Journal of Theoretical and Applied Information Technology
15th February 2018. Vol.96. No 3
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ANALYSIS OF PARALLELIZATION IN MAMDANI- AND SUGENO-TYPE QUALITY OF WEB SERVICE FUZZY MONITORING MODELS

MOHD HILMI HASAN, EMELIA AKASHAH PATAH AKHIR, NORSHAKIRAH A AZIZ, IZZATDIN ABDUL AZIZ, JAFREEZAL JAAFAR

Universiti Teknologi PETRONAS, Computer & Information Sciences Department, 32610 Seri Iskandar, Perak, Malaysia

E-mail: {mhilmi_hasan, emelia.akhir, norshakirah.aziz, izzatdin, jafreez}@utp.edu.my

ABSTRACT

Quality of web service (QoWS) monitoring is an important component in web service as it evaluates web service delivery performance and detects problems. Our previous work proposed a fuzzy model for QoWS monitoring due to uncertain nature of web service environment. However, fuzzy models are computationally costly. In this work, we propose a parallelization implementation of the models. The objective of this paper is to compare the performance between Mamdani- and Sugeno-based fuzzy inference systems (FIS) when they are applied to the QoWS monitoring models. The results suggested that Sugeno models produced less processing time than that of Mamdani models. However, Mamdani models benefited from parallelization more than that of Sugeno models by recoding higher percentage of improvement in terms of average processing time. This work will be expanded to investigate the implementation of the models in cluster computers and using a higher type of fuzzy logic, namely interval type-2 fuzzy.

Keywords: *Fuzzy Inference System, Fuzzy Parallelization, Sugeno, Mamdani, Web Service Monitoring*

1. INTRODUCTION

In web services environment, quality of web service (QoWS) monitoring has become one of the crucial procedures. This is because QoWS monitoring is related to non-functional aspects, hence it determines how good a service is in delivering its tasks. QoWS monitoring is also used to identify the existence of problems in web service environment. Moreover, QoWS monitoring results can also be used by existing users to decide whether or not extending their service subscription. This is because the monitoring results are used to compare with expected performance specified in service level agreement (SLA).

Existing QoWS monitoring models are based on exact computation. This means that, the decision is made based on hard computation. In contrast, our previous work proposed this QoWS monitoring to be carried out vaguely using fuzzy logic. The main motivation of the work was that web service environment contains high degree of uncertainty, hence its QoWS monitoring can be better performed using fuzzy logic. This capability of managing uncertainty by fuzzy logic was shown in previous works such as traffic regulations in wireless mesh networks [1], processing human perception [2], and network traffic policy [3].

In our preliminary experiment, we clustered a dataset of one type of QoWS parameter, namely availability, based on non-fuzzy algorithm (KMeans). We divided the dataset into three clusters; Good, Moderate and Poor. Table 1 shows the results, which indicate that the borders of Good-Moderate and Moderate-Poor clusters for the original dataset are 78.36% and 48.52%, respectively. Then, we imposed this original dataset with random errors. These random errors represent uncertainties that occur in the QoWS data. We named these datasets with uncertainties as synthetic datasets. We applied two error values; +-0.5% and +-10%, hence two synthetic datasets were constructed. As shown in Table 1, the border values of both synthetic models are different from the original dataset. That means, the level of QoWS cannot be determined precisely due to the occurrence of uncertainties. For example, assuming there is an availability value of 78.00%. This value is evaluated as Moderate if it is based on the original dataset. However, it is evaluated as Good if both synthetic datasets are used.

Table 1: Non-fuzzy clustering of availability datasets under uncertainies

Dataset	Good-Moderate Cluster (%)	Moderate-Poor Cluster (%)
Original	78.36	48.52
Synthetic +-0.5% error	77.64	47.74
Synthetic +-10% error	77.43	49.67

Apart from our previous work, a number of other works have also proposed fuzzy implementation in web service system. Mobedpour & Ding implemented fuzzy logic in QoWS-based web services selection system [4]. In their system, a user may select web service by specifying his/her expected QoWS values using linguistic variables such as “good”, “medium”, or “poor”. Bacciu, Buscemi, & Mkrtchyan also proposed fuzzy implementation in web service selection system [5]. Their system requires a user to input exact values to be matched with QoWS offered by providers. The matching procedure is carried out through fuzzy logic. Allenotor & Thulasiram applied fuzzy logic in QoS requirements framework for grid computing [6]. The framework was able to avoid over-committing of grid resources because fuzzy logic has the capability of handling the imprecision of users’ QoS requirements. Sherchan, Loke, & Krishnaswamy implemented fuzzy logic in web service reputation system [7]. Fuzzy logic was proposed because reputation system involves objective and subjective users’ ratings. The results showed that the proposed implementation could effectively manage these objective and subjective ratings.

However, fuzzy logic uses a more complex algorithm than that of exact computation. As a result, the implementation of fuzzy-based systems requires high computational time. This may prevent fuzzy logic from being widely adopted, despite its great advantages at managing uncertainty [8]. This is the main motivation of our research. We propose the implementation of fuzzy-based QoWS monitoring model using parallel computation, with the aim to reduce its computational time.

Chantrapornchai & Pipatpaisan argued that parallelization is required for fuzzy systems to reduce their computational time [9]. Fuzzy systems are computational costly due to their many computation steps and cycles. In the work, parallelization of fuzzy systems was performed using openMP through fine-grained and coarse-grained techniques. The work concluded that fuzzy systems may achieve better computation time by continuously running using openMP. Nguyen et al. produced mathematical equations for the possibility of parallelizing uncertainty computation such as in fuzzy systems [10]. The work proposed that whenever there are a number of processors available for computing uncertainty problems, parallelization should be implemented so that the computational time can be reduced. Bharathi et al. presented the parallelization of Fuzzy C-Means (FCM) clustering [11]. FCM is normally used in the development of data analytics tools that are based on data. The work argued that descriptive analytics tools require full data set, running on big data environment. The parallelization reduced the time taken for conducting the FCM clustering of the used data set.

Hence, in this work, we implemented parallelization in two types of QoWS monitoring model; Mamdani- and Sugeno- type. Improvement in their computational time are measured and compared between each other. To the best of our knowledge, there is no paper in the literature that compared parallelized Mamdani and Sugeno fuzzy models for QoWS monitoring. The main contribution of this work lies in the analysis of whether or not significant improvement can be gained from the parallelization of the models, and which implementation produced best results in terms of computational time. Therefore, the objective of this paper is to present the results of this comparative analysis.

This paper is organized as follows. The next section presents the materials and methods. Results of the work are presented in section 3. Discussion and conclusion are included in section 4.

2. MATERIALS AND METHOD

2.1 QoWS Data

In this work, we utilized the QoWS datasets provided by Al-Masri and Mahmoud (2007) [12]. The datasets contain various real data captured by their crawler. For the purpose of our experiments, three QoWS datasets were selected, namely availability (in % unit), response time (in millisecond (ms) unit), and latency (in ms unit). Each of these three datasets contained 1500 data points. We named these datasets as original datasets.

2.2 Fuzzy Inference Systems (FIS)

An FIS is best described by overviewing the general concept of fuzzy logic. Fuzzy logic is a soft computing technique that processes input into output through its if-else rules over fuzzy sets. The main strength of fuzzy logic is that it allows inferencing process to be done with uncertainty. This means that, fuzzy logic assigns each input with a membership degree of between 0 to 1 values to each cluster in fuzzy sets. In contrast, classical set that works based on crisp computation, assigns each input with either true or false membership to a cluster.

Assuming X is the universe of objects, and x is its element, $x \in X$, and there is a set C, where $C \subseteq X$. In crisp logic, each x, $x \in X$, can either belongs to set C or not; (x,0) or (x,1). In contrast, fuzzy logic assigns x with a membership degree of [0, 1] to a set (fuzzy set/cluster). This means that, in X universe with collections of x objects, a fuzzy set F is represented as follows:

$$F = (x, \mu_F(x)) \mid x \in X, \mu_F(x) \in [0,1] \quad (1)$$

The notation $\mu F(x)$ in Eq. 1 is a membership degree of x in the set F, and its value is in between 0 to 1 i.e. [0, 1]. If there are more than a single set in the universe, x may have a membership degree to each of the sets. For example, x can have 0.8 degree in set F, and 0.2 degree in set G. This behavior has made fuzzy logic to have a good capability of handling uncertainty.

The above description is called fuzzification, and it is done in one of the components in FIS, namely fuzzifier. As a whole, an FIS comprises a set of components that map inputs to outputs as shown in figure 1. After an input is assigned with membership degrees by fuzzifier, it will then be evaluated by inference engine. The inference engine will process these fired membership degrees according to rules specified in rulebase component. These rules are in the form of "if-else" where all of the fired membership degrees will be mapped with membership degrees in another set of fuzzy sets in the output. This fuzzy sets is called output membership function (MF). Two processes are carried out in the inference engine, namely implication and aggregation. The former does the mapping between membership degrees in input MF to those of in output MF, as stated above, while the latter totaled up all the fired membership degrees that are produced by the implication process. As a result, a clipped fuzzy set will be produced. This fuzzy set will be sent to defuzzifier for conversion to an output. This conversion process involves specific algorithms such as centroid largest of maximum and bisector.

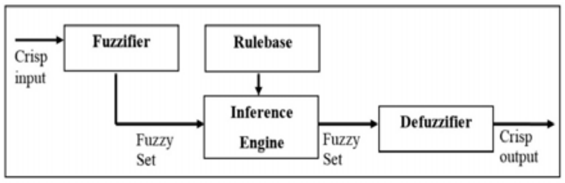


Fig. 1 Fuzzy inference system

The two most commonly used FIS are Mamdani and Sugeno. Mamdani FIS generates output by using fuzzy set format (in output MF) using specific algorithm as mentioned in the last paragraph. Sugeno, in the contrary, produces output based on a function of its input variables. For example, in a rule: if x is A and y is B then z is $f(x,y)$, the output is represented in a function of x and y. This is how Sugeno FIS differs from Mamdani FIS.

2.3 Methods

2.3.1 Development of Models

We developed the QoWS monitoring models using the original datasets mentioned in section 2.1. This development process involved a set of activities, which eventually materialized the models. The first activity was clustering validation process. This process was conducted with an aim to determine the optimal number of clusters for each of the original datasets. This process was required because the clustering algorithm, which was used to cluster the data points in the original datasets, needed the information on number of cluster prior to its execution.

A process known as clustering validity index (CVI) must be executed before clustering so that with the optimal number of clusters can be determined. CVI does not only determine the number of clusters, but it also identifies the optimal number of cluster that fits each of the underlying data sets.

CVI considers two parameters in its execution, namely compactness and separation of the data contained in an underlying data set [13, 14]. Compactness is the degree of closeness among the members of a cluster. This means that, a cluster with a good compactness has members (data points) that are close among each other. Normally, compactness is determined using variance, where a good compactness is derived from the lower variance values. On the other hand, separation measures how apart the clusters are among each other. Separation may consider one of these three measurements in its execution; the distance between the closest member of clusters, the distance between the most distant member of clusters, or the distance between the clusters' centers [14]. A clustering with a good separation has each of its clusters that is more distant from the other clusters [14]. To conclude, an optimal clustering is produced when it has high degrees of compactness and separation.

In determining the compactness and separation, some CVIs only consider membership degree values in computing their results. Some other CVIs will generate results based on membership degree values and also the underlying data set. It is evident that the second type of CVIs that takes both factors into consideration for performing their executions, has the ability to generate more representing results [14, 15].

In this work, we used Xie and Beni (XB) index because it has the ability to provide good representation of the datasets [16]. This is due to the fact that it considers both parameters, namely membership degree and underlying datasets, in its computation. Moreover, we also selected XB index due to the fact that it has the capability of performing well when the candidates of the number of clusters is in the range of two to ten [17]. Our work involved the number of candidates that are within this range.

Assume that there are a set of n data points, $X = \{x_1, x_2, \ldots, x_n\}$ and a vector of cluster centers, $V = [v_1, v_2, \ldots, v_c]$, where c is the number of clusters. Next, by assuming that μ_{ij} is the membership degree of x_j that belongs to v_i [18], XB index can be defined as follows [16, 19]:

$$V_{XB} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{m}\left\|x_j - v_i\right\|^2}{n \min_{i,j}\left\|v_i - v_j\right\|^2} \qquad (2)$$

The notation m in Eq. (2) represents a fuzzy weighting exponent. Eq. (2) comprises two parts; the degree of compactness, which is represented by its numerator; and the degree of separation that is represented by its denominator [19]. XB index solves its calculation by aiming for a candidate that generates a clustering validation result with the lowest value. This means that a candidate with the lowest value represents the most optimal number of clusters.

The maximum number of clusters, c_{max} and fuzzy weighting exponent, m, have to be identified before XB index executes. The candidate values of cluster numbers, c, are in the range of c=2 to c_{max}, which can be represented as $c_i \in \{2, 3, \ldots, c_{\max}\}$. In this work, we made an assumption of $c_{max} = 5$. This value is chosen because the results of XB validation upon the used data sets in this research suggest that the optimum numbers of clusters are within two to five.

As for m, previous work shows that the value in the range of 1.5 to 2.5 is the best to be employed [17]. Furthermore, an evident shows that the mean value of the particular range, m=2, is generally used in FCM computations [17]. Hence, in this work, we assumed m=2 for fuzzy weighting exponent.

The results of this clustering validation process are shown in Table 2.

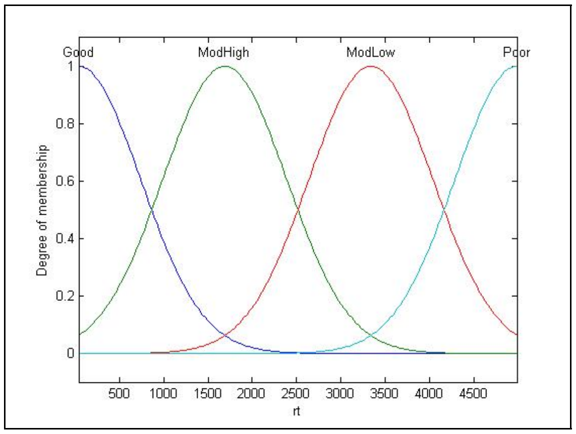
Table 2: Clustering validation results

QoWS Parameter	Number of Clusters			
	2	3	4	5
RT	2476.1	263.3	148.4	2362.9
AV	18.8	15.8	59.3	16.0
LAT	6535.2	537.8	958.1	26506.0

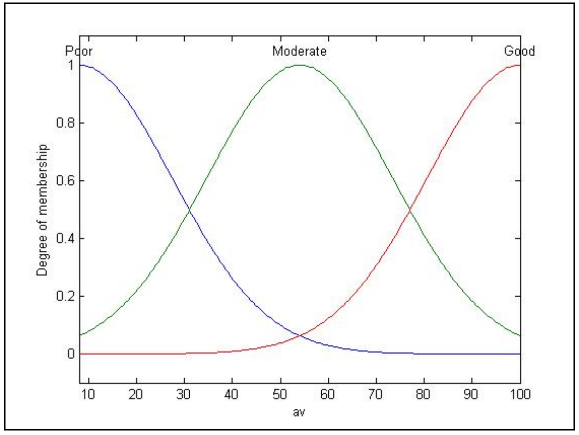
In Table 2, the notations RT, AV and LAT represent response time, availability and latency, respectively. In this work, we assumed that the candidates of number of clusters are 2 to 5. Xie and Beni index determines the optimal number of clusters by looking at the lowest clustering validation value. Based on Table 1, the optimal number of clusters for RT is 4. Meanwhile, the optimal number of clusters for AV and LAT is 3.

This means that, in the implementation of FIS, the input MF of RT will comprise four clusters, while the input MF of AV and LAT will contain three clusters.

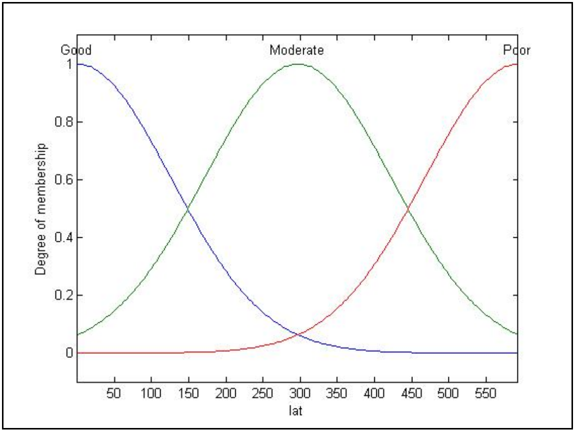
The next process was to conduct clustering on the three used original datasets. There was a difference between the construction of MF in Mamdani and Sugeno FIS. The MF of Sugeno FIS was constructed using grid partitioning method. With this method, the curves in the MF were constructed with equal distance among the curves' centers, and all of the curves have the same width. This is shown in the three MF in figure 2. The ranges of data points were 40-490 ms in RT dataset, 8-100% in AV, and 0.33-593.2 ms in LAT.



(a) Response time



(b) Availability



(c) Latency

Fig. 2 Sugeno input membership function

In contrast, Mamdani' MF were constructed based on the clustering of the three original datasets using FCM method. FCM clusters data based on fuzzy concept; each data point is assigned with a membership degree to each of the clusters (curves) in the MF [20.

FCM performs iterative computation with an objective to minimize an objective function. The aim of FCM clustering process is to produce the values of clusters' centers and membership degrees of each data point. These values are the optimal values based on given dataset. Let *n* as data points; represented by $\{X_1, X_2, \dots, X_n\}$ and *c* as number of clusters. Initially, FCM guess center of each cluster, c_i, $i=1, 2, \dots, c$. Using these centers' values, FCM assigns a membership degree for each cluster to every data point [21]. All of the generated membership degrees are contained in a matrix, *U*. This process is shown in Eq. 3.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \tag{3}$$

In Eq. 3, $dij = ||ci - xj||$ is an Euclidean distance from *j*th data point to the *i*th cluster center, while *m* is the FCM's fuzzifier value [21]. Then, FCM calculates the objective function as follows:

$$J(U, c_1, \dots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c}\sum_{j=1}^{n} \mu_{ij}^{m} d_{ij}^{2} \tag{4}$$

The iterative computation of FCM depends on the value of objective function; the computation stops if the objective function reaches its threshold that is represented by its minimum value [21]. As long as this threshold has not been reached, FCM will compute a new set of clusters' centers values, and then proceed with calculating the new membership degrees and objective function based on Eq. 3 and Eq. 4. The calculation of cluster centers is based on the following:

$$c_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} X_j}{\sum_{j=1}^{n} \mu_{ij}^{m}} \tag{5}$$

In this work, the clusters' centers produced by FCM are presented in Table 3. We named the four clusters in RT MF as *Good*, *Moderate High*, *Moderate Low* and *Poor*. For AV and LAT, the clusters were named as *Good*, *Moderate* and *Poor*.

Table 3: Clusters' centers

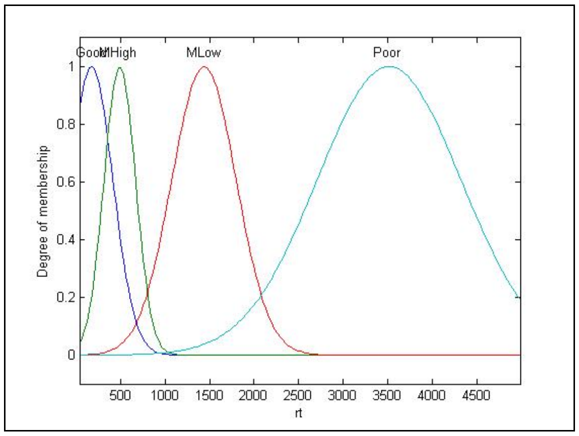
Cluster	Center	Cluster	Center	
	RT (ms)		AV (%)	LAT (ms)
Good	174.44	Good	90.69	12.11
Moderate High	491.35	Moderate	65.43	95.86
Moderate Low	1438.44	Poor	28.12	392.2
Poor	3516.57			

By using the clusters' centers in Table 2 and the membership degrees, *U*, MF curves were generated by using the following [22, 23]:

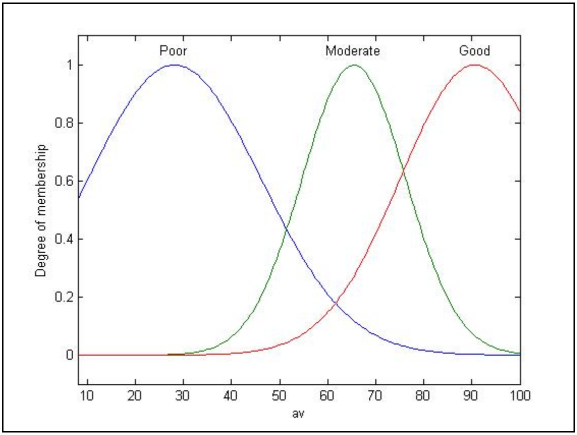
$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{6}$$

The MFs of the Mamdani FIS are shown in figure 3.

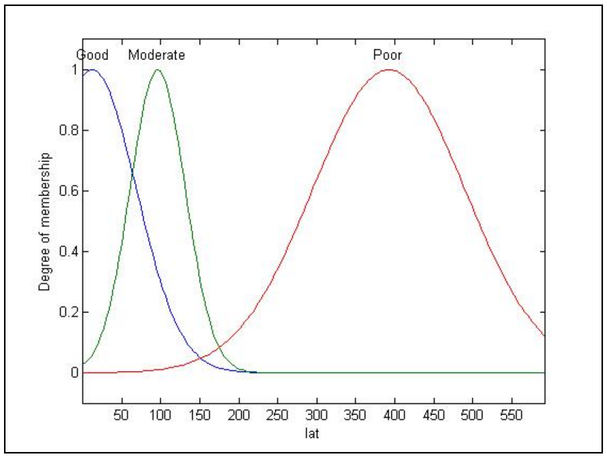
Overall, the QoWS monitoring models developed in this work is shown in figure 4. The constructed MFs, as described above, are contained in the FIS component of the monitoring node. The FIS is a multiple input with single output (MISO) model; a model that accepts three inputs, namely RT, AV and LAT for monitoring QoWS performance of web service. The monitoring process is conducted by comparing the delivered QoWS with the expected QoWS stated in an agreement. The model produces output of either "*Fail*" or "*Pass*" that represents meeting or not meeting the expected performance, respectively.



(a) Response time



(b) Availability



(c) Latency

Fig. 3 Mamdani membership function

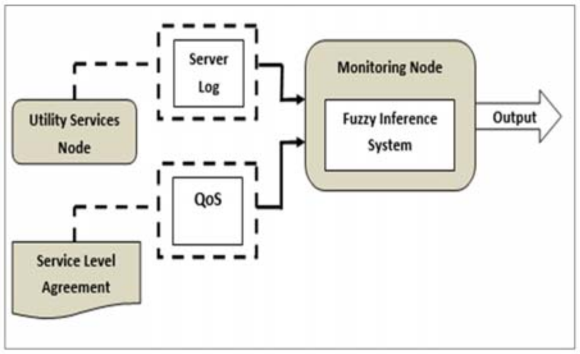


Fig. 4 QoWS monitoring model

2.3.2 Performance Measurement

Parallelization was implemented in the inference engine component of the FIS in figure 4. In this work, we compared Mamdani FIS model with two types of Sugeno FIS models. Those two types of Sugeno FIS differ in terms of their inference operation. One performed its computation using average operation, while another used summation operation.

To conduct performance measurement, we developed 25 synthetic models for each of the three types of FIS. These synthetic models were developed based on synthetic datasets. Synthetic datasets contained data points that originated from the original datasets, but they had been imposed with uncertainty (+/- with error). There were five error ranges introduced in this work. For RT, the ranges were +/-5ms, +/-10ms, +/-20ms, +/-30ms and +/-40ms. For AV, the ranges were +-1%, +-2%, +-3%, +-4%, +-5%. For LAT, the ranges of +-1.6ms, +-1.7ms, +-1.8ms, +-1.9ms, and +-2.0ms were used. Each of these ranges comprised five datasets, which means there were 25 synthetic models used in this work (5 ranges x 5 sets). This means that, including the original model, there were 26 models involved in performance measurement for each of the FIS. In total, 78 models were tested in this work (26 Mamdani FIS, 26 Sugeno-Average, 26 Sugeno-Sum).

The 78 models were executed with the same input dataset containing 9180 QoWS data points. These executions were repeated for three times. In each of the executions, the computation time was taken.

2.3.3 Methodology

Overall, this work was carried out based on the methodology shown figure 5.

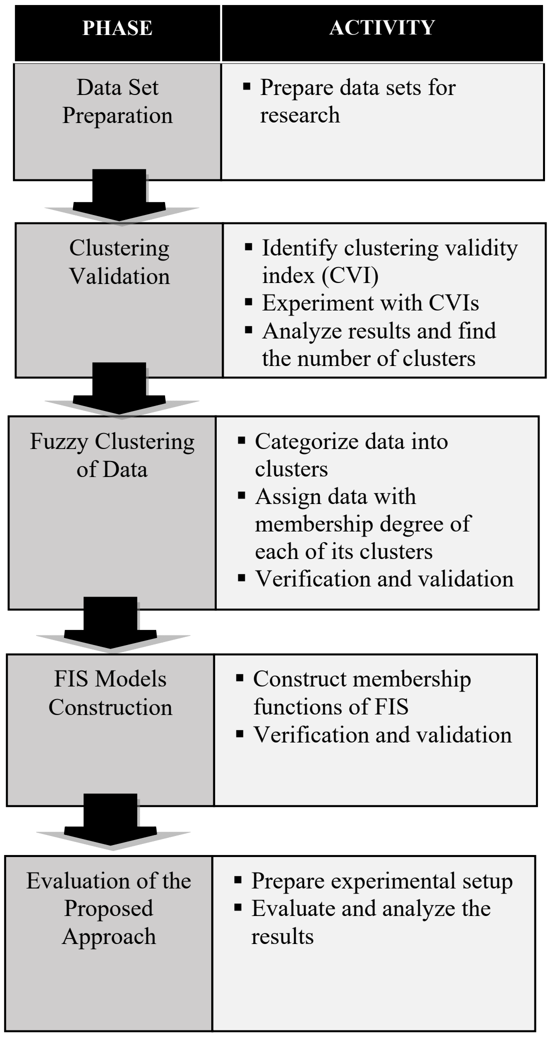

PHASE
ACTIVITY
Data Set Preparation
Prepare data sets for research
Clustering Validation
Identify clustering validity index (CVI)
Experiment with CVIs
Analyze results and find the number of clusters
Fuzzy Clustering of Data
Categorize data into clusters
Assign data with membership degree of each of its clusters
Verification and validation
FIS Models Construction
Construct membership functions of FIS
Verification and validation
Evaluation of the Proposed Approach
Prepare experimental setup
Evaluate and analyze the results

Fig. 5 Methodology

3. RESULTS

We ran 1 set of models (5 models) at one time. Each set was run for three times. The average time of these three runs are presented in this paper. Figure 6 shows the average time taken by Mamdani FIS models. It is shown that all sets had successfully reduced their computation time after parallelization.

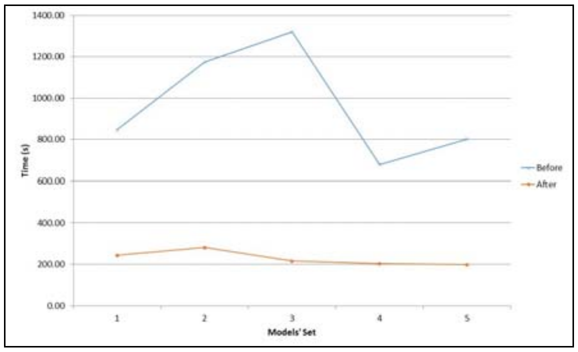


Fig. 6 Mamdani FIS models' performance

Figure 7 and figure 8 show the average time taken by Sugeno-Average and Sugeno-Sum FISs respectively. Similar to Mamdani FIS, both of Sugeno FIS models had performed better after they were parallelized.

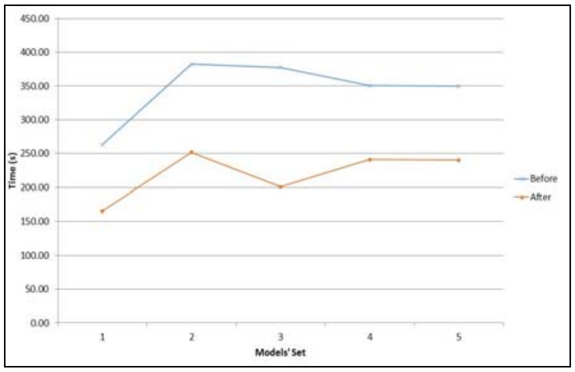


Fig. 7 Sugeno-Average FIS models' performance

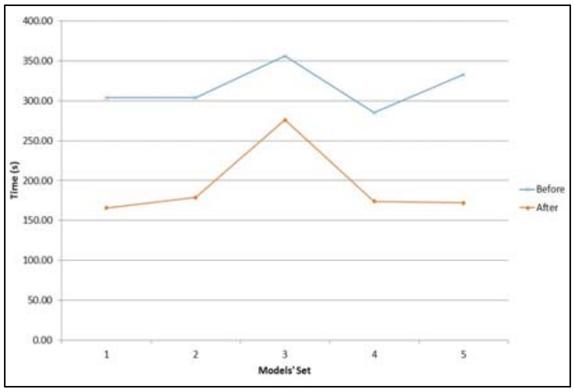


Fig. 8 Sugeno-Sum FIS models' performance

Table 4 shows the average time taken by all models before and after parallelization. It is shown that both Sugeno FIS models had performed better than that of Mamdani FIS model. However, it is also shown that the effect of parallelization is more significant in Mamdani model than that of the other two models. This is shown in figure 9; the time difference between before and after parallelization in Mamdani FIS models are between 70% to 83% time reductions. On the other hand, the time reduction in both of the Sugeno FIS models are between 22% to 48%.

Table 4: Average time taken by all models

Average Time	Mamdani (s)	Sugeno-Average (s)	Sugeno-Sum (s)
Before	964.31	344.68	316.15
After	228.32	219.96	193.36

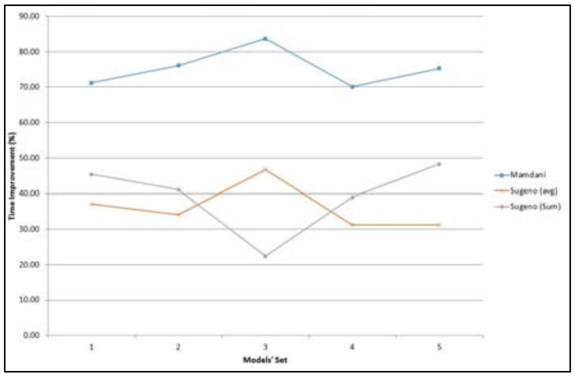


Fig. 9 Percentage of time difference before and after parallelization

4. DISCUSSION AND CONCLUSION

Parallelization has successfully reduced the computation time of the QoWS monitoring models. This can be seen from all types of FIS models, namely Mamdani, Sugeno-Average and Sugeno-Sum. Sugeno FIS models outperformed Mamdani FIS models by recording lower average computation time. This supports the findings of previous works that claimed that Sugeno FIS performed better than Mamdani FIS in terms of processing time [24, 25]. This is mainly due to the structure of Sugeno that carries out its operation using weighted average method which is less time consuming than Mamdani FIS's defuzzification. Nevertheless, the difference of average time taken after parallelization between Mamdani and Sugeno FIS models is very small, and both have acceptable performances.

The conducted experiments also showed that Mamdani FIS models had benefited from parallelization more than that of Sugeno FIS models. This was shown in a huge percentage of average time reduction in Mamdani FIS models after parallelization took place. A possible reason for this to happen is that the time consuming process of defuzzification and rules inferencing in Mamdani FIS is affected significantly by the concurrent processing after parallelization. Sugeno FIS models apply function of variables in its rules inferencing and weighted average in producing the final outputs. These two processes are efficient and hence the percentage of average time reduction after parallelization is not as much as that of Mamdani FIS models.

Our work has achieved the stated aim to compare the performance of fuzzy models of QoWS monitoring. For future work, we plan to investigate the performance of these models when they are executed in cluster computing. Other than that, this work will also be expanded to investigate the implementation of the models using the higher type of fuzzy logic i.e. type-2 fuzzy logic.

REFERENCES

[1] A. El Masri, A. Sardouk, L. Khoukhi, L. Merghem-Boulahia, and D. Gaiti, D., “Multimedia support in wireless mesh networks using interval type-2 fuzzy logic system,” 6th International Conference on New Technologies, Mobility and Security, 2014.

[2] M. Moharrer, H. Tahayori, L. Livi, A. Sadeghian and A. Rizzi, “Interval type-2 fuzzy sets to model linguistic label perception in online services satisfaction,” *Software Computing*, Vol. 19, 2015, pp. 237-250.

[3] P. Pangsub and S. Lekcharoen, “An adaptive type-2 fuzzy for control policing mechanism over high speed networks,” *The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2010.

[4] D. Mobedpour, C. Ding, “User-centered design of a QoS-based web service selection system,” *Service Oriented Computing and Applications*, Vol. 7, No. 2, 2011, pp. 1-11.

[5] B. Davide, G.B. Maria, M. Lusine, “Adaptive fuzzy-valued service selection,” *Proceedings of the 2010 ACM Symposium on Applied Computing*, ACM, Sierre, Switzerland, 2010, pp. 2467-2471.

[6] A. David, and K.T. Ruppa, “A fuzzy grid-QoS framework for obtaining higher grid resources availability,” *Proceedings of the 3rd international conference on Advances in grid and pervasive computing*, Springer-Verlag, Kunming, China, 2008, pp. 128-139.

[7] S. Wanita, W.L. Seng, and K. Shonali, “A fuzzy model for reasoning about reputation in web services,” *Proceedings of the 2006 ACM symposium on Applied computing*, ACM, Dijon, France, 2006, pp. 1886-1892.

[8] C. Wagner and H. Hagras, “zSlices—towards bridging the gap between interval and general type-2 fuzzy logic,” *FUZZ-IEEE*, 2008.

[9] C. Chantrapornchai and J. Pipatpaisan, “Fuzzy Application Parallelization using OpenMP,” *International Workshop on OpenMP*, 2010, pp. 122-132.

[10] H.T. Nguyen, V. Kreinovich, B. Wu and G. Xiang, “Computing Statistics under Interval Uncertainty: Possibility of Parallelization,” *Studies in Computational Intelligence*, Vol. 393, 2012.

[11] R. Bharathi, S.C. Shirwaikar and V. Kharat, “A Distributed, Scalable Parallelization of Fuzzy C-Means Algorithm,” *2016 IEEE Bombay Section Symposium*, 2016, pp. 1-7.

[12] E. Al-Masri, E. and Q.H. Mahmoud, “QoS-based Discovery and Ranking of Web Services,” *IEEE 16th International Conference on Computer Communications and Networks (ICCCN)*, 2007.

[13] M.J.A. Berry and G. Linoff, “Data Mining Techniques for Marketing, Sales and Customer Support”, John Wiley & Sons, Inc., USA, 1996.

[14] M. Halkidi, Y. Batistakis, Y. and M. Vazirgiannis, “On Clustering Validation Techniques,” *Journal of Intelligent Information Systems*, 17 (2-3), 2001, pp. 107–145.

[15] M.R. Rezaee, B.P.F. Lelieveldt and J.H.C. Reiber, “A new cluster validity index for the fuzzy c-mean,” *Pattern Recognition Letters*, 19 (3-4), 1998, pp. 237–246.

[16] X. Xie and G. Beni, “Validity measure for fuzzy clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 3, No. 8, 1991, pp. 841–847.

[17] N.R. Pal and J.C. Bezdek, “On cluster validity for the fuzzy c-means model, *IEEE Transactions on Fuzzy Systems*, 3 (3), 1995, pp. 370-379.

[18] Y. Tang, F. Sun, and Z. Sun, “Improved Validation Index for Fuzzy Clustering,” *Proceedings of the 2005 American Control Conference*, 2005.

[19] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy Sets and Systems*, 158, 2007, pp. 2095 – 2117.

[20] L. Wang and J. Wang, "Feature Weighting fuzzy clustering integrating rough sets and shadowed sets," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, 2012.

[21] H. Guldemır and A. Sengur, "Comparison of clustering algorithms for analog modulation classification," *Expert Systems with Applications*, Vol. 30, 2006, pp. 642-649.

[22] K.M. Tay, and C.P. Lim, "Optimization of Gaussian Fuzzy Membership Functions and Evaluation of the Monotonicity Property of Fuzzy Inference Systems," *2011 IEEE International Conference on Fuzzy Systems*, 2011.

[23] O. Castillo, and P. Melin, "Design of Intelligent Systems with Interval Type-2 Fuzzy Logic," *Type-2 Fuzzy Logic: Theory and Applications - Studies in Fuzziness and Soft Computing*, 223, 2008, pp. 53-76.

[24] A. Hamam, and N.D. Georganas, "A comparison of Mamdani and Sugeno fuzzy inference systems for evaluating the quality of experience of Hapto-Audio-Visual applications," *IEEE International Workshop on Haptic Audio visual Environments and Games*, 2008.

[25] J.J.V. doe Reis, T.R. Raddo, A.L.Sanches and B-H.V. Borges, "Comparison between Mamdani and Sugeno Fuzzy Inference Systems for the Mitigation of Environmental Temperature Variations in OCDMA-PONs," *ITCON 2015*, 2015.