Received: June 8, 2021. Revised: July 16, 2021.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Location and Time Aware Real Time Cloud Service Recommendation System Based on Multilayer Perceptron

S. Beghin Bose[1]* **S. S. Sujatha[2]**

[1]*ST Hindu College, Nagercoil, (Manonmaniam Sundaranar University) Abishekapatti, Trinelveli, 627012, Tamilnadu, India*
[2]*Computer science and applications, ST Hindu College, Nagercoil, Tamilnadu, India*
* Corresponding author's Email: s.beghinbose@gmail.com

Abstract: Cloud computing is the on-demand availability of internet-based computing services, especially software, large amounts of data storage, operating systems, and other computing resources. Service Level Agreement (SLA) violation is the most critical problem in cloud computing. SLA violation creates many problems for cloud service providers and cloud customers. Due to this, cloud customer gets low-quality cloud service. Thus, designing an effective cloud service recommendation algorithm is a critical research problem in cloud computing. The primary objective of this research is to determine the optimal cloud service from functionally equivalent cloud services that better fit the user's requirements (latency, throughput, response time, and cost). The efficiency of cloud services varies according to the time and location of the virtual machine. First, this proposed method determines the correlation between active user requirements and cloud services. Second, strongly correlated services are separated into two clusters based on the virtual machine's location and the cloud service's data transmission rate. For this purpose, two lightweight clustering algorithms have been proposed. A modified multilayer perceptron algorithm has been developed to recommend the optimal cloud service to the active user from the two clusters. The open-source WS-Dream dataset is used to train and validate the proposed MLP. The training efficiency, prediction accuracy, and performance of the proposed MLP-based cloud service recommendation system are experimentally compared to the existing cloud service recommendation systems analysed in the literature study [20, 2, 22, 23]. Compared to existing cloud service recommendation approaches, the MAE and RMSE values of the proposed cloud service recommendation system are less than one. In terms of accuracy, the suggested method obtains a precession of 94 %, a recall of 97 %, and an F1-Measure of 96 %, all of these are significantly better than the existing cloud service recommendation methods. Finally, experimental results prove that the overall performance of the proposed method's throughput (increase 10 MBPS), latency (reduce 10 ms), the response time (reduce 17 ms) and service recommendation time (reduce 5ms) is more robust than existing methods.

Keywords: Cloud computing, Cloud service recommendation system, Optimization, QoS, Artificial intelligence.

1. Introduction

Cloud computing is an on-demand computing service based on the internet [9]. It provides a platform, infrastructure, and software as a service respective of the need [10, 11]. This attracts the small and medium types of enterprises due to its popularity and flexibility. Due to cloud computing's enormous popularity, several service providers in the cloud industry deliver a diverse range of cloud services. Google App Engine, Amazon EC2, and Microsoft Azure are the most popular cloud service providers of this era [12]. Their primary objective is to provide the optimal cloud service according to the cloud customer's requirements.

In cloud computing, SLA is an important legal agreement between cloud service users and cloud service providers [13]. SLA contains information regarding the terms and conditions, details of cloud service and penalty details that cloud service provider has to pay to cloud customers during service-level violations. To avoid cloud service violations, cloud service providers should use a mandatorily good cloud service recommendation method [14]. By this, the unnecessary penalty is paid to the customers can be evaded. Thus, developing a good cloud service recommendation method is an urgent research problem.

Cloud computing's service architecture is generally composed of three layers: cloud service providers, application programming interfaces, and cloud service users [15, 16]. This is illustrated in Figure 1. Each cloud service provider's virtual machines and data centre are located in a different geographical location. These are interconnected together using secure networking. Cloud customers access cloud services using application programming interfaces and networking. Cloud computing speed varies depending on network traffic, time, and the geographical location of the cloud virtual machine. In general, throughput and latency are greatly dependent on the time (time to use cloud service), cloud virtual machine's geographical location. As a result, cloud service providers must design a better cloud service recommendation system for avoiding cloud service QoS violations.

Recently, substantial academic and business research has been conducted to improve the efficiency of cloud service recommendations. However, the majority of the methods are static and do not allow dynamic cloud services. Furthermore, these methods are not applied in real-time cloud architecture to demonstrate their efficacy. This proposed study suggested a reliable, highly accurate, and lightweight cloud service recommendation for cloud service providers. The method's important processing steps are presented below.

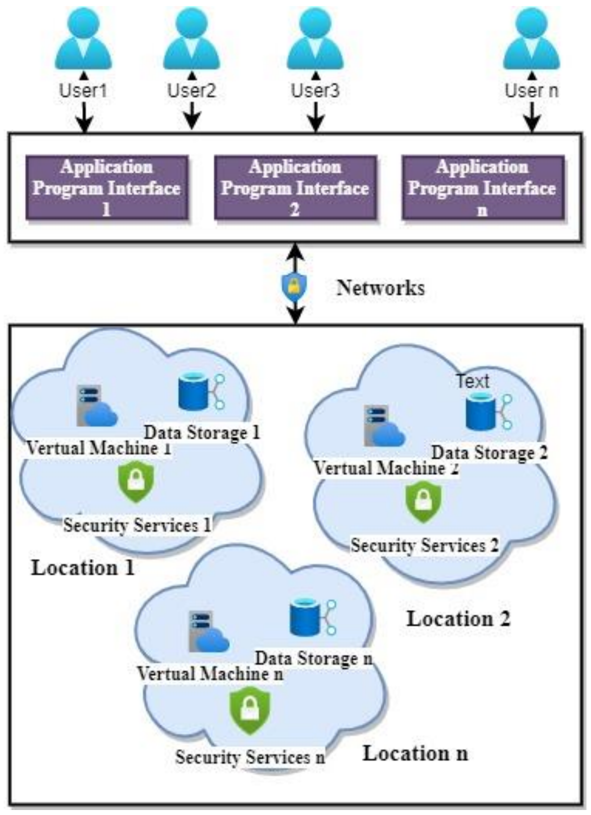


Figure. 1 The general architecture of service level cloud computing

Step1: First, the active user's important requirements such as throughput, response time, latency, and cost are listed. Next, list each cloud provider's cloud service details, such as cloud service type, minimum response time, maximum throughput, minimum latency, and so on.
Step2: Second, the correlation between these two lists (user requirement and cloud services) are determined. Cloud services with low correlation values are eliminated from the list. Values with a high correlation between user requirements and cloud services are then divided into two clusters.
Step3: Algorithm 1 creates the first cluster. This cluster is grouped according to the physical location of the cloud services' virtual machines and eliminates cloud services that exceed the threshold value.
Step4: Algorithm 2 generates the second cluster. Cloud services are grouped in this cluster according to the virtual machine's data transfer rate, and cloud services with a data transfer rate less than the threshold value are discarded.
Step5: Finally, the multilayer perceptron method extracts and recommends the ideal cloud service desired by the active user.

The important objectives of this proposed research are given below.

1. Deliver the high-quality service that clouds customers expect and thereby earn their trust.
2. Significantly improve the accuracy of optimal cloud service selection using deep learning methods.
3. The dynamic cloud service recommendation system is developed to overcome the limitations of existing static cloud service recommendation approaches in terms of adaptability.
4. Develop a cloud service recommendation system based on the location and the network bandwidth of the cloud services virtual machine, thereby compensate for the service loss.
5. Reducing cloud customer migration caused by poor QoS of cloud services and increasing the trust of cloud service users in cloud service providers.

Section 2 of this research discusses the merits and downsides of recently developed cloud service recommendation algorithms in detail. Section 3 details the four key components of the proposed cloud service recommendation method. Section 4 analyses the experiments and demonstrates the proposed method's efficacy. Finally, the conclusion and feature study are discussed.

2. Literature review

This section discussed recently developed methodologies for cloud service recommendation. Additionally, the merits and downsides of these techniques are discussed.

Priya, A.S.B., Bhuvaneswaran, R.S [1] created a cloud service recommendation system based on clustering for a multi-cloud environment. In this strategy, a trust agent module is introduced to recommend cloud services to the active user. The primary job of a trust agent is to assure the reliability of cloud services. Although this method recommends the most trustworthy service to the user, it ignores the cloud service's location and network bandwidth, which are likely to affect throughput, latency, and response time.

Jayapriya, K, Mary, N.A.B. and Rajesh, R.S [2] have developed a correlated QoS ranking algorithm and smoothing technique for cloud service recommendation. This method is mostly used to determine the user's similarities using the Pearson Correlation Coefficient (PCC), Kendall Rank Correlation Coefficient (KRCC), and Spearman Rank Correlation Coefficient (SRCC) approaches. In this method, experiments are carried out using the WS-Dream dataset. Unsuitable cloud service is often provided to the user because this method matches the cloud service based on the user's behaviour.

Indira. K, and Kavitha Devi, M.K [3] created the DBSCAN Algorithm to suggest cloud services in a multi-cloud environment. This method is mainly developed for online movie recommendations. Linear Discriminant Analysis (LDA) is used in this method to improve accuracy and extract optimal features. The extracted features are clustered by DBSCAN and the suitable cluster is recommended to the user. However, this recommendation technique is only suitable for online movie recommendations. The adaptability issue is more likely to come with this method.

Han SM, Mehedi Hassan M, Yoon CW, Lee HW, ad Huh EN [4] proposed a cloud resource recommendation system for optimal cloud service selection. In this method, cloud service recommendation and monitoring services are provided through the resource management module. The QoS values of the monitored cloud virtual machine are analyzed by the resource rank analysis module. In this method, only methods are proposed and no advanced algorithms are used to predict the new cloud service.

Liangmin Guo, Kaixuan Luan, Xiaoyao Zheng, and Jing Qian [5] proposed a cloud service recommendation system based on cloud user requirements. In this method, the cloud service is matched mainly on the basis of user behavior. In addition, the difference in ratings between the target user and similar users is taken into account to create the change. Service recommendation is based on the similarity of the user and the service, so unsuitable services are often matched to the cloud user.

Y. Wang, Q. He, X. Zhang, D. Ye and Y. and Yang [6] have developed a service recommendation system for multi-tenancy software architecture. Two approaches were proposed in this method for properly managing QoS in a multi-tenancy software architecture: clustering-based recommendation and a runtime service recommendation strategy based on Locality-Sensitive Hashing (LSH). In this method, the tenant is clustered, and important features are extracted according to the user's requirements. The runtime service recommendation mechanism suggests a tenant to the user based on their similarities. Although this strategy is excellent for managing tenant software architecture, the adaptability issue is not addressed satisfactorily.

In distributed cloud computing, L. Qi, X. Zhang, W. Dou and Q. Ni [7] invented a distributed locality-sensitive hashing algorithm for service recommendation. The WS-DREAM data set is utilized to validate the method's effectiveness. This solution addresses essential quality of service (QoS) criteria related to cloud computing, including efficiency and privacy. The approach was created specifically to address cloud privacy and security concerns. This will obviously have an impact on crucial QoS measures such as throughput, response time, and latency.

Y. Li, G. Tang, J. Du, N. Zhou, Y. Zhao and T. Wu [8] have developed a cloud service selection method based on user preference clustering and trust module. User preference clustering clusters a similar user according to the overall characteristics of the cloud user. This cluster recommends cloud services to users based on their similarities. Due to unsalable service recommendations, the false positive rate will be high, affecting overall precession and recall.

Shuai Ding, Yeqing Li, Desheng Wu, Youtao Zhang, and Shanlin Yang [20] created a time-aware cloud service recommendation system through the use of enhanced collaborative filtering and autoregressive integrated moving average model

(ARIMA) models. In this method, the time feature is extracted via optimized collaborative filtering. ARIMA model is used to predict cloud service for an active user. This approach compensates for QoS degradation caused by time variation. However, the QoS degradation caused by the location remains unaddressed.

K. Jayapriya, N. Ani Brown Mary, and R. S. Rajesh [2] designed a cloud service recommendation system by combining data smoothing and correlation techniques. This strategy highly depends on correlation techniques to optimize response time and throughput. The data smoothing technique has been developed to reduce the sparsity problem caused by the large QoS matrix. However, response time and throughput are location and time-dependent, which cannot be successfully solved using this strategy.

Mingsheng Fu, Hong Qu, Zhang Yi, Fellow, Li Lu, and Yongsheng Liu [22] have developed a recommendation system based on Deep Learning and collaborative filtering. Collaborative filtering computes the similarity between user and cloud service. Deep learning is used for service prediction. The experiment was conducted with the popular online movie services dataset MovieLens 1M and MovieLens 10M. Deep learning has improved prediction accuracy. However, there has been no improvement in throughput and latency.

Mingsheng Fu, Hong Qu, Zhang Yi, Fellow Li Lu, and Yongsheng Liu [23] developed a recommendation system that utilizes deep learning and collaborative filtering. Neighborhood information is extracted based on location using the Pearson Correlation Coefficient. Missing values are predicted by matrix factorization from neighborhood information. In this method, the service is predicted based on the user's location-based similarity.

1. Several cloud service selection methods have recently been created to provide the ideal cloud service to the cloud customer. However, the majority of them are static and dependent on historical data.
2. Many existing cloud service selection methods do not emphasize the network bandwidth and location of the cloud service. As a result, despite the improvement in the accuracy of cloud service selection, important QoS values such as response time and throughput are affected.
3. Many existing methods have been developed statically, so there is a high probability of facing adaptability issues when introducing new cloud services.
4. The majority of existing cloud service recommendation techniques suggest the cloud services based on the user's behaviour.

All the limitations in existing methods are solved by the proposed real-time cloud service recommendation system.

3. Proposed methodology

In this section, the proposed real-time cloud service recommendation system is explained in detail.

Accurately recommending cloud services that satisfy cloud consumers demand continues to be a challenge for cloud service providers. Cloud service recommendation problem has the potential to have a significant impact on cloud service providers, particularly in terms of trust degradation, revenue loss, and cloud client migration. This proposed research suggests a method for cloud service providers to solve this problem efficiently. Fig. 2 shows the architecture of the proposed real-time cloud service recommendation system.

3.1 Similar cloud services identification for the active user

Numerous cloud service providers offer a variety of cloud services to cloud customers on a pay-per-use basis. In this proposed research, the requirements of the cloud customer and customer details are listed first. The details of the customer requirements list and customer details are summarized in Table 1.

Customer requirements and information are stored in the variable C_{Req}, which is represented by the formula 1.

$$C_{Req} = \{C_{Id}, CS_{Type}, TP, RT, L, SS_T, SE_T, C, CL\} \quad (1)$$

Second, the specifics of each cloud service provided by the cloud service provider are listed. It is summarized in Table 2.

When a cloud service provider launches a cloud service, the details of that service are saved, as shown in Table 2. This listing simplifies the process of identifying cloud services and customer's behaviour. The details of the cloud service are stored in the variable CS_{DET}. The following formula is used to represent the cloud service details in the list.

$$CS_{DET}=\{CSP_{Id}, CS_{Id}, S_{Type}, MAX_T, MIN_{RT}, MIN_{Lat}, S_A, S_C, S_L\} \quad (2)$$

The correlation between active user requirements and cloud services is determined by formula 3. This formula is inspired by the Jaccard Similarity index [17].

$$Sim(C_{Req}, CS_{DET}) = \frac{|C_{Req} \cap CS_{DET}|}{|C_{Req}| + |CS_{DET}| - |C_{Req} \cap CS_{DET}|} \quad (3)$$

Table 1. Customer requirements details

Parameters	Description	Variable name
Customer Id	Indicates the unique identification number used to identify the cloud customer.	C_{Id}
Cloud service type	Indicates the type of cloud service desired by the cloud customer (e.x., software as a Service (SAAS), Infrastructure as a Service (IAAS), and others).	CS_{Type}
Throughput	Indicates the maximum throughput rate desired by the customer.	TP
Response Time	Indicates the customer's preferred minimum response time rate.	RT
Latency	Indicates the customer's preferred minimum latency rate.	L
Service start time	Indicates when the consumer intends to use the cloud service (the time when signing in to the cloud service)	SS_T
Service end time	Indicates the time when cloud customers sign out of the cloud service.	SE_T
Cost	Indicates the maximum price that the cloud customer pays for the cloud service.	C
Customer location	Indicates the geographical location of the cloud customer (latitude and longitude value).	CL

Table 3. Similarity values and similarity range

Similarity Values	**Similarity Range**
1	Very high correlated feature
0.9-0.7	High correlated feature
0.7-0.5	Medium correlated feature
0.5-0.3	Low correlated feature
0	No correlation

If the correlation between user requirements and cloud service is sufficiently strong, it will return 1, and else it will return 0. Table 3 summarises the Similarity range and Similarity Values.

To get the most suitable cloud service for the active user, cloud services with a similarity index between 1 and 0.7 are selected. This is described by formula 4.

Table 2. Details about each cloud service offered by the cloud service providers

Parameters	Description	Variable name
CSP Id	This refers to the unique identification numbers of cloud service providers.	CSP_{Id}
CS Id	This indicates the unique identification number of the cloud service.	CS_{Id}
Service Type	This indicates the type of cloud service (e.x., software as a Service (SAAS), Infrastructure as a Service (IAAS), and others).	S_{Type}
Maximum Throughput	This refers to the maximum throughput value of the cloud service.	MAX_T
Minimum Response Time	This refers to the minimum response time of the cloud service.	MIN_{RT}
Minimum Latency	This refers to the minimum latency of the cloud service.	MIN_{Lat}
Service Availability	This is the day and hour when the cloud service will be available online.	S_A
Service Cost	This refers to the cost of purchasing a cloud service per hour.	S_C
Service Location	This refers to the area (geographical location) where the cloud service's virtual machine or data centre is located (latitude and longitude value).	S_L

$$Cloud_{Services} = \begin{cases} 0.7 \leq TP \leq 1; \\ 0.7 \leq RT \leq 1; \\ 0.7 \leq L \leq 1; \\ 0.7 \leq C \leq 1. \end{cases} \quad (4)$$

3.2 Clustering the cloud services based on location

The physical location of cloud services (the geographical location of the cloud virtual machine or data centre) is closely connected to their performance. The best throughput and latency are achieved when the cloud user's data centre or virtual machine is relatively close. Formula 5 is used to determine the locations of similar cloud services. Cloud services are grouped into n number of clusters based on their geographical location. Finally, the cloud service cluster closest to the user's location is used for the final cloud service recommendation.

Received: June 8, 2021. Revised: July 16, 2021.

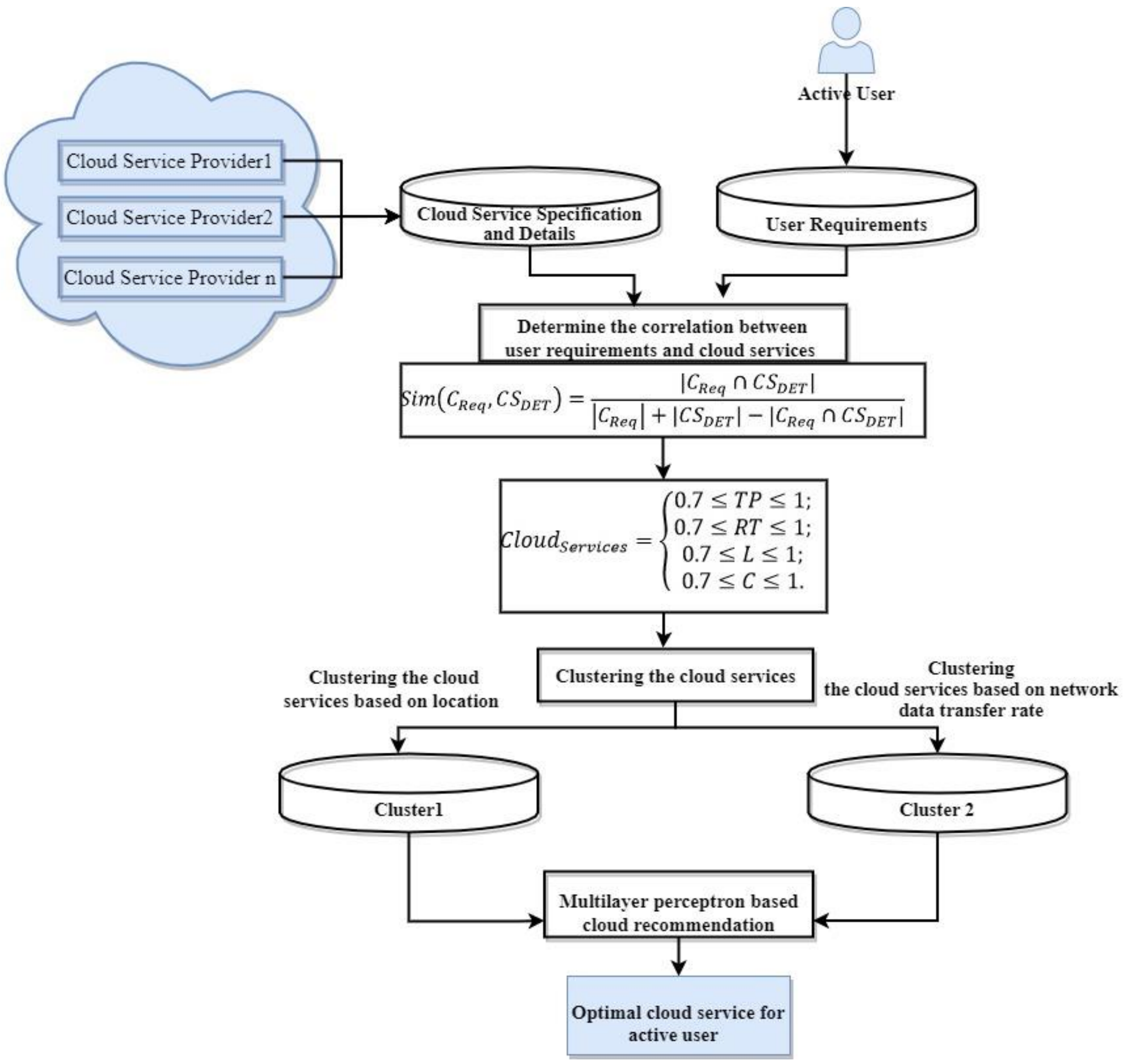

Active User
Cloud Service Provider1
Cloud Service Provider2
Cloud Service Provider n
Cloud Service Specification and Details
User Requirements
Determine the correlation between user requirements and cloud services
$Sim(C_{Req}, CS_{DET}) = \frac{|C_{Req} \cap CS_{DET}|}{|C_{Req}| + |CS_{DET}| - |C_{Req} \cap CS_{DET}|}$
$Cloud_{Services} = \begin{cases} 0.7 \le TP \le 1; \\ 0.7 \le RT \le 1; \\ 0.7 \le L \le 1; \\ 0.7 \le C \le 1. \end{cases}$
Clustering the cloud services
Clustering the cloud services based on location
Clustering the cloud services based on network data transfer rate
Cluster1
Cluster 2
Multilayer perceptron based cloud recommendation
Optimal cloud service for active user

Figure. 2 The overall architecture of the proposed location and time-aware real-time cloud service recommendation system based on a multilayer perceptron

$$CS_{Loc} = sin^2\left(\frac{\Delta\varphi}{2}\right) + cos\varphi1.cos\varphi2.sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (5)$$

CS_{Loc} describes the geographical location of a cloud service's virtual machine or data centre. φ Indicates the latitude of the cloud service's virtual machine. λ Denotes the longitude of the cloud service's virtual machine. Next, cloud services are clustered based on the locations of similar cloud services. Algorithm 1 performs location-based cloud service clustering.

Algorithm1 : Clustering the cloud services based on location

```
Input: CS_Det={( CS_Id1, $CS_{Loc}1$), ( CS_Id2, $CS_{Loc}2$), ( CS_Id3, $CS_{Loc}3$), ............ (CS_Idn, $CS_{Loc}n$)}, $\intercal_L$;
Output: C1;
Begin
for (i=1 to n)
{
Calculate the cloud service location using formula (5);
If( $CS_{Loc} \le \intercal_L$ )
{
C1← CS_Id;
}
Else
{
The cloud service is considers as a noise and eliminate from the list.
}
}
Return c1;
End
```

In algorithm 1, $\daleth_L$ represents the threshold value. The threshold can be set according to the needs of the cloud service provider. According to algorithm 1, a cloud service with a location exceeding the threshold value will be considered noise and eliminated from the list. The remaining cloud services are added to cluster 1 (c1).

3.3 Clustering the cloud services based on network efficiency

QoS is strongly intertwined with a cloud service virtual machine's network effectiveness. QoS values are higher whenever the network connected to the cloud service's virtual machine has a faster data transfer rate. At the same time, the data transferring rate will vary depending on the network traffic. Proper clustering of cloud services based on data transferring rate can definitely improve throughput and reduce latency. In this proposed method, cloud service virtual machines are clustered based on data transferring rate by algorithm 1. The data transferring rate of each cloud service virtual machine are determined by formula 6.

$$\mathrm{d_{rate}} = \frac{(Din_c - Din_p) \times 8 \times 100}{C_t - P_t} \qquad (6)$$

$\mathrm{d_{rate}}$ refers to the data transfer rate of the cloud virtual machine. Din_c this variable describes the current bit value of the data received rate (incoming data) to the cloud virtual machine. Din_p this variable describes the previous value of the data received rate (incoming data) to the cloud virtual machine which is in bits. To convert byte, the value 8 is used. $\mathrm{C_t}$ this variable denotes the current time of data arrival (incoming data) to the cloud virtual machine. P_t this variable describes the previous time of data arrival (incoming data) to the cloud virtual machine.

Algorithm2: Clustering the cloud services based on network data transfer rate

```
Input: CS_Det={( CS_Id1, $\mathrm{d_{rate}}$1),
( CS_Id2, $\mathrm{d_{rate}}$2), ( CS_Id3, $\mathrm{d_{rate}}$3), .............
(CS_Idn, $\mathrm{d_{rate}}$n)}, $\daleth_{dr}$;
Output: C2;
Begin
for (i=1 to n)
{
    Calculate the virtual machine's data transfer rate using formula (6);
    If($\mathrm{d_{rate}} \geq \daleth_{dr}$)
    {
        C2← CS_Id1;
    }
    Else
    {
        The cloud service is considers as a noise and eliminate from the list.
    }
}
Return c2;
End
```

In algorithm 2, $\daleth_{dr}$ denotes the data rate's minimal threshold value. The threshold value is adjusted dynamically based on the data rate. Cloud services with low data rate threshold values are eliminated from the list. Cloud services with a data rate greater than the threshold value are added to cluster 2 (c2).

3.4 Recommending the optimal cloud service to the active user using MLP

Deep learning is being used in many fields due to data growth and the advent of GPU processors. The cloud QoS data is multi-dimensional and noisy. In general, Multilayer Perceptron (MLP) performs better with multidimensional and noisy data. Due to this, MLP has been chosen for cloud service recommendation in this research. MLP is the sub-class of deep learning [18, 19]. The proposed MLP is shown in figure 3. Furthermore, the momentum coefficient and the Exponential-Based Learning Rate Schedule (EBLRS) will be used in this study to maximize MLP's accuracy and efficiency.

The details of cloud services in clusters generated by Algorithm1 and Algorithm2 are given as input to MLP in vector format. Each vector member encapsulates the information contained in the formula (2). MLP's input format is given below.

$$X1 = \{\overrightarrow{V1_1}, \overrightarrow{V1_2}, \overrightarrow{V1_3}, \dots\dots\dots \overrightarrow{V1_n}\},$$
$$X2 = \{\overrightarrow{V2_1}, \overrightarrow{V2_2}, \overrightarrow{V2_3}, \dots\dots\dots \overrightarrow{V2_m}\} \qquad (7)$$

X1 has n number of cloud services and X2 has m number of cloud services which are based on algorthm1 and algorithm2. During the learning process, these input vectors are divided into *n*-dimensional weighted vectors. Once vector representation is complete, MLP is used for cloud service recommendations. Its hidden layer facilitates the learning and prediction process. The key function of this deep learning is given below.

$$a_H = f1(W_H a_H + b_H) \qquad (8)$$

Received: June 8, 2021. Revised: July 16, 2021.

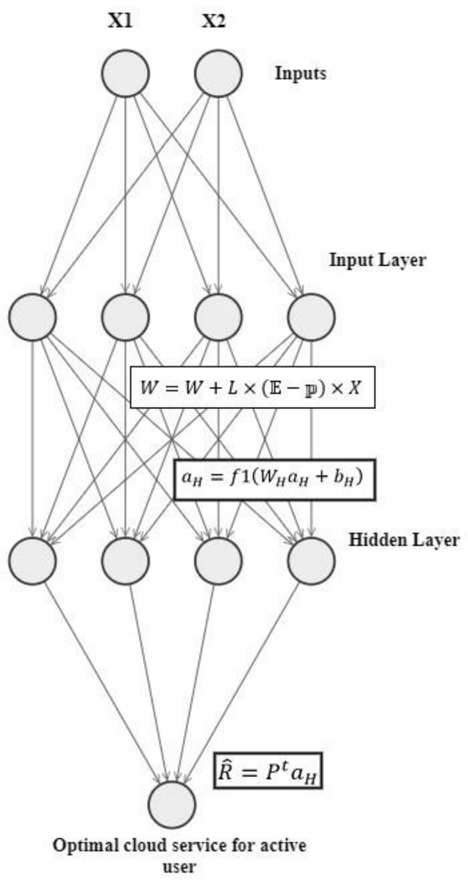

X1
X2
Inputs
Input Layer
$W = W + L \times (\mathbb{E} - \mathbb{p}) \times X$
$a_H = f1(W_H a_H + b_H)$
Hidden Layer
$\hat{R} = P^t a_H$
Optimal cloud service for active user

Figure. 3 Proposed MLP model for cloud service recommendation

$f1$ represents the activation function of MLP. H denotes the number of hidden layers in the proposed deep learning model that are $H = \{h_1, h_2,. h_3, \dots \dots \dots h_n\}$. W_H and a_H denote network mapping and weighted vector. The non-linear activation function ReLU is used for higher-level feature learning in the proposed MLP. Finally, the training feature t and the last hidden layer a_H are transformed for cloud service recommendation system.

$$\hat{R} = P^t a_H \tag{9}$$

Where, $\hat{R}$ denotes the optimal cloud service for active user. According to this study, dynamically change the network weight when a prediction error is discovered.

$$W = W + L \times (\mathbb{E} - \mathbb{p}) \times X \tag{10}$$

- Where W denotes the initial weight.
- Where L denotes learning rate.
- Where $\mathbb{E}$ denotes the expected result.
- Where $\mathbb{p}$ denotes predicted result.

3.5 MLP training optimization

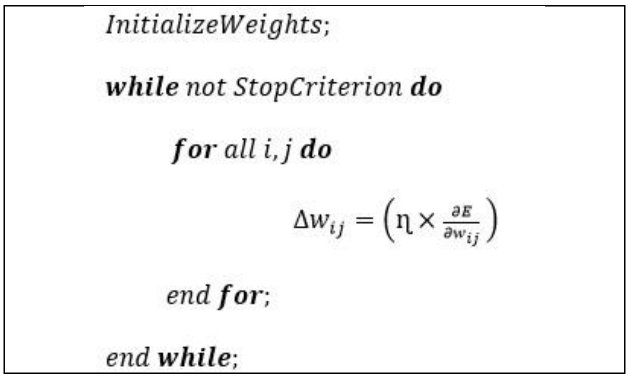
Learning plays a significant role in MLP. Even the best MLP model may fail to give an actual result if the best learning method or algorithm is not used in MLP. Therefore, using advanced learning algorithms can benefit MLP, significantly saving time for training and improving prediction accuracy. Accordingly, learning is an essential part of MLP. A variety of training algorithms are used to train the MLP model. One of the most widely used methods is the back propagation (BP) algorithm [24, 25]. In general, the BP algorithm takes more time for training because it requires lower learning rates for stable learning. The main reason for this slowdown, it adjusts the weight of the hidden neurons in the forward and backward positions until the global minima arrive. As a result, for many practical applications, it will be prolonged. The pseudo-code of the traditional BP algorithm is given in Figure 3.6. η> 0 defines the learning rate which controls the learning speed. The primary purpose of this research is to reduce unnecessary computing resources and reach global minima more efficiently.

In traditional back propagation the weight updating process is based on formula 11. Where $w_{(t+1)}$ denotes weight increment, η denotes learning rate, and $d(E)$ denotes total error. The learning rate is a small positive number. Its range is usually between 0 and 1. It determines how fast or slow the neural network model trains.

$$w_{(t+1)} = w_t - \eta \frac{d(E)}{d(w_1)} \tag{11}$$

The momentum coefficient is used with the traditional BP algorithm to improve this proposed method's accuracy and training speed. The weight of MLP is updated by formula 12. α Indicates momentum coefficient.

$$\Delta w_t = \alpha \Delta w_{(t-1)} + \eta \frac{d(E)}{d(w_1)} \tag{12}$$



```
InitializeWeights;
while not StopCriterion do
    for all i,j do
        $\Delta w_{ij} = \left(\eta \times \frac{\partial E}{\partial w_{ij}}\right)$
    end for;
end while;
```

Figure. 4 Pseudo-code of the traditional BP algorithm

The efficiency of the neural network depends on the learning rate ƞ and coefficients α. If ƞ is low, it will take longer to reach the global minima, whereas ƞ is high there is a chance of oscillation. In this research, the Exponential-Based Learning Rate Schedule (EBLRS) method is used to select the optimal learning rate ƞ. It is implemented by formula 13. Where $ƞ_0$ is the initial learning rate, d is the decay rate and, n is the iteration step

$$ƞ_n = ƞ_0 e^{-dn} \tag{13}$$

The error of each output neuron is calculated by the squared error function. To get the total error, the error values are summed by Eq. (14) where t_v is the target value and o_v is the observed value.

$$E_t = \sum_1^n \frac{1}{2}(t_v - o_v)^2 \tag{14}$$

In this research, the learning efficiency of MLP has been improved in the following algorithm.

```
Algorithm3 : Training optimization of MLP
Step 1) Initialize all weights and biases.
Step 2) The input and output of the proposed MLP model is normalized by Eq. (12). Where $x$ is the value to be normalized, α is the minimum value, b is the maximum value and $x^!$ is the normalized value.
        $x^! = \frac{x-\alpha}{b-\alpha}$    (15)
Step 3) Set the initial value of the Momentum coefficient $\alpha$ and EBLRS ƞ.
Step 4) Compute the total error between the target value and the observed value by Eq. (14).
Step 5) Adjust the MLP weight $\Delta W$ using Eq. (11) and adjust the learning rate ƞ using Eq. (12).
Step 6) Continue step 5 until the error rate reaches the minimum.
```

4. Experimental results

4.1 Hardware and software details

A range of experiments are performed to assess the proposed cloud service recommendation system's efficiency and compared it with the state-of-the-art methodologies. JDK 9.1, Eclipse 8.2, Cloudsim 3.0.3, and Windows 10 have been used for these experiments. In addition, Matlab 2020 is used to simulate results and evaluate the efficiency of the proposed MLP. An artificial intelligence server configured with an Intel Xeon processor, 2TB of storage, a NIVIDA GPU processor, and 128 GB of RAM is used to conduct these experiments efficiently. For further real-time implementation, five desktop PCs configured with 1 TB of storage, 8 GB of RAM, and an Intel i7 processor are used.

To demonstrate the efficacy of the proposed method, the recently developed most popular cloud service recommendation methods are employed, which are listed below.

- Enhanced Collaborative Filtering (ECF) [20].
- Correlated QoS Ranking Prediction (CQRP) [2].
- Deep Learning-Based Collaborative Filtering (DLBCF) [22].
- Neighbourhood Enhanced Matrix Factorization (NEMF) [23].

4.2 Dataset details

The most prominent WS-DREAM dataset is used in this case to assess the efficacy of the proposed approach and train the deep learning model. This is the freely available open-source dataset [26, 27]. It is widely used by QoS researchers. This data set contains real-time QoS values of throughput and response time collected from web services. Furthermore, this dataset comprises real-time QoS values used by users with varying computing requirements at varying time intervals from various geographical areas around the world. Fig. 3 summarizes the most significant details of the WS-DREAM dataset.

4.3 Training the model and training efficiency analysis

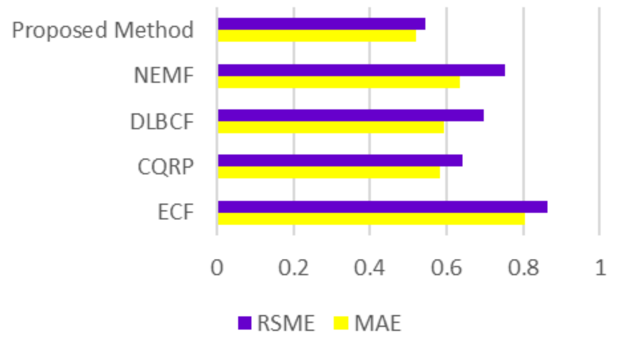
The Proposed MLP based cloud service recommendation model is trained using the most important features in the WS-DREAM data set. The data is separated into three different sizes for modeltraining: 20%, 40%, and 60%. Finally, a detailed comparative study is conducted to demonstrate the proposed model's training efficiency.

Table. 3 details of WS-DREAM dataset

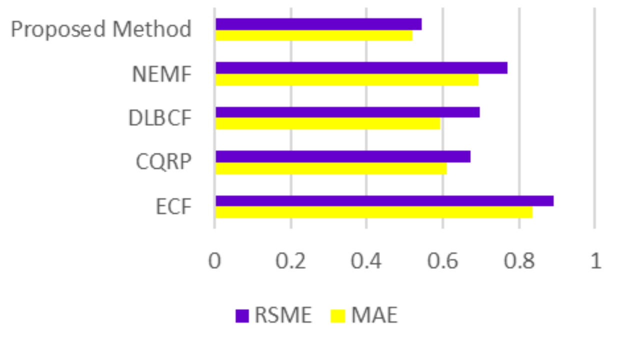
Statistics	Values
Total number of web service invocations	1974674
Total number of users	339
Total number of services	5824
Total number of countries	30
Total number of web service countries countries	73
Mean of response time	1.43 s
Standard deviation of response time	31.9 s
Mean of throughput	10286 kbps
Standard deviation of throughput	531.85 kbps

Two important accuracy metrics have been used to evaluate the training efficiency of the proposed cloud recommendation system: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Absolute Error (AE) refers to the total amount of error of the proposed method. Mean Absolute Error (MAE) refers to the average Absolute Error of the proposed cloud service recommendation system. It is calculated by formula 16.

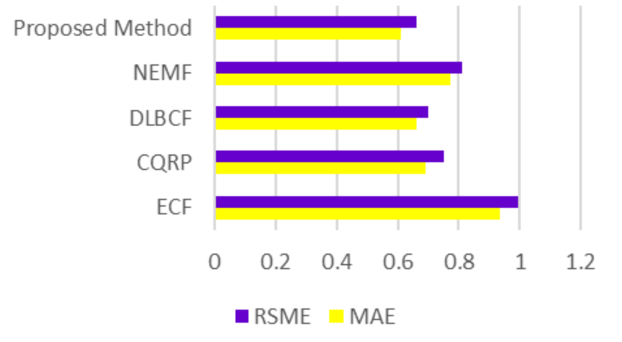
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \tag{16}$$



(a)



(b)



(c)

Figure. 5 Throughput MAE and RMSE values associated: (a) 20% training data, (b) 40% training data, and (c) 60% training data

n Indicates the total number of errors of the proposed cloud service recommendation method. $x_i - x$ Indicates the absolute errors of the proposed cloud service recommendation method.

$$RSME = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|yi - \widehat{yi}|^2} \tag{17}$$

The RMSE returns the standard deviation of the difference between the proposed cloud service recommendation model's observed value and estimated value.

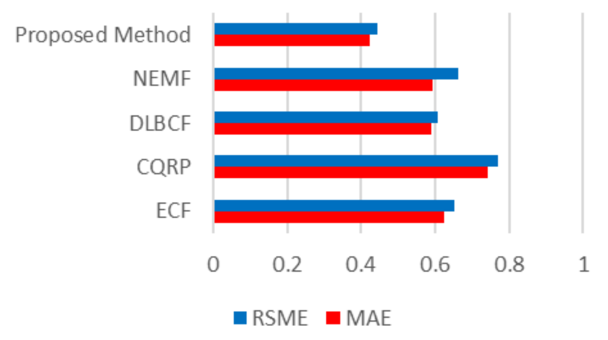
The throughput error values (MAE and RMSE) comparison of the proposed method and the state-of-the-art cloud recommendation methods are shown in Fig. 5(a), (b) and (c). According to Fig. 5(a), (b) and (c) the proposed technique has lower error values than existing cloud service recommendation approaches. The MAE and RMAE of existing cloud service recommendation techniques increase when the data density increases. At the same time, the MAE and RMAE values of the proposed method are not greatly increased. From these experimental results, it is obvious that the location of the cloud service's virtual machine has a strong correlation with throughput. As a result, the geographical location of the cloud service's virtual machine is critical for cloud service recommendation. In addition, the experimental results prove the deep learning is far more efficient in recommending cloud services than mathematical models.

Fig. 6(a), (b) and (c) shows the proposed method's response time error values (MAE and RMSE) with the state-of-the-art cloud service recommendation methods. The MAE and RMAE values of the response time in the proposed method are very low. In the proposed method, time is also used as an important input parameter for cloud service recommendation. In general, the cloud service quality changes as time changes because the data transfer rate varies from time to time due to network traffic.

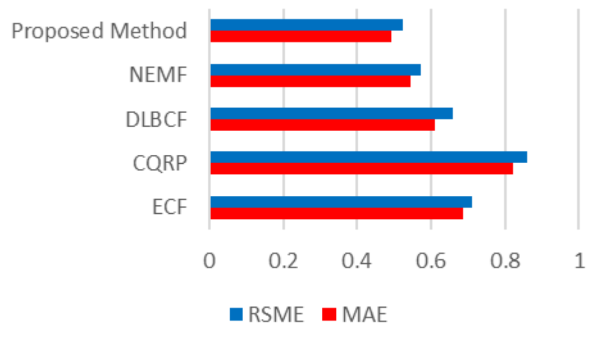
4.4 Performance metrics and accuracy evaluations

The proposed cloud service recommendation system is tested with standard accuracy metrics, including Precession (P), Recall (R), and F1-Measure (F1-M). Time (T) is used to determine the proposed cloud service recommendation model's prediction time. The prediction time indicates the amount of time required to recommend the optimal cloud service to the active user. The recently developed QoS recommendation methods are used to

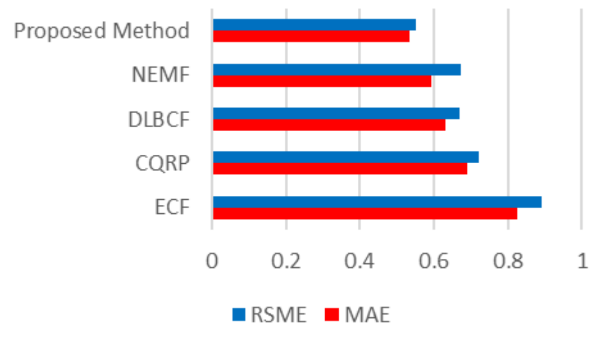
Received: June 8, 2021. Revised: July 16, 2021.



(a)



(b)



(c)

Figure. 6 Response time MSE and RMSE values associated: (a) 20% training data, (b) 40% training data, and (c) 60% training data

demonstrate the effectiveness of the proposed QoS recommendation method.

Performance metrics such as Precession (P), Recall (R), and F1-Measure (F1-M) are determined by the vital accuracy variables: True Positive Cloud Service Recommendation ($TPCS_R$), True Negative Cloud Service Recommendation ($TNCS_R$), False Positive Cloud Service Recommendation ($FPCS_R$), and False Negative Cloud Service Recommendation ($FNCS_R$). If the proposed cloud service recommendation system's FP and FN rates are low, the prediction accuracy will be excellent.

True Positive Cloud Service Recommendation ($\boldsymbol{TPCS_R}$): If the proposed method recommends the cloud service that best suits the active user's requirement, it is called True Positive (TP), and the variable $TPCS_R$ denotes it.

True Negative Cloud Service Recommendation ($\boldsymbol{TNCS_R}$): Rarely, Cloud service providers may not have a cloud service that meets the active user's requirements. If the proposed method correctly predicts this, it is called True Negative (TN). The variable $TNCS_R$ represents this.

False Positive Cloud Service Recommendation ($\boldsymbol{FPCS_R}$):If the proposed approach predicts a cloud service, which is opposite to the active user's requirement, this is considered a False Positive (FP). The variable $FPCS_R$ is used to denote this.

False Negative Cloud Service Recommendation ($\boldsymbol{FNCS_R}$) : Cloud service providers have a service that best meets the requirements of the active user, but the proposed method is unable to predict that service correctly. This is called False Negative (FN). This is referred to as the $FNCS_R$ variable. Formula 18 is used to get the precession rate for the proposed method.

$$P = \frac{TPCS\_R}{\mathrm{TPCS_R+TPCS_R}} \tag{18}$$

Formula 19 is used to determine the recall rate for the proposed method.

$$R = \frac{\mathrm{TPCS\_R}}{\mathrm{TPCS_R+FNCS_R}} \tag{19}$$

The proposed method's F1-Measure is calculated using formula 20.

$$F = \frac{2(P\times R)}{\mathrm{P+R}} \tag{20}$$

The precession rate of the proposed method and the recently developed cloud service recommendation methods are given in Fig. 7.

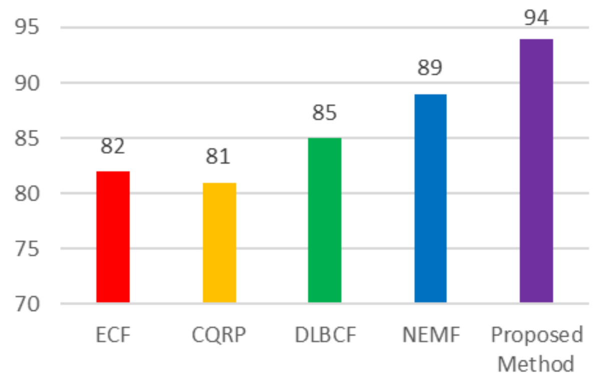


Figure. 7 Precession rate comparison of the proposed method with the existing cloud service recommendation methods

Received: June 8, 2021. Revised: July 16, 2021.

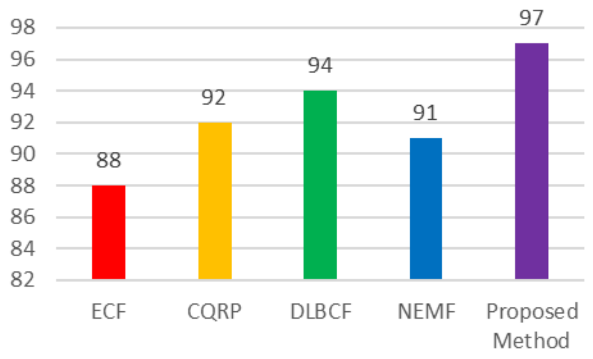


Figure. 8 Recall rate comparison of the proposed method with the existing cloud service recommendation methods

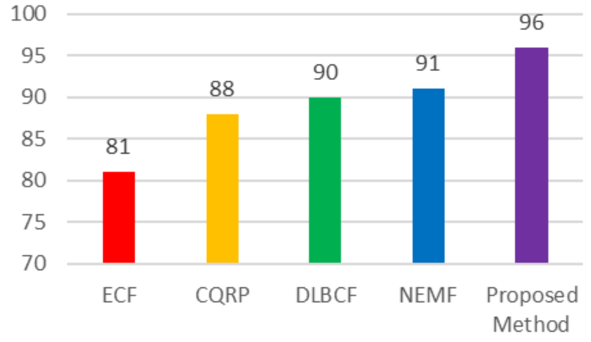


Figure. 9 F1-Messure comparison of the proposed method with the existing cloud service recommendation methods

Fig. 7 makes it clear that the proposed method achieves the highest precession rate of 94 %. Thus, it becomes evident that the proposed method recommends the exact cloud service that the active user needs.

Fig. 8 illustrates the recall comparison between the proposed method and the recently established cloud service recommendation methods. Comparative results indicate that the proposed method has a higher recall rate. The proposed approach yields a 97% recall rate. The proposed approach also has a very low FN rate because of the low recall rate. Additionally, the proposed method makes it clear that it does not recommend unsuitable cloud services to cloud users.

Fig. 9 demonstrates the comparative results for the proposed method's F1-Measure. F1-Measure is typically beneficial when the precession and recall rates are high. The proposed method achieves the highest precession rate of 96%. These experimental results demonstrate that the proposed method recommends the most appropriate cloud service for the active user.

4.5 Performance analysis

This section compares the proposed method's throughput, latency, and response time with recently established cloud service recommendation methods. Fig. 10 depicts the throughput comparison results of the proposed method with the recently existing cloud service recommendations approaches. The proposed approach has a substantially greater throughput value than the existing approaches. To obtain the most reliable results, the experiment is repeated 30 times at three different time intervals (morning hours, peak hours, and night hours), and the average value is used for this comparative analysis. The comparison results of the response time are displayed in Fig. 11, which makes it clear that the proposed method has a short response time. Fig. 12 depicts the latency values of the proposed approach versus existing approaches. It is evident that the proposed approach has a very low latency rate. These simulation results indicated that clustering (based on the virtual machine location and virtual machine networks of the cloud service) could significantly reduce the latency and response time of the cloud service. The comparison results for the proposed method's prediction time are shown in Fig. 13. The simplest clustering algorithms used in this method significantly reduce the cloud service recommendation time.

4.6 Discussion

Only the best QoS recommendation system can

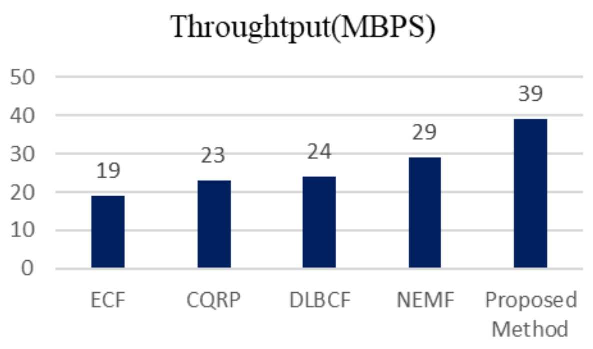


Figure. 10 Throughput comparison results of the proposed method with the existing cloud service recommendation methods

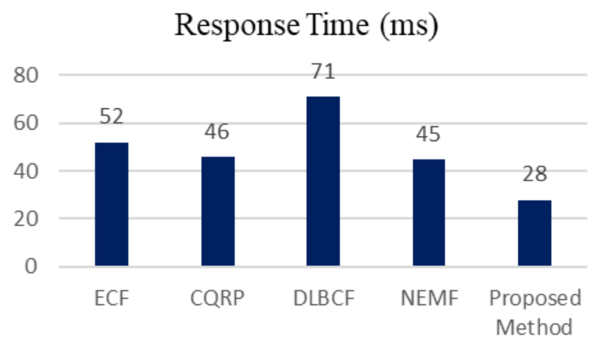


Figure. 11 Response time comparison results of the proposed method with the existing cloud service recommendation methods

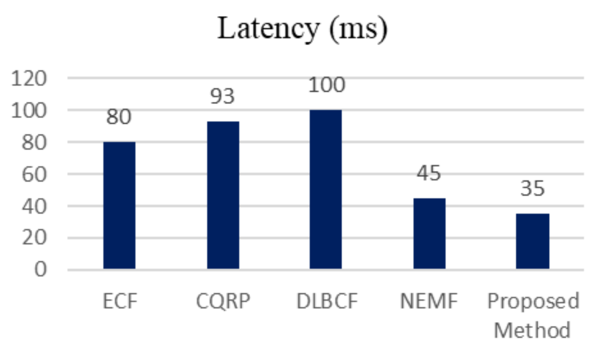


Figure. 12 Latency comparison results of the proposed method with the existing cloud service recommendation methods

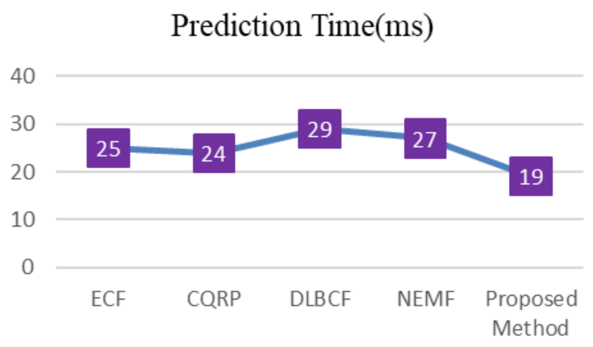


Figure. 13 Recommendation time comparison results of the proposed method with the existing cloud service recommendation methods

deliver the ideal cloud service that cloud customers expect.

Numerous cloud service recommendation systems have been developed using methods such as Collaborative Filtering [20], Correlated Quality of Service Ranking [2], Deep Learning [22], and Matrix Factorization [23]. Most of these existing works recommend cloud services based on similar user interests and behaviour. In addition, these methods use the most complicated and time-consuming computational methods to recommend cloud services. The important QoS metrics of cloud computing, including throughput, response time and latency, vary depending on location and network bandwidth. However, the existing methods do not recommend cloud service based on location and network bandwidth. Two simple clustering algorithms have been developed in the proposed method to recommend cloud services based on location and network bandwidth. As a result, the proposed cloud service recommendation model outperforms the competition in terms of throughput (highest 39 Mbps), the response time (lowest 28ms), latency (35ms) and prediction time (lowest 19ms). MLP has also been enhanced to improve the recommendation system's prediction efficiency. It has the lowest error rate and the highest prediction accuracy. The proposed methodology achieves the highest precession of 94%, recall of 97%, and F1-Measure of 96%.

5. Conclusion

Due to the popularity of cloud computing, the number of cloud service providers and cloud users are increasing day by day. However, even if there is a minor issue with a cloud service provider, cloud users would seek out another cloud service provider. This research has built an ideal cloud service recommendation system to simply avoid this service migration problem and ensure that cloud users deserve the highest possible service.

The primary goal of this research is to achieve optimal cloud service for active users by employing three major modules: user service correlation, location and network bandwidth based clustering and improving multilayer perceptron for optimal cloud service prediction. User service correlation module selects highly correlated cloud services from functionally equivalent cloud services suitable for user requirements. Proposed clustering algorithms compensate for the service loss caused by the cloud service's virtual machine location and the cloud service's virtual machine data transfer rate. Clustering improves throughput and latency and greatly reduces the time it takes for cloud service recommendations. Furthermore, the multilayer perceptron algorithm has been used to recommend the optimal cloud service to the active user in this research. Finally, the efficiency of the proposed method has been demonstrated by a detailed implementation. When training, the proposed method generally produces low error values (MAE and RMSE). Additionally, it provides 5% more precession, 6% more recall, and 7% more F1-Measure than existing approaches.

Conflicts of Interest

S. Beghin Bose and S. S. Sujatha has no conflict of interest.

Author Contributions

S. Beghin Bose and S. S. Sujatha conducted the Abstract, Introduction, Literature Review, Proposed Methodology, Experimental Analysis, Comparative Study, Results and Discussion, and Conclusions.

References

[1] A. S. B. Priya and R. S. Bhuvaneswaran, “Cloud service recommendation system based on clustering trust measures in multi-cloud environment”, *Journal of Ambient Intelligence*

and Humanized Computing, Vol. 12, pp. 7029-7038, 2021.

[2] K. Jayapriya, N. A. B. Mary, and R. S Rajesh, "Cloud Service Recommendation Based on a Correlated QoS Ranking Prediction", *Journal of Network and Systems Management*, Vol. 24, pp. 916-943, 2016.

[3] K. Indira and M. K. K. Devi, "Multi Cloud Based Service Recommendation System Using DBSCAN Algorithm", *Wireless Personal Communications*, Vol. 115, pp. 1019-1034, 2020.

[4] S. M. Han, M. M. Hassan, C. W. Yoon, H. W. Lee, and E. N. Huh, "Efficient Service Recommendation System for Cloud Computing", In: *Proc. of International Conf on Grid and Distributed Computing*, Berlin, pp. 117-124, 2009.

[5] L. Guo, K. Luan, X. Zheng, and Jing Qian, "A Service Recommendation Method Based on Requirements for the Cloud Environment", *Journal of Control Science and Engineering*, Vol. 2021, pp. 41-51, 2021.

[6] Y. Wang, Q. He, X. Zhang, D. Ye, and Y. Yang, "Efficient QoS-Aware Service Recommendation for Multi-Tenant Service-Based Systems in Cloud", *IEEE Transactions on Services Computing*, Vol. 13, pp. 1045-1058, 2020.

[7] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A Distributed Locality-Sensitive Hashing-Based Approach for Cloud Service Recommendation From Multi-Source Data", *IEEE Journal on Selected Areas in Communications*, Vol. 35, pp. 2616-2624, 2017.

[8] Y. Li, G. Tang, J. Du, N. Zhou, Y. Zhao, and T. Wu, "Multilayer Perceptron Method to Estimate Real-World Fuel Consumption Rate of Light Duty Vehicles", *IEEE Access*, Vol. 7, pp. 63395-63402, 2019.

[9] S. S. Yadav and Z. W. Hua, "CLOUD: A computing infrastructure on demand", In: *proc. of 2010 2nd International Conf on Computer Engineering and Technology*, China, pp. 423-426, 2010.

[10] A. Cardoso and P. Simões, "Cloud Computing: Concepts Technologies and Challenges", *In: proc. of International Conference on Virtual and Networked Organizations, Emergent Technologies*, Portugal, pp. 127-136.

[11] S. Kroschwald, "Cloud Computing", *Informationelle Selbstbestimmung in der Cloud*, pp. 7-16.

[12] M. Firdhou, S. Hassan, O. Ghazali, and M. Mahmuddin, "Evaluating Cloud System Providers: Models, Methods and Applications", *Cloud Systems in Supply Chains*, pp. 121-149.

[13] I. K. Kim, J. Hwang, W. Wang, and M. Humphrey, "Guaranteeing Performance SLAs of Cloud Applications under Resource Storms", *IEEE Transactions on Cloud Computing*, pp. 1-14, 2020.

[14] S. Mubeen, S. A. Asadollah, A. V. Papadopoulos, M. Ashjaei, H. P. Breivold, and M. Behnam, "Management of Service Level Agreements for Cloud Services in IoT: A Systematic Mapping Study", *IEEE Access*, Vol. 6, pp. 30184-30207, 2018.

[15] G. G. Contreras, J. L. Garrido, S. B. Díaz, and C. R. Domínguez, "A Context-Aware Architecture Supporting Service Availability in Mobile Cloud Computing", *IEEE Transactions on Services Computing*, Vol. 10, No. 6, pp. 956-968, 2017.

[16] D. Lin, A. C. Squicciarini, V. N. Dondapati, and S. Sundareswaran, "A Cloud Brokerage Architecture for Efficient Cloud Service Selection", *IEEE Transactions on Services Computing*, Vol. 12, No. 1, pp. 144-157, 2019.

[17] K. D. Yong, E. Meryam, and J. Hongtaek, "Examining Bitcoin mempools Resemblance Using Jaccard Similarity Index", In: *Proc. of Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Korea (South), pp. 287-290, 2020.

[18] J. Tang, C. Deng, and G. Huang, "Extreme Learning Machine for Multilayer Perceptron", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 27, No. 4, pp. 809-821, 2016.

[19] J. Tang, C. Deng, and G. Huang, "Extreme Learning Machine for Multilayer Perceptron", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 27, No. 4, pp. 809-821, 2016.

[20] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model", *Decision Support Systems*, Vol. 107, pp. 103-115, 2018.

[21] N. P. Kumar, "Hybrid User-Item Based Collaborative Filtering", *Procedia Computer Science*, Vol. 60, pp. 1453-1461, 2015.

[22] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A Novel Deep Learning-Based Collaborative Filtering Model for Recommendation System", *IEEE Transactions on Cybernetics*, Vol. 49, No. 3, pp. 1084-1096, 2019.

[23] Y. Feng and B. Huang, "Cloud manufacturing service QoS prediction based on neighbourhood

enhanced matrix factorization", *Journal of Intelligent Manufacturing*, Vol. 31 pp. 1649-1660, 2020.

[24] R. Rojas, "The Backpropagation Algorithm", *Neural Networks*, pp. 149-182.

[25] P. Munro, Backpropagation. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA, 2011.

[26] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, "Towards Automated Log Parsing for Large-Scale Log Data Analysis", *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2017.

[27] I. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS Prediction for Runtime Service Adaptation via Adaptive Matrix Factorization", *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2017.