

Article

Hybrid Approach to Improve Recommendation of Cloud Services for Personalized QoS Requirements

Sadhna Samadhiya and Cooper Cheng-Yuan Ku *

Institute of Information Management, National Yang Ming Chiao Tung University, Hsinchu City 300093, Taiwan; sadhna23.c@nycu.edu.tw

* Correspondence: cooperku@nycu.edu.tw

Abstract: Cloud-service recommendation systems make suggestions based on ratings provided by cloud users. These ratings may contain sparse data, which makes it difficult to speculate on suitable cloud services. Moreover, new cloud users often suffer from cold-start difficulties. Therefore, in this study, we attempt to better overcome these two challenges, i.e., cold start and data sparsity, using a hybrid approach incorporating neural matrix factorization, deep autoencoders, and suitable questionnaires. The proposed approach provides a list of the top N cloud service providers for old cloud users based on the predicted preferences using quality of service data and asymmetrically weighted cosine similarity. To address the cold start problem, we design a questionnaire to survey new user preferences and suggest personalized cloud providers accordingly. The experiments based on the Cloud Armor database demonstrate that our approach outperforms other models. The proposed approach has a precision of 85% and achieves a mean absolute error (MAE) of 0.05 and root-mean-square error (RMSE) of 0.14 for the differences between the input and predicted values. We also receive a satisfaction level of nearly 78.5% for recommendation lists provided to new cloud service customers.

Keywords: cloud services; deep learning; hybrid approach; neural matrix factorization; deep autoencoder



Citation: Samadhiya, S.; Ku, C.C.-Y. Hybrid Approach to Improve Recommendation of Cloud Services for Personalized QoS Requirements. *Electronics* **2024**, *13*, 1386. <https://doi.org/10.3390/electronics13071386>

Academic Editor: George Angelos Papadopoulos

Received: 23 February 2024
Revised: 23 March 2024
Accepted: 5 April 2024
Published: 6 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cloud computing is an increasingly common prototype for accessing computational resources. Cloud service providers primarily offer three service models: software as a service, platform as a service, and infrastructure as a service [1]. In addition to an interface allowing users to access their virtualized resources, it comprises computing power, networking, and storage. Figure 1 shows the basic categories mentioned above for cloud services. According to Kelkar [2], implementing cloud computing involves several challenges, including carefully managed service-level agreements, confidentiality, integrity, and availability. Therefore, it is not easy to implement recommendation services for cloud computing, especially when cloud service providers have new users and no prior information about their preferences in the database [3]. Although many cloud services are available, developing a recommendation system based on quality of service (QoS) is tedious, and establishing trustworthiness is not simple. It is also challenging for cloud service providers to offer new cloud users expectations that align with their complex demands.

Mohamed et al. [4] provide a detailed survey of recommendation systems, approaches, and problems. As well known, collaborative filtering (CF), content-based filtering, and hybrid filtering (a combination of CF and content-based filtering) are the three major types of recommendation systems. One of the critical drawbacks of many approaches is due to the cold start problem, which occurs because of the need for more data on new entries and robustly affects precision [5,6]. However, the cold start problem cannot be effectively solved using content-based methods and CF, which are the traditional approaches of recommender systems [7]. For cloud service recommendations, the data sparsity challenge is rising

due to the partial ratings provided by existing cloud customers and the various troubles encountered in obtaining the required information [8]. Some strategies for handling this issue are provided by integrating neural networks into recommender systems. The accuracy of suggestions and predictions has risen regardless of whether more profound learning algorithms with more layers or the ability to retrieve hidden data are used [7]. The cold start problem, which is due to insufficient data on new customers, purchase history, and browsing behavior, and the data sparsity issue, potentially led by the interactions among a small number of elements, are two major concerns for the design of recommendation systems. These problems negatively affect the suggestion and significantly reduce the variety of options on the specific platform. Many deep-learning (DL) algorithms have already been proposed to reduce the influence of data scarcity. Some studies discuss prediction ratings based on prior knowledge gained from the service ratings of existing cloud users. To improve the rating prediction, a deep autoencoder could extract the QoS latent representations from the input data [9]. However, further improvement is possible using a hybrid deep-learning architecture with neural factorization and deep autoencoder (NFDA). The proposed framework incorporates auxiliary information to learn sparse data matrices via matrix factorization with neural network (NeuMF) [10]. The QoS latent representations are extracted from the input data using the concatenation of two models to enhance rating prediction. Furthermore, we carefully structure a questionnaire from the literature review to handle cold start issues. We aim to alleviate the problem of data sparsity and cold start and refer a personalized recommendation list with N service providers to the cloud users using asymmetrically weighted cosine similarity (AWCS). In summary, this paper makes the following contributions:

1. We design the hybrid DL cloud recommendation system and direct the coming old users to cloud service providers according to their preferences.
2. Applying the above system with explicit historical data for old cloud users and using survey data to provide suggestions for new ones or existing customers with no personal data at our hand.
3. Our solution performs better than other proposed models with enhanced personalized cloud service recommendations.

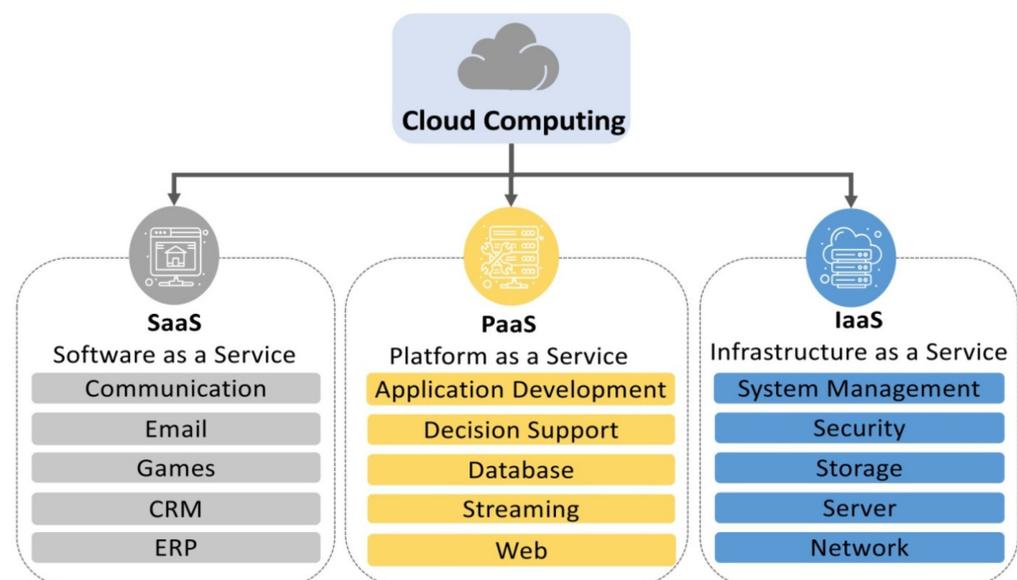


Figure 1. Cloud computing categories.

2. Literature Review

Several studies have been conducted to reduce the issues of cold starts and missing data. For example, the hybrid DL-based recommendation system efficiently addresses

the cold start problem and data sparsity challenge. Berisha and Bytyçi [7] have examined around 40 research papers that attempted to use neural networks to solve the cold start problem. They have studied the implementation of neural networks in recommender systems, how they are integrated into systems, which neural network algorithms are more efficient at dealing with the cold start problem, and which algorithms have improved recommendation accuracy. Zheng et al. [11] recommended a CF process for the cloud service recommendation that uses Spearman's coefficient. This method predicts the ratings and rankings of cloud services.

In [12], the authors proposed a strategy for recommending cloud services based on learning and data mining approaches to improve user visibility. In addition, they suggest using this approach to study how various users interact with cloud services (consulting, using, unsubscribing, etc.) to specify user requirements in real time while accounting for changes in requirements over time. In [13], association rules and clustering were used to address the issues of accuracy and sparsity for the suggested items. They addressed the cold start problem by employing the root-mean-square error to determine the standard of the rated songs.

An autoencoder is an artificial neural network used to learn an encoded representation of an input data set, and it is generally used to reduce data dimensionality [9]. A feed-forward neural network in the form of an autoencoder has an input layer, a hidden layer, and an output layer. A deep autoencoder is an enhancement of the simple autoencoder. The extra hidden layers allow the autoencoder to recognize more difficult underlying patterns in the data mathematically. CF requires additional layers to handle increasingly complicated data. The NeuMF method [10] is a CF technique used in recommender systems. CF algorithms estimate a user's interests based on their previous activities or the behaviors of similar users. NeuMF divides the user-item interaction matrix into two lower dimensional matrices: user and item matrices. These matrices represent the latent factors or features contributing to a user's interest in objects. According to Han et al. [14], the cosine similarity index determines the two vectors' similarities in the inner product space. The cosine of the angle formed by two vectors is used to determine whether or not they roughly point in the same direction. To calculate the cosine similarity between the two vectors, we first computed their dot product. The sum of the two vectors' element-wise products is the dot product. Then, they divide the dot product by the product of the magnitudes of the two vectors. The square root of the sum of the squares of a vector's elements determines its magnitude.

Pearson correlation and cosine similarity are two popular measurements of similarity. However, these metrics regard any resemblance between two users or products as equivalent. According to Mishra et al. [15], the AWCS enables the same features in many vectors with varying weights. This implies that a given user may value "action" differently from another, and the similarity calculation takes these different feature weights into effect. To calculate the missing QoS values, CF has become a popular solution to the sparsity problem. Fu et al. [16] presented a novel neighbor-based QoS prediction technique for service suggestions. In addition, the popularity of each stable or unstable candidate is determined by the closest graph method. A notion and computation technique is used to characterize the stable state of users and services with measurable QoS values.

Khan et al. [17] discussed how DL-based techniques improve performance and create representations from scratch. With new studies verifying its usefulness, DL has also influenced the research of information recovery and recommender systems. A comprehensive overview of deep-learning-based rating prediction algorithms is presented in this article to assist new researchers interested in the field. A deep autoencoder is powerful for reducing the data dimensions. Bougteb et al. [18] helped accurately fill in the missing data values. The autoencoder learns the user's interest, reconstructs the user's missing ratings, and then uses singular value decomposition (SVD)++ decomposition to hold information on correlation; it also provides a deep analysis of the top N recommendation items.

Another study used the hybrid model to recommend cloud services; Sahu et al. [19] anticipated an ensemble deep neural network, a hybrid of neighborhood and neural networks, to predict QoS values. Deep CF is essential when recommending web services. Xiong et al. [20] addressed this by merging CF with textual content; furthermore, researchers offer a novel hybrid deep learning strategy for web service recommendation. A deep neural network that can differentiate the complicated relationships amid mashups and services is seamlessly linked to their invocation interactions and functions. The effectiveness of the proposed methodology in service recommendation is demonstrated by experiments carried out on a real-world web service dataset, which show that this strategy can produce a superior recommendation performance compared to several state-of-the-art methods.

Efficient assessment of the user-side quality of cloud components has emerged as a pressing and significant research issue. Evaluating every cloud component available from the user's perspective would be costly and unfeasible. It is easier to perform appropriate service recommendations with QoS values. Most current algorithms approach the QoS prediction problem by combining the QoS values of the user's similar neighbors, which naturally results in local optimization. The authors of [21] formulated it as a global search optimization issue in the QoS value distribution space and introduced a new algorithm, PSO-USRec. The approach enhanced and customized particle swarm optimization (PSO) by smoothing the outlier particles and diversifying the initial solutions. On a well-known public QoS dataset, comparison experiments were carried out to verify the efficiency of the PSO-USRec algorithm. A hybrid recommendation strategy was proposed by Nabli et al. [22] and provided the active user with a list of personalized cloud services. User and service clustering are the basis of this strategy. Cloud services are recommended according to the user's functional and non-functional needs and preferred QoSs. Following that, the services are organized based on cost and reliability, and the recommendation approach provides a list of various cloud services.

Memory and model-based techniques have been combined into hybrid CF techniques. A few papers [23,24] presented privacy protection strategies to maintain user privacy while achieving high QoS prediction accuracy, as gathering QoS values may provide privacy challenges. The advantages of model-based approaches and memory are combined in these strategies. They involve many calculations as well. To address cold start and data sparsity issues, Patro et al. [25] developed a novel methodological technique, termed sparsity and cold start aware hybrid recommended system (SCSHRS), to reduce the impact of data sparsity and cold start in recommendation systems. Chen et al. [26] proposed a hybrid QoS prediction technique that utilized empirical mode decomposition and a multivariate long short-term memory (LSTM) model. Their feature extraction and preliminary data treatment retain considerable potential for improvement, as do their network architecture and QoS forecast accuracy. A combined QoS prediction technique for web services based on the profound fusion of features was presented to address the issue of insufficient accuracy of QoS prediction. Ding et al. [27] considered the common characteristics of multi-class QoS and the hidden environmental preference information in QoS. Initially, QoS data were modeled as a user-service bipartite graph. Next, features were extracted and mapped using a multi-component graph convolution neural network (CNN). Finally, the same dimensional mapping of multiple classes of QoS features was accomplished using the weighted fusion approach.

To avoid costly and time-consuming web service invocations, Xiong et al. [28] suggested a location-aware collaborative method for the QoS prediction of web services using the usage history of service users. They begin by gathering and processing client-side geographical location data. Then, a method is developed to obtain a higher prediction accuracy for web service QoS values based on the gathered QoS and location data, integrating geographical location constraints, the (location-aware matrix factorization method) LFM, and considering the imbalanced data distribution. To build a sophisticated ML system, Handri and Idrissi [29] used singular value decomposition (SVD) and matrix factorization. This strategy was designed to achieve multi-criteria decision-making support and dominate

query approaches. Concurrently, applying the resilient distributed dataset paradigm in cloud computing creates an environment conducive to big data management. According to extensive experimental results, the new method has an advantage over other Topk algorithms in precision and scalability. Although DL-based techniques have been proposed for QoS prediction, their neural network topologies still need to be improved to increase prediction accuracy. The study of Zhang et al. [30] added individual assessments and multi-stage multi-scale feature fusion to a DL model for more precise QoS prediction. This model extracts three-scale features, i.e., global, local, and individual, using non-negative matrix factorization. Similarity in distance is utilized to identify similar users and services. Then, a multi-stage deep neural network is designed to fuse multi-scale features, and the individual evaluations are input to each stage to correct QoS features.

Many cloud suppliers offer services, frameworks, and tools that can be utilized to deploy applications on a cloud infrastructure to simplify the migration process. Identifying the infrastructure and services that best enable a given application to deliver the desired performance at the lowest possible cost may be difficult. Chahal et al. [31] offered a strategy for moving a recommender system based on a deep-learning model to a serverless architecture and machine learning (ML) platform. They also present their experimental evaluation of Lambda, a serverless platform service, and SageMaker, an AWS-ML platform. They discussed the tradeoff between performance and cost when leveraging the cloud infrastructure.

3. Proposed Model

This proposed hybrid solution is structured to address the challenges of the sparse data and cold start problems. The flowchart in Figure 2 provides an overview of this method and its related design. The QoS matrices of old cloud users and service providers are inputted to NeuMF to obtain a dense rating matrix; subsequently, these matrices are utilized by a deep autoencoder to mine more deeply into the QoS of cloud users and providers to make a better prediction. We use these ratings to suggest the top cloud providers to old users. For new users or existing customers with no personal data at our hand, survey data are preprocessed to generate the rating matrices. They are used to calculate AWCS on QoS features to obtain personalized service recommendations.

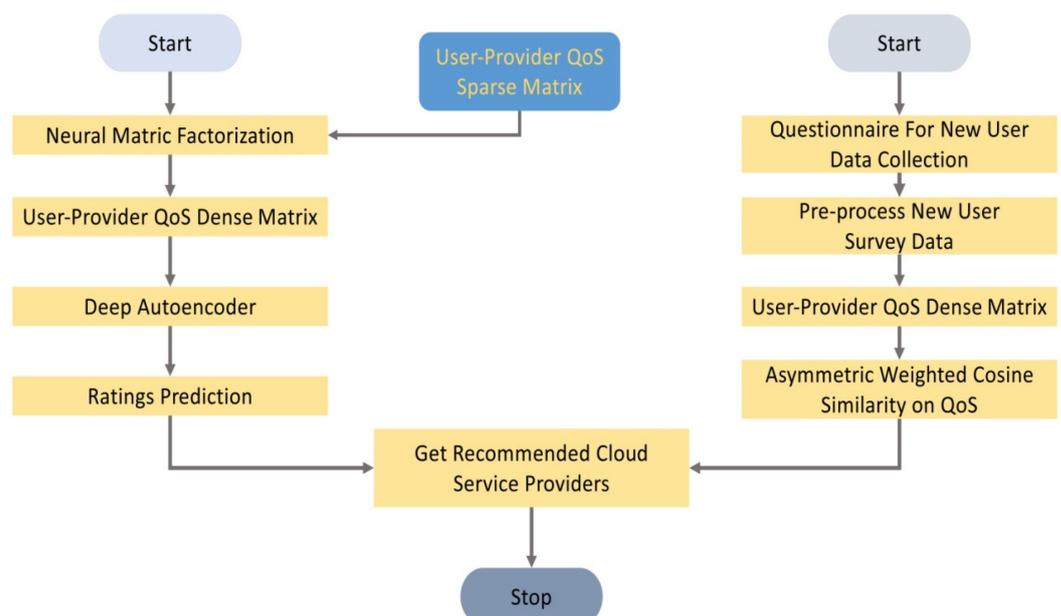


Figure 2. Flowchart for existing and new users.

3.1. Preliminary

We define the QoS values of our cloud users and cloud providers. First, the cloud user set is denoted as $U = \{u_1, u_2, u_3, \dots, u_m\}$ and user i is represented by u_i where $i = 1, 2, 3, \dots, m$. The QoS attribute set is represented by $Q = \{q_1, q_2, q_3, \dots, q_n\}$, and QoS j is represented as q_j where $j = 1, 2, 3, \dots, n$. $R = r_{ij} \in R^{m \times n}$ is the rating matrix, where r_{ij} represents the rate that cloud user u_i gives about q_j . Moreover, b_u denotes the user bias, b_q is the quality attributes bias, b_i represents the bias term for user i , and b_j is the bias term for QoS j . Additionally, as cloud users do not rate every QoS attribute, the matrices R are often sparse, and most entries are missing. In our design, we handle sparse data matrices using NeuMF to make them denser and perform DL with a deep autoencoder.

3.2. Data Collection for New Users

For requests from new users or existing customers with no personal data at our hand, a Google Survey with the questionnaire is adopted to collect preference data, which are exported to a CSV file for further analysis. This questionnaire is designed based on the questions retrieved from many research papers [32–45]. A 5-point Likert scale is used for rating each question as follows: (1) Not Important, (2) Less Important, (3) Neutral, (4) Important, and (5) Extremely Important. Demographic features are concluded to determine the distribution of types of new cloud users. Furthermore, the QoS sets with the same attributes for new cloud users are created from collected preference data. Based on the QoS rates provided by new cloud users, we can work on their preferences and predict suitable recommendations of service providers. The cited papers have already validated the survey questions, and the details of this questionnaire are introduced in Appendix A.

3.3. Neural Network Matrix Factorization

The NeuMF is commonly used in recommendation systems, particularly for rating prediction tasks. It models the connections between users and objects in a rating matrix by combining the power of matrix factorization and neural networks. NeuMF uses underlying latent features that influence user ratings. Low-dimensional matrices are used to predict the missing values in the original scanty matrices, which tremendously helps. The goal of matrix factorization is to divide the rating matrix R into two lower-rank matrices, the user matrix U and the QoS matrix Q . These matrices record the latent qualities or factors affecting the ratings. An approximation of the rating is given by the dot product of the latent representations of the users and the objects. The use of NeuMF can capture nonlinear interactions between the user and object embedding, which can lead to more accurate predictions. The following is an expression for the factorization:

$$R \approx U * Q \quad (1)$$

where U is an $m \times k$ matrix, Q is an $k \times n$ matrix, and k is the dimensionality of the latent space. Row i of U represents the latent representation of user i , and column j of Q represents the latent representation of QoS attributes j . Here, $*$ represents the dot product between two matrices.

Figure 3 indicates that cloud users and QoS features are collections of matrix communications. The matrix factorization also adopts bias terms, which may excel in model performance. By incorporating bias terms, calculating the error rate with binary cross-entropy, and introducing regularization, the NeuMF model can capture user and object biases, predict ratings within the desired range, manage overfitting, and enhance recommendation performance [46].

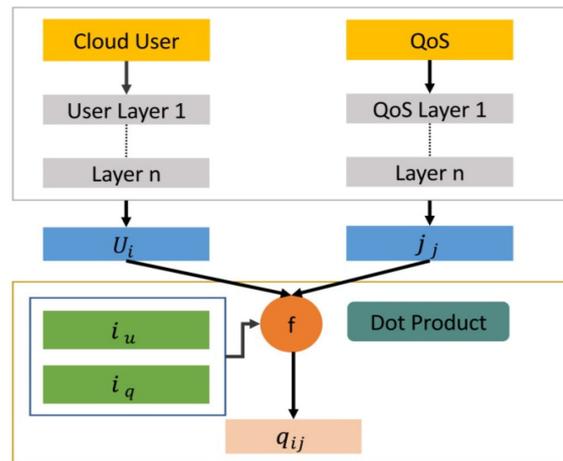


Figure 3. NeuMF.

The fundamental matrix factorization approach is expanded with biased terms for customers and cloud service providers in NeuMF. These biases can capture user and object choices that are not influenced by the latent factors. The rating matrix is referred to as R in the context of cloud-service recommendations, where each entry r_{ij} reflects the rating provided by the user i for QoS attribute j . The purpose of this matrix is to predict the missing ratings.

$$\hat{r}_{ij} = \mu + b_i + b_j + U_i + Q_j \tag{2}$$

where \hat{r}_{ij} represents the predicted rating for user i for QoS j , μ is the global average rating, b_i is the bias term for user i , b_j is the bias term for QoS attributes j , U_i represents the latent representation of user i , and Q_j represents the latent representation of QoS j .

We used weight matrices $W_u (m \times k)$ and $W_q (k \times n)$ to denote the user and object latent factor matrices. These weight matrices enable learning and updating parameters U and Q during optimization. The latent factor vectors of users and objects are modified by multiplying them by their respective weight matrices as follows:

$$U_i = W_{u,i} * u_i \tag{3}$$

$$Q_j = q_j * W_{q,j} \tag{4}$$

During the optimization process, the weight matrices $W_{u,i}$ and $W_{q,j}$ are updated by adjusting their parameters based on the gradient of the loss function. The specific updating rules, such as gradient descent, depend on the chosen optimization algorithm. The weight matrices control the contribution of each latent factor for each user and object, allowing the model to learn the importance of different factors for prediction. After that, we can better train the matrix factorization model by minimizing the mean squared error between the predicted and actual rating scores [47]. The objective function is defined as

$$Loss = \sum_{i,j} (r_{ij} * \log(\hat{r}_{ij}) + (1 - r_{ij}) * \log(1 - \hat{r}_{ij})) + \lambda * (\|U\|^2 + \|Q\|^2 + \|b_u\|^2 + \|b_q\|^2) \tag{5}$$

where the regularization rate is λ . The regularizing term $\lambda * (\|U\|^2 + \|Q\|^2 + \|b_u\|^2 + \|b_q\|^2)$ penalizes the scale of the parameters to prevent overfitting. Stochastic gradient descent or Adam, two optimization algorithms, can be used to learn the model parameters [47].

3.4. Deep Autoencoder

A deep autoencoder is a form of the deep neural network whose input and output vectors have the same dimensions. It is frequently used in hidden layers to learn the representation of something very close to the original data. A deep autoencoder, as shown in Figure 4, with three hidden layers representing the modified feature, an optimal dropout,

and an output layer identical to the input layer for reconstruction, is selected in our framework. The input layer represents the input feature vectors. The dropout is considered to reduce overfitting.

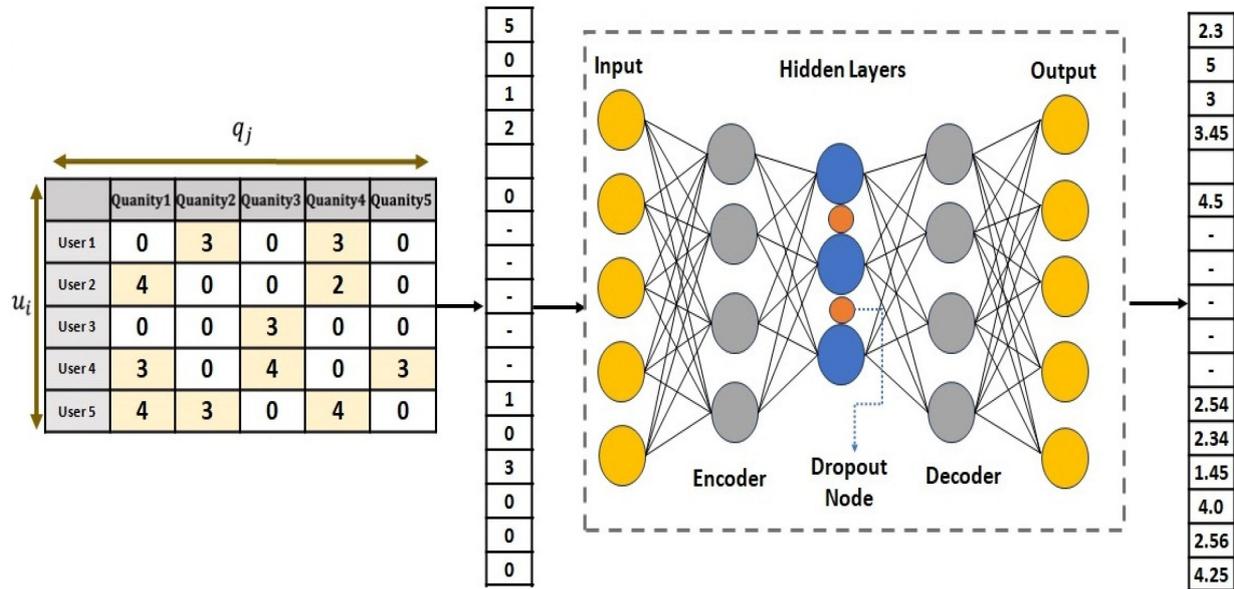


Figure 4. The deep autoencoder for rating prediction.

The encoder unit of the deep autoencoder maps input data, i.e., the NeuMF output, into a lower-dimensional space as $R^{m \times n} \rightarrow R^d$. A decoder unit that maps input data from a lower-dimensional space d to the realm of the rating matrix as $R^d \rightarrow R^{m \times n}$ [48]. The encoder $f(u_i, \Phi_1)$ produces latent representations from the row of rating matrix R (vector u_i) as follows

$$Encoding : R = f(u_i, \Phi_1) = f(u_i W_h + b_h) \tag{6}$$

where $\Phi_1 = \{W_h, b_h\}$. $W_h \in R^{n \times h}$ and $b_h \in R^h$ are the user-encoding layers' weights and biases, respectively, with dimensions h and d representing the amount of user hidden layer units and the amount of user and QoS parameters, and f being the activation function [48].

The decoder unit reconstructs the original row vector using encoded representation q_j . A reverse operation of the encoder is performed in the decoder component, which comes after the bottleneck layer as follows

$$Decoding : \hat{R} = f(q_j, \Phi_2) = f(q_j W_h^q + b_h^q) \tag{7}$$

where $\Phi_2 = \{W_h^q, b_h^q\}$. $W_h^q \in R^{m \times n}$ and $b_h^q \in R^n$ are the weight and bias of the object hidden layer, respectively, with the same number of hidden units h [48]. \hat{R} represents the reconstructed row vector, and f is the activation function.

$$argmin \sum_{i=1}^m \sum_{j=1}^n \|r_{ij} - h(r_{ij}, \theta)\|^2 + \frac{\lambda}{2} \left(\|W_h\|^2 + \|W_h^q\|^2 + \|b_h\|^2 + \|b_h^q\|^2 \right) \tag{8}$$

A deep autoencoder typically experiences overfitting owing to a large number of parameters. To control the overfitting of this model using two regularized terms of the loss function, we include a parameter λ with dropout to avoid this problem.

3.5. Rating Similarities

After making predictions, the most suitable cloud services should be suggested to the cloud customers. Therefore, the cloud services with top degrees of preference from other

users similar to the arriving users are better candidates based on a CF algorithm. Cosine similarity measures how similar two vectors are in a two-dimensional space. Figure 5 shows the distance between two users and their cosine distance, which ranges from 0 to 1. The $\cos \theta$ can calculate the cosine distance and cosine similarity between two vectors (data points), U_1 and U_2 , on the plane with two axes X and Y . The distance between users is obtained by computing their cosine similarity using their QoS metrics. A value close to 1 indicates a significant correlation between the two variables. If the value is near 0, these independent variables are not correlated. An asymmetric weight refers to distinct and potentially customized feature weighting assignments for each vector, whereas symmetric weight refers to equal or identical feature weighting results for both vectors. The similarity between two users who have evaluated the same QoS features can be determined using weighted cosine similarity.

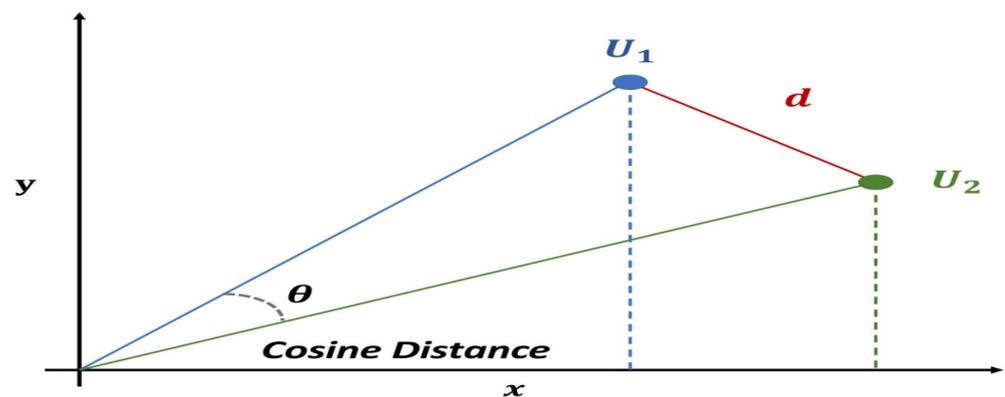


Figure 5. Cosine similarity between two users (objects).

To compute the similarities between users, we choose the algorithm of AWCS [15]. According to Filali and Yagoubi [49], their approach is beneficial for determining how similar two users are. They suggest that precise similarity measurements might not be obtained if user ratings are evaluated exclusively based on particular criteria or fundamental demographic information. Thus, studying a more comprehensive range of user attributes becomes necessary to describe people with different backgrounds accurately. We need to standardize features so that they reliably record user preferences and make data interpretation easier. A detailed evaluation of various aspects to achieve this regularity is also required. These factors include usage history, behavioral tendencies, contextual data, and demographic information. Another complication to the standardization process is guaranteeing the representativeness and trustworthiness of the accompanying data. Moreover, many computing techniques are available to determine user weights of the AWCS method. This flexibility allows different weights to be assigned to shared features between vectors.

Figure 6 illustrates that the auxiliary data from users and QoS features can identify asymmetric similarities between users and give personalized recommendations. These auxiliary data could include various QoS metrics such as response time, trust, availability, price, etc., which are essential for evaluating cloud service providers. The AWCS considers the different importance of each QoS in formatting the similarity between users. This weighting ensures that more significant QoS factors contribute more to the similarity calculation than less important ones. Each user receives a personalized QoS score calculated based on the AWCS computations. The approach determines which cloud service providers are the most suitable for each customer based on their QoS score. With each customer's unique requirements, these providers can deliver the preferred QoS features. Generally, the consumer is recommended a fixed number of highly rated providers, e.g., the top N providers. The following steps were performed:

1. All ratings each user provides for QoS attributes would be averaged to form the vector for individual users.
2. Each cloud user assigns every QoS attribute a weight value, reflecting the user’s preference for various dimensions. We assign weights based on ratings in the Likert scale. Higher weights indicate stronger preferences and vice versa.
3. A matrix with weights is created. The columns represent the QoS features, whereas the rows represent users.
4. The weighted cosine similarity of the two user vectors is calculated by dividing their dot product by the product of their norms.

$$AWCS(u_1, u_2) = \frac{\sum_i W_i u_{1i} u_{2i}}{\sqrt{\sum_i (W_i u_{1i})^2} \sqrt{\sum_i (W_i u_{2i})^2}} \tag{9}$$

where u_{1i} and u_{2i} are the user vectors, W_i is the weight vector, and i indexes the features of the QoS set.

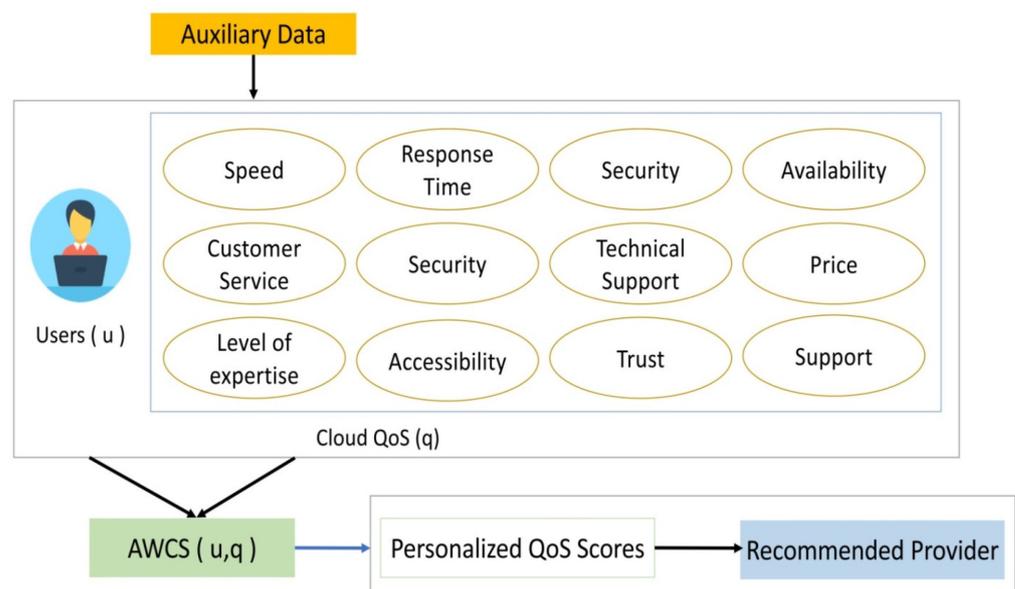


Figure 6. AWCS.

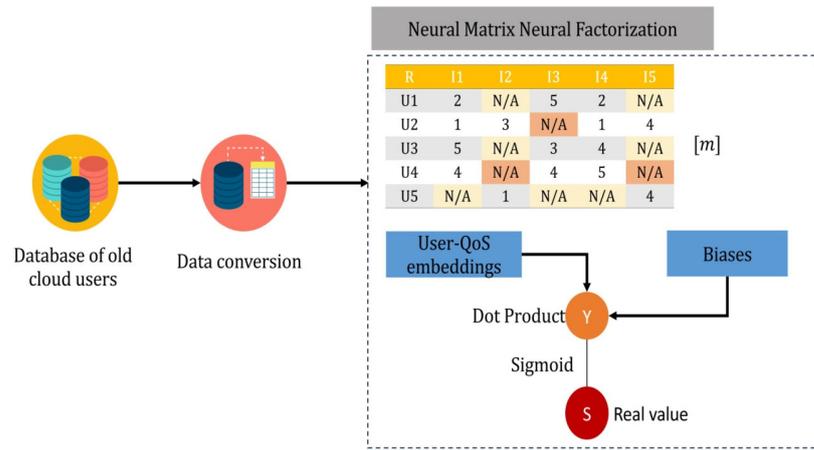
3.6. Proposed Design of Hybrid Approach

Our hybrid approach design is divided into three parts, as shown in Figure 7a–c. We deal with the old and new cloud users individually through DL processes and survey questionnaires.

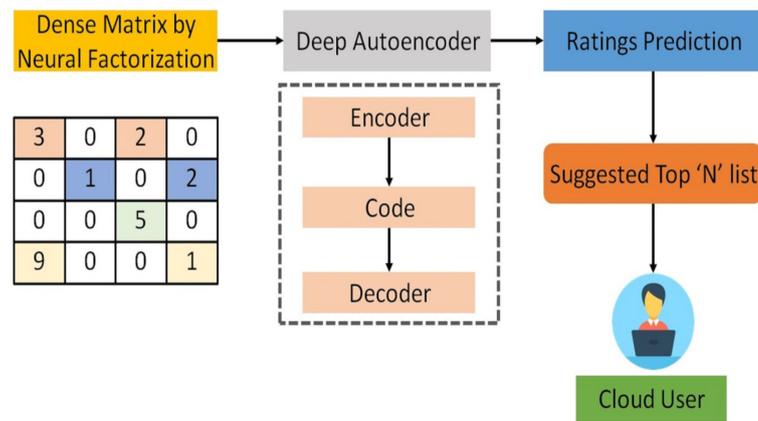
We utilize stored feedback results from existing users about different cloud providers’ services and convert them into matrices. These matrices are subjected to a NeuMF algorithm to be converted from sparse to dense, as shown in Figure 7a.

The NeuMF outputs are fed into the well-trained deep autoencoder, and we can generate better rating predictions. Then, top N cloud service providers are offered to the arriving cloud user for reference based on the predicted rates, as demonstrated in Figure 7b. The deep autoencoder is selected because it often develops a more compact and efficient representation of user–provider interactions, which is perfect for recommendation.

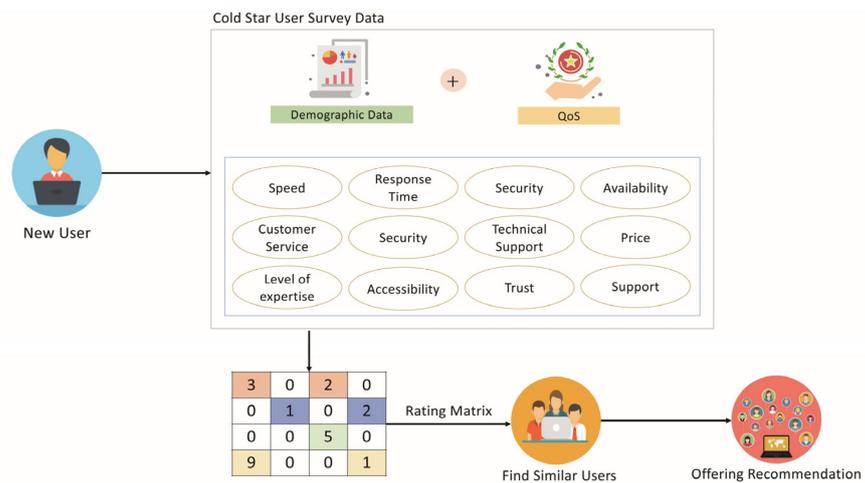
Figure 7c indicates the service recommendation to the new cloud user, where data are collected through a customer survey. Once the data are collected, they are compared with existing user profiles, and the weighted similarities between existing and new users are calculated. Finally, a recommendation list of personalized service providers is created.



(a)



(b)



(c)

Figure 7. (a) Existing cloud users’ sparse data are handled with NeuMF. (b) Deep autoencoder provides better prediction and top list is generated. (c) Recommendation system for new cloud users.

4. Evaluation

4.1. Data Description

The proposed design is tested using the Cloud Armor dataset [50]. This dataset comprises evaluations by cloud users for twelve QoS parameters: Availability, Security,

Response time, Accessibility, Price, Speed, Storage, Ease use, Technical support, Customer service, Level of expertise, Trust. They are provided to the cloud service providers together with timestamp data associated with each evaluation. All the features are rated using a 5-point scale varying from 1 to 5; the higher, the better. The following process proceeds to clean the data:

1. The timestamp columns are removed, as they are unnecessary for our research.
2. All entries with a trust value of less than four are removed. In this research, we believe the trust value should be equal to or greater than four so that the user can obtain reliable cloud services from recommended providers.
3. The cloud service providers with less than 100 customers have also been removed.
4. All the blank values in QoS are filled up with 0.

To recommend service providers, we consider the ratings provided for the QoS factors or parameters and then determine the average of all predicted ratings using the proposed NFDA. The providers of the top $N = 10$ averages from the predicted QoS are recommended to the coming cloud user. We used 70% of the data from the Cloud Armor dataset as training data, 10% as validation data, and 20% as testing data. We used the Windows 10 operating system with an Intel®Core™ i5-8265U CPU @ 1.60 GHz 1.80 GHz processor, 4.00 GB RAM, and a 64-bit operating system. With Python 3.9, the proposed approach is implemented using the TensorFlow package Keras and Anaconda 3. As for the new user preferences, the necessary data are collected from the carefully designed questionnaire. Then, we compare user similarities using AWCS and recommend three cloud service providers according to the results deduced from survey information.

4.2. Experimental Results and Comparison

In NeuMF, model training is split into two sections: user and QoS embedding. Each representative vector of users and QoSs passes through the input layer, embedding layer, and dense layer with 128 neurons as well as two hidden layers with ReLU activation for 50 epochs and ten steps per epoch. We also adopt the Adam optimizer with a 0.001 learning rate for the neural matrix data. The activation function is set to SELU with 150 epochs, 128 batch sizes, and a regularization term for the loss functions. Dropout is another deep-learning regularization technique. This model uses dropout layers to avoid overfitting. All parameters are summarized in the following Table 1.

Table 1. Parameters for experiments.

Parameters	Values
Embedded layers/hidden layers	128 neurons with two hidden layers
Activation	ReLu and SELU
Epochs	50–150
Steps per epoch	10
Optimizer	Adam
Leaning rate	0.001
Dropout layer value	0.5

Figure 8 indicates that the root-mean-square error (RMSE) and mean absolute error (MAE) with a learning rate of 0.001 is the best, i.e., it incurs the minimum error. The precision in predicting the best ten service providers is shown in Figure 9.

In Figure 10, we compare our research results with other QoS prediction models based on precision. In addition, Tables 2 and 3 list all the comparison results with the different methods. Our approach has shown improvement in terms of precision, RMSE, and MAE.

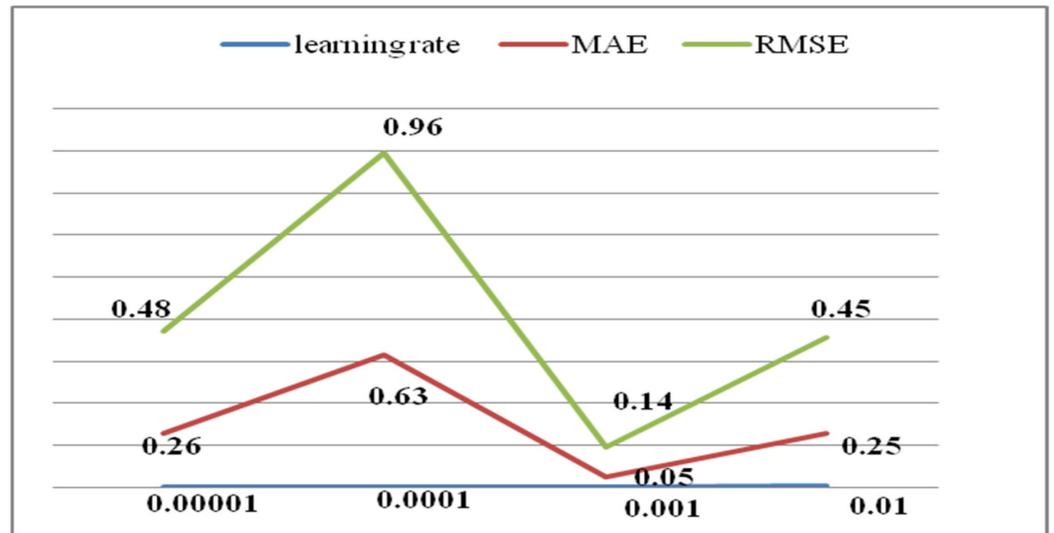


Figure 8. Performance of the proposed model’s leaning rates on the Cloud Armor dataset.

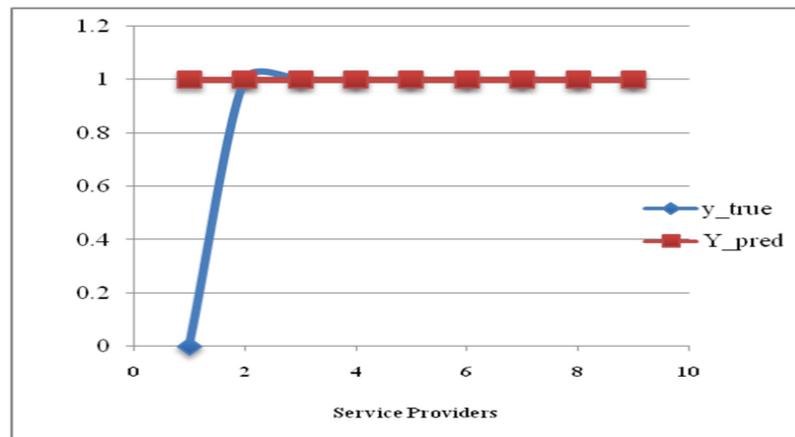


Figure 9. Precision of actual and predicted ratings.

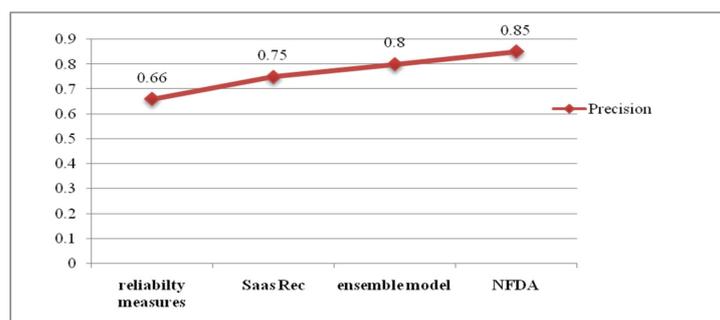


Figure 10. Precision comparison with other methods.

Table 2. Precision comparison using Cloud Armor dataset [50].

Methods	Precision
Nassiri et al. [51] (reliability measurements)	0.66
Afify et al. [52] (SaaS recommender)	0.75
Taneja and Kaur [53] (ensemble model)	0.8
Our proposed NFDA	0.85

Table 3. RMSE and MAE comparison using Cloud Armor dataset [50].

Methods	RMSE	MAE
TRUST_CV_20BIN [54]	0.45	0.44
Regression Trust [54]	0.59	0.41
Our proposed NFDA	0.14	0.05

For existing cloud service customers, the top $N = 10$ service providers recommended by our model are essential references for making intelligent decisions. Figure 11 demonstrates the top 10 cloud service providers suggested on average by the proposed method based on the customer and rating data in the Cloud Armor dataset. Moreover, a Google Form questionnaire link was created and hosted online in July 2023 for new cloud service users or existing customers with no personal data at our hand. The respondents were approached using snowball sampling methods to send e-mails to our students, colleagues, and friends for initial data collection. In general, it is easier to reach potential users interested in or without direct experience with cloud services. In the e-mail, we explained the complete experiment in detail and also asked them to invite their qualified companions to attend this survey. They understood that a recommendation list of the top 3 cloud service providers would be provided according to the filled-in preferences in the questionnaire. We received 74 response results from Google Surveys, and four of them were identified as false or incomplete. Subsequently, we used these 70 valid responses for experimental and computational purposes. The survey results are presented in Table 4 in detail.

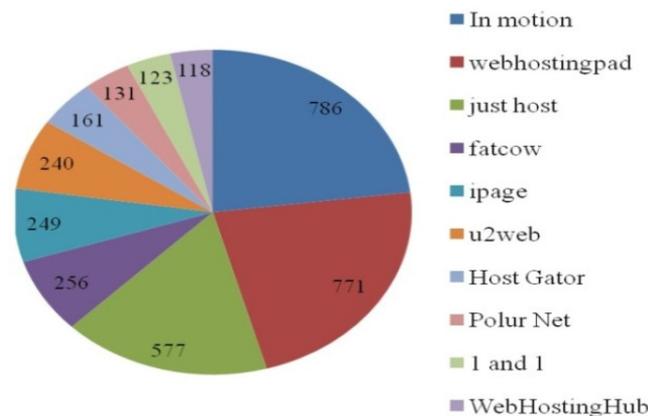


Figure 11. Top cloud providers in QoS.

Table 4. Summary results of the survey.

Personal Information	Age—19 and under (14.3%), 20–29 (38.5%), 30–39 (24.3%), 40–49 (20.0%), Others (2.9%) Gender—Male (51.4%), Female (47.1%), Others (1.5%) Preference language—English (77.1 %), Others (22.9%) Education level—Masters or equivalent (42.8%), Others (57.2%) Cloud usage—Rate themselves as average (42.8%) Employment status—Student (28.6%), IT professional (27.1%), Cloud developer (22.9%), Others (21.4%) Learn to secure data—More than 30 min (77.1%) Store data online—Daily (58.6%) Free storage—Free Storage (97.1%)
QoS Data	According to data collected from the questionnaire, most new cloud users prefer more than 4.5 ratings for cloud service providers. Most of the QoSs are considered extremely important.

As introduced above, the AWCS approach was chosen to assess how closely multiple features resembled each other. For a personalized recommendation list of cloud service

providers, the AWCS is used in our method to calculate the similarity between customers for greater accuracy. To evaluate the performance of our suggestion to new cloud service customers, feedback questionnaires were sent to the original respondents again via e-mail. We received replies from 57 users. Table 5 shows the summarized feedback results from new cloud users that demonstrate their satisfaction levels.

Table 5. Feedback from new cloud users regarding recommendation list.

1. Occupation	Student: 16, IT: 15, Cloud developer: 13, and Others: 13												
2. How does the recommendation list fulfill your personal needs and preferences?	<table border="1"> <caption>Satisfaction Levels for Recommendation List Fulfillment</caption> <thead> <tr> <th>Satisfaction Level</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Very Much</td> <td>33%</td> </tr> <tr> <td>Acceptable</td> <td>46%</td> </tr> <tr> <td>Neutral</td> <td>19%</td> </tr> <tr> <td>Not acceptable</td> <td>0%</td> </tr> <tr> <td>Not at all</td> <td>2%</td> </tr> </tbody> </table>	Satisfaction Level	Percentage	Very Much	33%	Acceptable	46%	Neutral	19%	Not acceptable	0%	Not at all	2%
Satisfaction Level	Percentage												
Very Much	33%												
Acceptable	46%												
Neutral	19%												
Not acceptable	0%												
Not at all	2%												
3. How are you satisfied with the provided recommendation?	<table border="1"> <caption>Satisfaction Levels with Provided Recommendation</caption> <thead> <tr> <th>Satisfaction Level</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Very Satisfied</td> <td>39%</td> </tr> <tr> <td>Satisfied</td> <td>39%</td> </tr> <tr> <td>Neutral</td> <td>19%</td> </tr> <tr> <td>Dissatisfied</td> <td>3%</td> </tr> <tr> <td>Very Dissatisfied</td> <td>0%</td> </tr> </tbody> </table>	Satisfaction Level	Percentage	Very Satisfied	39%	Satisfied	39%	Neutral	19%	Dissatisfied	3%	Very Dissatisfied	0%
Satisfaction Level	Percentage												
Very Satisfied	39%												
Satisfied	39%												
Neutral	19%												
Dissatisfied	3%												
Very Dissatisfied	0%												
4. Any further remarks?	Most new cloud users give importance to trust and price.												

Based on the feedback satisfaction levels, we believe that, on average, 78.5% of respondents are satisfied with the recommendation list even though we do not have their preference information for QoS or they are truly new cloud users.

4.3. Limitations

Even with the proposed techniques used in this study, this research still has some limitations. Firstly, the survey’s sample size is modest, primarily including computer-literate users, IT professionals, and cloud developers. This may restrict how the results can be applied to other populations/occupations. Secondly, the data of old users were collected from a public dataset; therefore, some sensitive information is not available for use. Thirdly, designing a better questionnaire to retrieve the customers’ preferences for QoS is challenging. Further validation tests may be necessary. Finally, the research only uses quantitative metrics; however, qualitative data are also a possible choice for measuring customers’ characteristics.

5. Conclusions

Hybrid model-based recommendation systems have attracted significant attention in personalized services. This paper proposes an appropriate technique of rating prediction for existing customers in recommendation systems. It combines NeuMF and a deep autoencoder in the hybrid framework. To predict ratings, our design examines the effects of combining user data, QoS data, a DL method (NeuMF), and an autoencoder. Integrating NeuMF and a deep autoencoder capable of handling complicated and high-dimensional data yields a robust and accurate recommendation result. The experiments based on the Cloud Armor database are evaluated according to three indexes, i.e., RMSE, MAE, and precision. The hybrid approach offering top $N = 10$ cloud service providers improves the

accuracy and efficiency of suggestions. For new cloud service users, we adopt a carefully designed questionnaire and AWCS to derive the recommendation list, which includes three service providers. A feedback questionnaire with personalized suggestions is used to assess the users’ satisfaction levels.

Our approach to providing the recommendation lists of cloud service providers is measured with an MAE of 0.05, an RMSE of 0.14, and a precision rate of 85%. These measurements show that our approach can handle the problems of data sparsity and cold start with relatively low errors, indicating it is successful in detecting the underlying patterns in the data. We also provide a satisfaction level of nearly 78.5% for new cloud service customers. In this study, both cloud service provider recommendations for existing and new customers demonstrate notable performance. To summarize, our research has provided valuable insights on how to handle data sparsity and cold start.

Author Contributions: Conceptualization, S.S.; methodology, S.S.; software, S.S.; formal analysis, S.S.; resources, C.C.-Y.K.; data curation, S.S. and C.C.-Y.K.; writing—original draft preparation, S.S.; writing—review and editing, C.C.-Y.K.; visualization, S.S. and C.C.-Y.K.; supervision, C.C.-Y.K.; funding acquisition, C.C.-Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding and the APC was funded by the Ministry of Science and Technology in Taiwan, ROC, under project MOST 111-2410-H-A49-033-MY3.

Data Availability Statement: Cloud Armor dataset used in the study is openly available in [50] and the survey data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Questionnaire for New Cloud Users

Appendix A.1. Section 1: Personal Information

1. In what age group are you in?

<input type="radio"/> 19 and under	<input type="radio"/> 20–29	<input type="radio"/> 30–39	<input type="radio"/> 40–49	<input type="radio"/> 50–59	<input type="radio"/> 60+
------------------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------	---------------------------

2. Gender:

<input type="radio"/> Male	<input type="radio"/> Female
----------------------------	------------------------------

3. What is your preferred language?

<input type="radio"/> English	<input type="radio"/> Spanish	<input type="radio"/> Chinese	<input type="radio"/> French	<input type="radio"/> German
<input type="radio"/> If other, please specify:				

4. Please write your residing country: _____

5. What best describes your education level?

<input type="radio"/> High School	<input type="radio"/> College–University	<input type="radio"/> Master’s or equivalent	<input type="radio"/> PhD
-----------------------------------	--	--	---------------------------

6. What best describes your employment status?

<input type="radio"/> Student	<input type="radio"/> IT Professional	<input type="radio"/> Manager	<input type="radio"/> Cloud Developer	<input type="radio"/> Academic Faculty
<input type="radio"/> Other (please specify)				

7. How would you rate yourself with respect to your knowledge of cloud computing?

1 (very poor)	2	3	4	5 (excellent)
---------------	---	---	---	---------------

8. Would you accept it if the cloud provider provided a significant amount of free online storage?

<input type="radio"/> Yes	<input type="radio"/> No
---------------------------	--------------------------

9. What amount of time would you be willing to invest to learn how to secure your personal data online?

<input type="radio"/> 0 min	<input type="radio"/> 0–30 min	<input type="radio"/> 30 min–1 h	<input type="radio"/> 1–2 h	<input type="radio"/> More than 2 h
-----------------------------	--------------------------------	----------------------------------	-----------------------------	-------------------------------------

10. How often do you use the Internet to store data of any type?

<input type="radio"/> Never	<input type="radio"/> Daily	<input type="radio"/> A few times a week	<input type="radio"/> A few times a month	<input type="radio"/> A few times a year
-----------------------------	-----------------------------	--	---	--

Appendix A.2. Section 2: Cloud QoS Questions

References	Cloud QoS
	Availability
	In cloud services, cloud outages, and multi-location backup are the main concerns for ensuring data availability.
[32–35]	Cloud services provide authorized users with access to data, software, and hardware.
	The cloud service has robust availability measures to handle and mitigate Distributed Denial-of-Service (DDoS) attacks and flooding attacks, ensuring that the data remain accessible and available.
	Security
	Cloud service providers do not access or distribute your data without your written permission.
[36–38]	Data in cloud computing is secured against unauthorized alterations.
	Cloud computing is always available based on user credentials.
	Security is the main concern for online data storage.
	Service response
	The cloud service provider’s staff is truly interested in resolving user issues.
[37]	The cloud service provider’s staff keeps users updated on the status of problem resolution.
	The cloud service provider’s staff is prompt in offering services to users.
	Accessibility
[39,40]	Provides 24 × 7 worldwide accessibility to cloud resources.
	Cloud services performance gives intense access to the system.
	Price/Cost saving
	The price of the cloud service is less expensive than buying a laptop or PC software.
[39]	The price of the cloud service is affordable, especially when considering the maintenance costs I would have to pay if I used my own resources.
	The price of the cloud service is costly in comparison to the expense of employing open-source or Internet apps.
	Speed
	The cloud service’s speed is adequate for backup and storage purposes.
[39]	It is exactly the same as uploading or downloading files to any other data.
	The cloud service’s speed is adequate for my everyday tasks.
	Storage
	The cloud service provides safe solutions for data storage, reducing concerns about the possibility of losing control over data in a data warehouse.
[32,41,42]	The cloud service offers data area server capabilities, ensuring that data storage is optimized for performance and efficiency by placing it closer to the relevant users or computing resources.
	The cloud service employs robust authentication mechanisms to control access to data storage, ensuring that only authorized users can access and interact with the stored data.
	Ease use

References	Cloud QoS
	Using cloud service facilities is simple to understand.
[39,43]	The cloud service provides a seamless user experience by offering assistance and support in troubleshooting any issues that may arise during cloud usage.
	The cloud service offers user-friendly resources and documentation to facilitate the study and understanding of how to use cloud computing apps effectively.
	Technical support
	The cloud service offers clear and comprehensive technical support, ensuring transparency and clarity regarding the use limitations and the terms and conditions of the contracted services.
[40,44]	The cloud service provides free technical support to assist users with any inquiries or issues they may encounter while utilizing the cloud services.
	Cloud services incorporate advanced tools and backup policies that align with development needs and business requirements, guaranteeing a reliable and efficient operating environment.
	Customer service
	When compared to in-house IT systems, you are satisfied with cloud services.
[37,40]	Cloud services' support solves users' problems efficiently.
	The cloud service is dedicated to providing transparency and accuracy by promptly fixing identified errors, ensuring a reliable and trustworthy user experience.
	Level of expertise/IT knowledge
	The cloud service team is highly skilled and knowledgeable in IT, being well-versed and experienced in working with various cloud computing services.
[36]	The cloud service team is familiar with the uniqueness of cloud computing services that are significant to the company.
	When evaluating cloud services, they can often be assessed by reviewing informational brochures and materials provided by the cloud service provider.
	Trust
	Cloud services are stable and trustworthy.
[36,45]	Cloud services are capable of conquering any technological challenge.
	Cloud services defend their subscribers' rights.

References

- Ghuman, S. Cloud Computing—A Study of Infrastructure as a Service. *Int. J. Comput. Sci. Mob. Comput.* **2015**, *12*, 435–440. Available online: <https://api.semanticscholar.org/CorpusID:212451750> (accessed on 10 December 2022).
- Kelkar, S. Challenges and Opportunities with Cloud Computing. *Int. J. Innov. Res. Comput. Commun. Eng.* **2015**, *3*, 2719–2724.
- Ruby, J.; Aisha, B. Cold Start Problem in Recommending Cloud Services: A Survey. *J. Adv. Res. Dyn. Control Syst.* **2017**, *9*, Sp-16. Available online: https://www.academia.edu/39345675/COLD_START_PROBLEM_IN_RECOMMENDING_CLOUD_SERVICES_A_SURVEY (accessed on 3 January 2023).
- Mohamed, M.H.; Khafagy, M.; Ibrahim, M.H. Recommender Systems Challenges and Solutions Survey. In Proceedings of the IEEE International Conference on Innovative Trends in Computer Engineering, Aswan, Egypt, 2–4 February 2019; pp. 149–155. [CrossRef]
- Natarajan, S.; Vairavasundaram, S.; Natarajan, S.R.; Gandomi, A.H. Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data. *Expert Syst. Appl.* **2020**, *149*, 113248. [CrossRef]
- Hsieh, M.Y.; Weng, T.H.; Li, K.C. A Keyword-Aware Recommender System Using Implicit Feedback on Hadoop. *J. Parallel Distrib. Comput.* **2018**, *116*, 63–73. [CrossRef]
- Berisha, F.; Bytyçi, E. Addressing Cold Start in Recommender Systems with Neural Networks: A Literature Survey. *Int. J. Comput. Appl.* **2023**, *45*, 485–496. [CrossRef]
- Idrissi, N.; Zellou, A. A Systematic Literature Review of Sparsity Issues in Recommender Systems. *Soc. Netw. Anal. Min.* **2020**, *10*, 15. [CrossRef]
- Kuchaiev, O.; Ginsburg, B. Training Deep Auto Encoders for Collaborative Filtering. *arXiv* **2017**, arXiv:1708.01715.

10. Dziugaite, G.K.; Roy, D.M. Neural Network Matrix Factorization. In Proceedings of the 6th International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; Available online: <https://arxiv.org/abs/1511.06443> (accessed on 8 January 2023).
11. Zheng, X.; Xu, L.D.; Chai, S. QoS Recommendation in Cloud Services. *IEEE Access* **2017**, *5*, 5171–5177. [[CrossRef](#)]
12. Djiroun, R.; Guessoum, M.A.; Boukhalifa, K.; Benkhalifa, E. A Novel Cloud Services Recommendation System Based on Automatic Learning Techniques. In Proceedings of the International Conference on New Trends in Computing Sciences 2017, Amman, Jordan, 11–13 October 2017; pp. 42–49. [[CrossRef](#)]
13. Mohamed, M.H.; Khafagy, M.H.; Elbeh, H.; Abdalla, A.M. Sparsity and Cold Start Recommendation System Challenges Solved by Hybrid Feedback. *Int. J. Eng. Res. Technol.* **2019**, *12*, 2735–2742.
14. Han, J.; Kamber, M.; Pei, J. *Getting to Know Your Data, Data Mining*, 3rd ed.; The Morgan Kaufmann Series in Data Management Systems; Morgan Kaufmann: Burlington, MA, USA, 2012; pp. 39–82. [[CrossRef](#)]
15. Mishra, S.; Panda, S.K. Asymmetrically Weighted Cosine Similarity Measure for Recommendation Systems. In Proceedings of the International Conference of Advances in Distributed Computing and Machine Learning, Warangal, India, 15–16 January 2022; pp. 489–500.
16. Fu, Y.; Ding, D.; Ahmed, S. Using Nearest Graph QoS Prediction Method for Service Recommendation in the Cloud. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8680758. [[CrossRef](#)]
17. Khan, Z.Y.; Niu, Z.; Sandiwarno, S.; Prince, R. Deep Learning Techniques for Rating Prediction: A Survey of the State-of-the-Art. *Artif. Intell. Rev.* **2021**, *54*, 95–135. [[CrossRef](#)]
18. Bougteb, Y.; Ouhbi, B.; Frikh, B.; Zemmouri, E. A Deep Autoencoder-Based Hybrid Recommender System. *Int. J. Mob. Comput. Multimed. Commun.* **2022**, *13*, 1–19. [[CrossRef](#)]
19. Sahu, P.; Raghavan, S.; Chandrasekaran, K. Ensemble Deep Neural Network Based Quality of Service Prediction for Cloud Service Recommendation. *Int. J. Neurocomput.* **2021**, *465*, 476–489. [[CrossRef](#)]
20. Xiong, R.; Wang, J.; Zhang, N.; Ma, Y. Deep Hybrid Collaborative Filtering for Web Service Recommendation. *Expert Syst. Appl.* **2018**, *110*, 191–205. [[CrossRef](#)]
21. Chen, J.; Mao, C.; Song, W.W. QoS Prediction for Web Services in Cloud Environments based on Swarm Intelligence Search. *Knowl.-Based Syst.* **2023**, *259*, 110081. [[CrossRef](#)]
22. Nabli, H.; Ben Djemaa, R.; Amous Ben Amor, I. Improved Clustering-based Hybrid Recommendation System to Offer Personalized Cloud Services. *Clust. Comput.* **2023**, 1–30. [[CrossRef](#)]
23. Zhu, X.; Jing, X.; Wu, D.; He, Z.; Cao, J.; Yue, D.; Wang, L. Similarity Maintaining Privacy Preservation and Location-Aware Low-rank Matrix Factorization for QoS Prediction Based Web Service Recommendation. *IEEE Trans. Serv. Comput.* **2021**, *14*, 889–902. [[CrossRef](#)]
24. Zhang, Y.; Zhang, P.; Luo, Y.; Ji, L. Towards Efficient, Credible and Privacy-Preserving Service QoS Prediction in Unreliable Mobile Edge Environments. In Proceedings of the International Symposium on Reliable Distributed Systems 2020, Shanghai, China, 21–24 September 2020; pp. 309–318. [[CrossRef](#)]
25. Patro, S.G.; Mishra, B.K.; Panda, S.K.; Kumar, R.; Long, H.V.; Taniar, D. Cold Start Aware Hybrid Recommender System Approach for E-commerce Users. *Soft Comput.* **2023**, *27*, 2071–2091. [[CrossRef](#)]
26. Chen, X.; Li, B.; Wang, J.; Zhao, Y.; Xiong, Y. Integrating EMD with Multivariate LSTM for Time Series QoS Prediction. In Proceedings of the 2020 IEEE International Conference on Web Services (ICWS), Beijing, China, 19–23 October 2020; pp. 58–65.
27. Ding, L.; Liu, J.; Kang, G.; Xiao, Y.; Cao, B. Joint QoS Prediction for Web Services based on Deep Fusion of Features. *IEEE Trans. Netw. Serv. Manag.* **2024**; early access. [[CrossRef](#)]
28. Xiong, W.; Gu, Q.; Li, B.; Wu, Z.; Yuan, L. Collaborative Web Service QoS Prediction via Location-Aware Matrix Factorization and Unbalanced Distribution. *J. Internet Technol.* **2018**, *19*, 1063–1074.
29. Handri, K.E.; Idrissi, A. Parallelization of Topk Algorithm Through a New Hybrid Recommendation System for Big Data in Spark Cloud Computing Framework. *IEEE Syst. J.* **2021**, *15*, 4876–4886. [[CrossRef](#)]
30. Zhang, P.; Huang, W.; Chen, Y.; Zhou, M.; Al-Turki, Y. A Novel Deep-Learning-based QoS Prediction Model for Service Recommendation Utilizing Multi-Stage Multi-Scale Feature Fusion With Individual Evaluations. *IEEE Trans. Autom. Sci. Eng.* **2024**; early access. [[CrossRef](#)]
31. Chahal, D.; Mishra, M.; Palepu, S.; Singhal, R. Performance and Cost Comparison of Cloud Services for Deep Learning Workload. In Proceedings of the International Conference on Performance Engineering 2021, New York, NY, USA, 19–23 April 2021; pp. 49–55. [[CrossRef](#)]
32. Singh, S.; Jeong, Y.S.; Park, J.H. A Survey on Cloud Computing Security: Issues, Threats, and Solutions. *J. Netw. Comput. Appl.* **2016**, *75*, 200–222. [[CrossRef](#)]
33. Choi, J.; Choi, C.; Ko, B.; Choi, D.; Kim, P. Detecting Web Based DDoS Attack Using MapReduce Operations in Cloud Computing Environment. *J. Internet Serv. Inf. Secur.* **2013**, *3*, 28–37.
34. Feng, J.; Chen, Y.; Summerville, D.; Ku, W.S.; Su, Z. Enhancing Cloud Storage Security Against Stroll Back Attacks with a New Fair Multi-party Non-repudiation Protocol. In Proceedings of the IEEE Consumer Communications and Networking (CCNC) 2011, Las Vegas, NV, USA, 9–12 January 2011; pp. 521–522. [[CrossRef](#)]
35. Alghofaili, Y.; Albattah, A.; Alrajeh, N.; Rassam, M.A.; Al-rimy, B.A.S. Secure Cloud Infrastructure: A Survey on Issues, Current Solutions, and Open Challenges. *Appl. Sci.* **2021**, *11*, 9005. [[CrossRef](#)]

36. Wahsh, M.A.; Dhillon, J.S. An Investigation of Factors Affecting the Adoption of Cloud Computing for E-government Implementation. In Proceedings of the IEEE Student Conference on Research and Development, Kuala Lumpur, Malaysia, 13–14 December 2015; pp. 323–328.
37. Agarwal, R.; Dhingra, S. Factors Influencing Cloud Service Quality and Their Relationship with Customer Satisfaction and Loyalty. *Heliyon* **2023**, *9*, e15177. [[CrossRef](#)] [[PubMed](#)]
38. Horvath, A.S.; Agrawal, R. Trust in Cloud Computing. In Proceedings of the Southeast Conference 2015, Fort Lauderdale, FL, USA, 9–12 April 2015; pp. 1–8. [[CrossRef](#)]
39. Changchit, C. Students' Perceptions of Cloud Computing. *Issues Inf. Syst.* **2014**, *15*, 312–322.
40. Freitas, A.L.; Neto, M.M. Assessing the Service Quality in Software-as-a-Service from the Customers' Perspective: A Methodological Approach and Case of Use. *Prod. J.* **2017**, *27*, e20170020. [[CrossRef](#)]
41. Ahmed, A.S.; Hassan, R.; Ali, Z. Eliminate Spoofing Threat in IPv6 Tunnel. In Proceedings of the 8th International Conference on Information Science and Digital Content Technology 2012, Jeju Island, Republic of Korea, 26–28 June 2012; Volume 1, pp. 218–222.
42. Feng, J.; Chen, Y.; Ku, W.S.; Liu, P. Analysis of Integrity Vulnerabilities and a Non-repudiation Protocol for Cloud Data Storage Platforms. In Proceedings of the 39th International Conference in Parallel Processing Workshops, San Diego, CA, USA, 13–16 September 2010; pp. 251–258. [[CrossRef](#)]
43. Jahangiri, P.; Saberi, M.K.; Vakilmofrad, H. Development and Psychometric Evaluation of the Cloud Computing Acceptance Questionnaire for Academic Libraries. *J. Acad. Librariansh.* **2021**, *47*, 102395. [[CrossRef](#)]
44. Wright, K.B. Researching Internet-Based Populations: Advantages and Disadvantages of Online Survey Research, Online Questionnaire Authoring Software Packages and Web Survey Services. *J. Comput. Mediat. Commun.* **2005**, *10*, 1034. [[CrossRef](#)]
45. Amron, M.T.; Ibrahim, R.; Bakar, N.A.A.; Chuprat, S. Development and Validation of a Questionnaire to Measure the Acceptance of Cloud Computing in Public Sectors. *HCTL Open Int. J. Technol. Innov. Res.* **2019**, *2*, 85–95.
46. Anelli, V.W.; Bellogin, A.; Di Noia, T.; Pomo, C. Reenvisioning Collaborative Filtering vs Matrix Factorization. In Proceedings of the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021–1 October 2021. [[CrossRef](#)]
47. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. 21. Recommender Systems. In *Dive into Deep Learning*; Cambridge University Press: Cambridge, UK, 2023; Available online: https://d2l.ai/chapter_recommender-systems/mf.html (accessed on 9 November 2023).
48. Nisha, C.C.; Mohan, A. A Social Recommender System Using Deep Architecture and Network Embedding. *Appl. Intell.* **2019**, *49*, 1937–1953. [[CrossRef](#)]
49. Filali, F.Z.; Yagoubi, B. Classifying and Filtering Users by Similarity Measures for Trust Management in Cloud Environment. *Scalable Comput. Pract. Exp.* **2015**, *16*, 289–301. [[CrossRef](#)]
50. Noor, T.H.; Sheng, Q.Z.; Yao, L.; Dustdar, S.; Ngu, A.H.H. CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 367–380. [[CrossRef](#)]
51. Nassiri, R.; Latifi, F.; Mohsenzadeh, M.; Mostafaei, H. Cloud Service Recommendation Using Trust Data and Reliability Measurements. *SSRN Electronic J.* **2022**, 4064465. [[CrossRef](#)]
52. Afify, Y.M.; Moawad, I.F.; Badr, N.L.; Tolba, M.F. A Personalized Recommender System for SaaS Services. *Concurr. Comput. Pract. Exp.* **2016**, *29*, e3877. [[CrossRef](#)]
53. Taneja, H.; Kaur, S. An Ensemble Classification Model for Fake Feedback Detection Using Proposed Labeled Cloud Armor Dataset. *Comput. Electr. Eng.* **2021**, *93*, 107217. [[CrossRef](#)]
54. Hosseinnezhad, M.; Azgomi, M.A.; Dishabi, M.R.E. A Probabilistic Trust Model for Cloud Services using Bayesian Networks. *Soft Comput.* **2024**, *28*, 509–526. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.