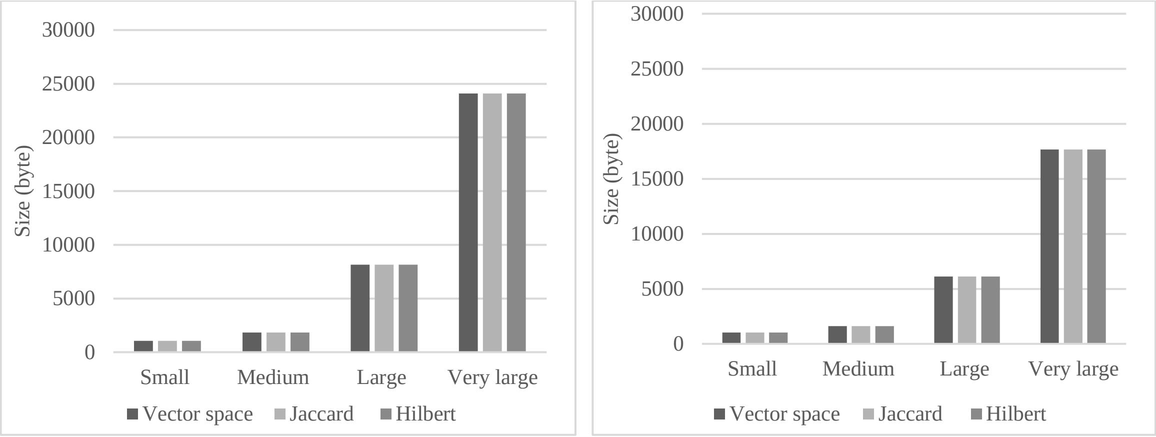
Message size	Cluster size	Average compression ratio					
		Fixed-Length			Huffman		
		Vector space	Jaccard	Hilbert	Vector space	Jaccard	Hilbert
40 small messages	2	2.6817	2.7312	2.5275	2.5412	2.5859	2.3818
	3	2.9393	3.0216	2.8206	2.8070	2.8837	2.6907
	4	3.1780	3.3830	3.0514	3.0514	3.2525	2.9229
	5	3.4557	3.6333	3.4909	3.3214	3.5196	3.3586
	6	3.6542	3.7918	3.7436	3.5502	3.6750	3.6262
	7	3.8593	4.0062	3.8747	3.7411	3.9004	3.7798
	8	4.0152	4.1651	4.2027	3.9736	4.1311	4.1125
	9	4.3798	4.2864	4.1909	4.3550	4.2593	4.1418
	10	4.4772	4.6874	4.6473	4.5388	4.6711	4.6146
40 medium messages	2	4.3825	4.5410	4.3479	4.3588	4.4851	4.3133
	3	5.2813	5.4425	5.2720	5.4162	5.6039	5.4188
	4	6.2065	6.3297	6.2094	6.5076	6.6514	6.5102
	5	7.0880	7.0736	7.0228	7.5759	7.5639	7.4844
	6	7.5078	7.5658	7.5611	8.1419	8.1980	8.1405
	7	8.1134	8.1336	8.1562	8.8889	8.8918	8.9054
	8	8.8320	8.8329	8.8241	9.8971	9.7701	9.8607
	9	8.9571	9.0239	9.0013	10.0083	10.1656	10.1373
	10	9.6994	9.6857	9.6995	10.9465	11.0132	11.0931
40 large messages	2	10.2730	10.4167	10.3753	11.9534	12.1170	12.0951
	3	11.6299	11.7641	11.7829	13.9755	14.2386	14.2157
	4	12.6629	12.7052	12.7444	15.6154	15.6911	15.7280
	5	13.3683	13.2153	13.3090	16.7904	16.5509	16.6104
	6	13.6398	13.5308	13.6579	17.2260	17.1477	17.4341
	7	14.0068	13.8217	13.9644	17.9608	17.6124	17.6663
	8	14.2733	14.2038	14.3486	18.2231	18.4295	18.4507
	9	14.3288	14.1396	14.3027	18.6047	18.0002	18.3985
	10	14.6920	14.5253	14.6729	19.0743	18.8139	19.0686
40 very large messages	2	13.9401	13.9677	13.9215	17.7889	17.8508	17.7472
	3	14.6335	14.6252	14.5865	18.9377	18.9502	18.9830
	4	15.0704	15.0632	15.0602	19.8853	19.8691	19.8445
	5	15.2933	15.3079	15.2925	20.4160	20.4445	20.3609
	6	15.4299	15.4229	15.4195	20.5136	20.6768	20.6945
	7	15.5468	15.5483	15.5386	20.7741	20.9945	20.9160
	8	15.6701	15.6720	15.6610	20.9646	20.9946	21.1399
	9	15.6832	15.6729	15.6649	20.9584	21.0574	20.7996
	10	15.7956	15.8018	15.7998	21.2841	21.4861	21.5178

Table 2. Average clustering time (ms) of VSM, Jaccard and Hilbert clustering models for small, medium, large and very large groups with 40 messages in both fixed-length and variable length (Huffman) compression techniques

Message group	Average clustering time (ms)					
	Fixed-Length			Huffman		
	Vector space	Jaccard	Hilbert	Vector space	Jaccard	Hilbert
40 small messages	2.3333	2.5555	2.3333	2.4444	5.2222	1.8888
40 medium messages	18.7777	37	15.7777	21.4444	36.5555	14.1111
40 large messages	78.1111	197.2222	76.8888	86	256.5555	83.3333
40 very large messages	625	1112.8888	653.4444	608.5555	1081.4444	604.7777

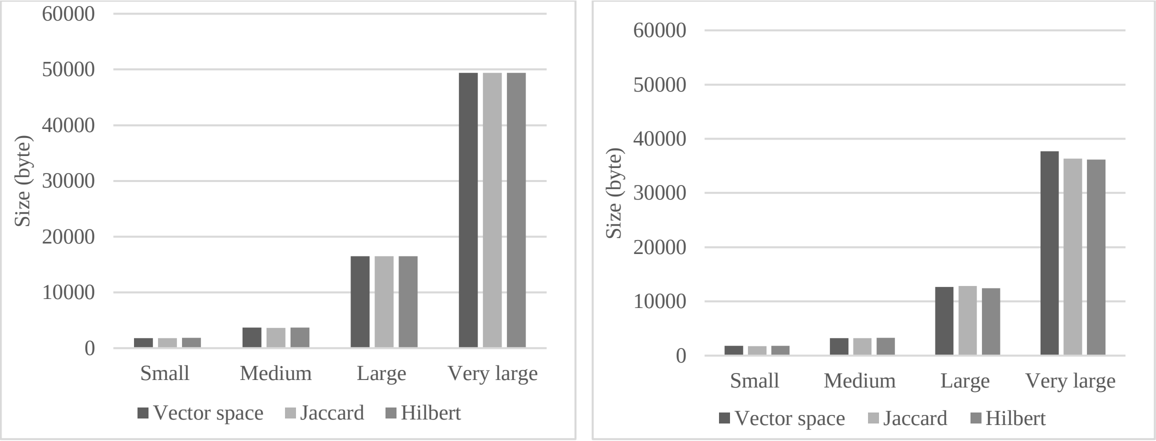
Table 3. Overall average compression ratio of K-means, PCA+K-means, Hilbert, vector space and Jaccard for small, medium, large and very large groups with 40 messages in both fixed-length and variable length (Huffman) compression techniques

Parameter	K-means	K-means + PCA	Hilbert	Vector space	Jaccard
		40 small messages			
Fixed-length average cr.	3.920295	3.850353	3.616667	3.626756	3.745148
Huffman average cr.	3.820822	3.7136433	3.514367	3.542261	3.653239
A.V clustering time (ms)	50.8831	65.3333	2.11105	2.38885	3.88885
		40 medium messages			
Fixed-length average cr.	6.766314	6.797503	7.347333	7.340936	7.403237
Huffman average cr.	7.715699	7.843987	7.984901	7.971288	8.038162
A.V clustering time (ms)	52.3342	62.8888	14.9444	20.11105	36.77775
		40 large messages			
Fixed-length average cr.	12.943645	12.815021	13.239845	13.208350	13.146983
Huffman average cr.	16.020012	16.279294	16.629769	16.602675	16.511307
A.V clustering time (ms)	54	68.1111	80.11105	82.05555	226.88885
		40 very large messages			
Fixed-length average cr.	15.109293	15.127478	15.216113	15.229273	15.231377
Huffman average cr.	20.163554	20.253857	20.222642	20.169220	20.258264
A.V clustering time (ms)	53.6231	70.4444	629.11105	616.7777	1097.1666



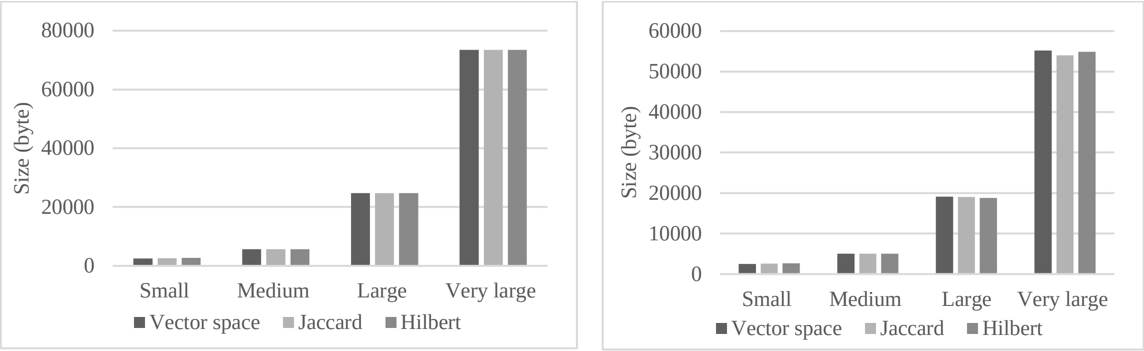
(a) (b)

Figure 3. Compressed size for 10 messages with original sizes (4672, 17088, 121010, 383350) for small, medium, large, and very large respectively in both (a) fixed-length and (b) Huffman techniques



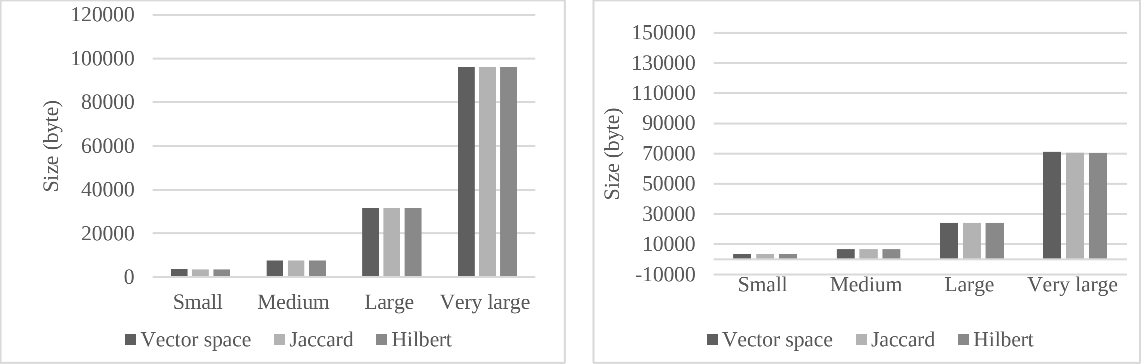
(a) (b)

Figure 4. Compressed size for 20 messages with original sizes (8566, 34241, 245413, 784034) for small, medium, large, and very large respectively in both (a) fixed-length and (b) Huffman techniques



(a) (b)

Figure 5. Compressed size for 30 messages with original sizes (12,146, 55,053, 366,157, 1,516,448) for small, medium, large, and very large respectively in both (a) fixed-length and (b) Huffman techniques



(a) (b)

Figure 6. Compressed size for 40 messages with original sizes (16,474, 74,310, 465,015, 1,516,448) for small, medium, large, and very large respectively in both (a) fixed-length and (b) Huffman techniques

7. CONCLUSION

In conclusion, this paper has introduced a new Hilbert clustering model based on convex set. The main technique is based on computing the similarity values between web service messages and clustering them. The experimental results have shown the effectiveness of the proposed model in terms of enabling the aggregation approach to dramatically reduce network traffic in comparison to other traditional methods. The proposed model is capable of achieving high compression ratio that reached up to 21 with Huffman technique. Moreover, it has provided the considerable reduction in processing time. In the future, we would apply the proposed model on large dataset and using Hilbert similarity measurements for dynamic clustering.

REFERENCES

[1] B. Natarajan, M. S. Obaidat, B. Sadoun, R. Manoharan, S. Ramachandran, and N. Velusamy, "New Clustering-based semantic service selection and user preferential model," *IEEE Systems Journal*, vol. 15, no. 4, pp. 4980–4988, Dec. 2021, doi: 10.1109/JSYST.2020.3025407.

[2] S. Pawar, V. Nayak, G. Laxmi, and N. N. Chiplunkar, "A novel web service search system using WSDL," in *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, Aug. 2020, pp. 1252–1257, doi: 10.1109/ICSSIT48917.2020.9214201.

[3] R. Elfwing, U. Paulsson, and L. Lundberg, "Performance of SOAP in web service environment compared to CORBA," in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 2002, pp. 84–93, doi: 10.1109/APSEC.2002.1182978.

[4] M. Kuehnhausen and V. S. Frost, "Application of the Java message service in mobile monitoring environments," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1707–1716, Sep. 2011, doi: 10.1016/j.jnca.2011.06.003.

[5] V. Diamadopoulou, C. Makris, Y. Panagis, and E. Sakkopoulos, "Techniques to support web service selection and consumption with QoS characteristics," *Journal of Network and Computer Applications*, vol. 31, no. 2, pp. 108–130, Apr. 2008, doi: 10.1016/j.jnca.2006.03.002.

[6] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, Jan. 2011, doi: 10.1016/j.jnca.2010.07.006.

[7] J. Puttonen, A. Lobov, M. A. C. Soto, and J. L. M. Lastra, "Cloud computing as a facilitator for web service composition in factory automation," *Journal of Intelligent Manufacturing*, vol. 30, no. 2, pp. 687–700, Feb. 2019, doi: 10.1007/s10845-016-1277-z.

[8] K. Zeng and I. Paik, “Semantic service clustering with lightweight BERT-based service embedding using invocation sequences,” *IEEE Access*, vol. 9, pp. 54298–54309, 2021, doi: 10.1109/ACCESS.2021.3069509.

[9] M. Crasso, A. Zunino, and M. Campo, “Research review: A survey of approaches to Web Service discovery in service-oriented architectures,” *Journal of Database Management*, vol. 22, no. 1, pp. 102–132, 2011, doi: 10.4018/jdm.2011010105.

[10] D. Al-Shammary, I. Khalil, and Z. Tari, “A distributed aggregation and fast fractal clustering approach for SOAP traffic,” *Journal of Network and Computer Applications*, vol. 41, no. 1, pp. 1–14, May 2014, doi: 10.1016/j.jnca.2013.10.001.

[11] D. Andresen, D. Sexton, K. Devaram, and V. P. Ranganath, “LYE: a high-performance caching SOAP implementation,” in *Proceedings of the International Conference on Parallel Processing*, 2004, pp. 143–150, doi: 10.1109/icpp.2004.1327914.

[12] M. M. Hassan and H. O. Abdullahi, “Clustering simple object access protocol (SOAP) web services: a review,” *International Journal of Scientific and Engineering Research*, vol. 9, no. 11, pp. 1493–1496, 2018.

[13] L. Purohit and S. Kumar, “Clustering based approach for web service selection using skyline computations,” in *Proceedings - 2019 IEEE International Conference on Web Services, ICWS 2019 - Part of the 2019 IEEE World Congress on Services*, Jul. 2019, pp. 260–264, doi: 10.1109/ICWS.2019.00052.

[14] C. Werner, C. Buschmann, and S. Fischer, “Compressing SOAP messages by using differential encoding,” in *Proceedings - IEEE International Conference on Web Services*, 2004, pp. 540–547, doi: 10.1109/ICWS.2004.1314780.

[15] D. Al-Shammary, I. Khalil, Z. Tari, and A. Y. Zomaya, “Fractal self-similarity measurements based clustering technique for SOAP Web messages,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 5, pp. 664–676, May 2013, doi: 10.1016/j.jpdc.2013.01.005.

[16] C. Bo *et al.*, “Development of web-telecom based hybrid services orchestration and execution middleware over convergence networks,” *Journal of Network and Computer Applications*, vol. 33, no. 5, pp. 620–630, Sep. 2010, doi: 10.1016/j.jnca.2010.03.025.

[17] D. Al-Shammary and I. Khalil, “Dynamic fractal clustering technique for SOAP web messages,” in *Proceedings - 2011 IEEE International Conference on Services Computing, SCC 2011*, Jul. 2011, pp. 96–103, doi: 10.1109/SCC.2011.15.

[18] N. G. Rezk, A. Sarhan, and A. Algergawy, “Clustering of XML documents based on structure and aggregated content,” in *Proceedings of 2016 11th International Conference on Computer Engineering and Systems, ICCES 2016*, Dec. 2017, pp. 93–102, doi: 10.1109/ICCES.2016.7821981.

[19] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese, “Fast detection of XML structural similarity,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 160–175, Feb. 2005, doi: 10.1109/TKDE.2005.27.

[20] J. Liu, J. T. L. Wang, W. Hsu, and K. G. Herbert, “XML clustering by principal component analysis,” in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2004, pp. 658–662, doi: 10.1109/ICTAI.2004.122.

[21] G. Yongming, C. Dehua, and L. Jiajin, “Clustering XML documents by combining content and structure,” in *2008 International Symposium on Information Science and Engineering, ISISE 2008*, Dec. 2008, vol. 1, pp. 583–587, doi: 10.1109/ISISE.2008.301.

[22] D. Al-Shammary and I. Khalil, “Compression-based aggregation model for medical web services,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC’10*, Aug. 2010, pp. 6174–6177, doi: 10.1109/IEMBS.2010.5627759.

[23] Y. Wang and Y. H. Li, “Deep web entity identification method based on improved Jaccard coefficients,” in *ICRCCS 2009 - 2009 International Conference on Research Challenges in Computer Science*, Dec. 2009, pp. 112–115, doi: 10.1109/ICRCCS.2009.36.

[24] H. Liu, H. Bao, J. Wang, and D. Xu, “A novel vector space model for tree based concept similarity measurement,” in *ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering*, 2010, vol. 2, pp. 144–148, doi: 10.1109/ICIME.2010.5477749.

[25] J. H. Hwang and M. S. Gu, “Clustering XML documents based on the weight of frequent structures,” in *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, Nov. 2007, pp. 845–849, doi: 10.1109/ICCIT.2007.101.

[26] “Standards - W3C,” W3C, 2012, Accessed: Jun. 18, 2013. [Online]. Available: http://www.w3.org/standards.

BIOGRAPHIES OF AUTHORS

Nawras A. Al-Musawi received her M.Sc. degree in Computer Science from the College of Computer Science and Information Technology at the University of Al-Qadisiyah-Iraq in 2022. She has obtained a bachelor degree in Computer Science from the College of Science at the University of Al-Qadisiyah-Iraq in 2009. She has worked in teaching of computer science courses for high school students for several years. Nawras has the interest in research on web services, data compression, clustering techniques and encoding methods. She can be contacted at email: nawras.kadhim@qu.edu.iq.

Dhiah Al-Shammary has received his Ph.D. in Computer Science in 2014 from RMIT University, Melbourne, Australia. He is awarded as the best Ph.D. student and top publication during his Ph.D. period. He has several years’ experience in both education and industry. His main industrial experience came from Silicon Valley based companies working on security projects including non-traditional and quantum scale encryption. He has worked at several universities in both Australia and Iraq like RMIT University and University of Al-Qadisiyah. His research interests include performance modelling, web services, compression and encoding techniques, and distributed systems. He has several publications in the areas of improving the performance of web services and encoding techniques. He can be contacted at email: d.alshammary@qu.edu.iq.