

A study the web services using machine learning for personalized QoS are recommended

Amita Boral^{1*}, Dr. Kishan Kumar²

1 Research Scholar, Shri Krishna University, Chhatarpur, M.P.

ouriginal.sku@gmail.com

2 Professor, Shri Krishna University, Chhatarpur, M.P.

Abstract- This experiment addresses the challenge of predicting Quality of Service (QoS) values for web services using various linear regression methods, including fitrree (binary regression decision tree), fit ensemble with LSBoost and Bag, and lasso regression. By leveraging past user experiences, these models aim to recommend web services based on predicted QoS values with a focus on minimizing prediction errors. Data for the study will be collected from real-world web services across diverse locations. Four linear regression models will be implemented, each evaluating prediction accuracy. Initial findings suggest that fit ensemble with Bag and lasso regression perform better in predicting QoS values with minimal accuracy deviations, making them more effective for recommending web services. Experiments will be conducted using a QoS dataset derived from millions of real-world web service interactions, involving web services from 29 countries and users from 31 countries. Predictions will cover all users to recommend 1,292 unique web services. Recommendations will be tailored for continent-specific users, providing regionally optimized services. The study includes an in-depth analysis of datasets, focusing on the country-wise and continent-wise distribution of web services and users.

Keywords- QoS, Machine Learning, Linear Regression, Prediction Accuracy, Web Services.

INTRODUCTION

It is essential to place a strong emphasis on QoS characteristics such as response time, reliability, price, failure rate detection, high-performance computing, availability of web services, and high throughput in order to strengthen the efficiency of web services in environments that are completely heterogeneous or rapidly changing. This is necessary in order to persuade end-users to choose web services that are hosted in the cloud. A difficult research challenge is the intelligent & cost-effective transformation of cloud services from diverse sources into a popular service. In order to address the research problem, this section explains an approach.

Web service QoS is extensively used to reveal non-functional attributes. According to Zhang (2011), Zheng (2010), Albu (2013), and Ding (2014), quality of service is characterized by a collection of characteristics including availability, reputation, reaction time, throughput, and so on. Some QoS factors, such as response speed and user pragmatic accessibility, need user-

side calculation of their values (Zheng, 2010). Therefore, providers may not always be able to realistically acquire such QoS data. In addition, the user's situation (such as the user's continent or network condition) & unpredictable Internet environment can affect these QoS values.

To address this issue, one can employ computational prediction models like fitree, which is a linear regression method, or fitensemble, which is an ensemble of learners for classification and regression, along with parameters like LSBoost, Bag, or lasso. These models can then use the available user data to forecast the QoS-value. Predicting the tailored QoS aware online services was a clear way to recommend them based on the testing findings.

Typically, non-functional aspects of online services are described using QoS. Although QoS-values on the server side are better indicators of server capacity, QoS-values on the client side are more accurate measures of how users actually feel about the service's performance. Buyerya (2010), Bakshi (2009), Mell (2011), Zhang (2011), and Zheng (2010) cited prior research on web service QoS. The client-side web service QoS attributes that are commonly utilized include:

- **Response-time:** Duration between service users' requests and responses.
- **Throughput:** The typical rate at which messages are delivered over a communication channel is called throughput. Kilobits per second (kbps) is the usual notation for it.
- **Failure-probability:** the chance that an attempt to invoke a web service will be unsuccessful.

MOTIVATION

Quality of service (QoS) data is essential for planning, performance evaluation, & prediction, and it is obtained from a large array of domains via real-world service invocations by users. Data has become an invaluable tool for organizations in the age of the information economy. There isn't a perfect setting for extracting insights from data models that use complicated computational techniques. Therefore, a capacity environment that is elastic and available on demand is necessary. Due to the utilization of a huge QoS dataset, failure has become the standard in the current system. An essential architectural objective of the suggested architecture is the discovery of missing values & frequent predictions from web service precedent custom data. QoS concerns like as availability, response time, throughput, failure

rate detection, high-performance computing, and comprehensive diversity & rapid change must be addressed if users are to be satisfied. Finding an efficient way to transform various cloud services into intelligence and then distribute that intelligence to the appropriate resource while keeping costs low is no easy feat. In order to conduct experiments, QoS datasets that have some missing values are being utilized.

PREDICTION PROTOCOLS

Fitrtree

Fitrtree first calculates the number of branch nodes and then divides every node in the current layer in order to accommodate MaxNumSplits (Breiman, 1984). Optimal tree balance will be achieved. If there are more branch nodes than available splits, fitrtree:

- Determines which branch nodes must be divided in order to achieve MaxNumSplits branch nodes.
- Sort the nodes of the branches in descending order of impurity gains.
- Remove the fraction of the total number of branches that were successful.
- The decision tree that has been generated up to this point should be returned.

Fitensemble

According to Breiman (1996), it combines the data used to train weak learner models with the actual data. With the help of the cumulative predictions of its weak learners, an ensemble may anticipate its response to fresh data. Finding the sweet spot between speed & accuracy is what it takes to choose an ensemble's dimension. Predictions generated by robust ensembles may take more time, and there is a chance that these algorithms will be incorrect.

Consider beginning with a small ensemble size (dozens to hundreds of members), training it, and then testing its quality (test ensemble quality) to determine an optimal size. You can use techniques like classification & regression to increase the size of your group if it seems like you need it. Adding additional members to the ensemble does not enhance its quality.

Steps of the ensemble-based algorithm:

Data is represented by the matrix x . One observation is contained in each row, and one predictor variable is contained in each column.

- The number of observations in t , the reply vector, is equal to the number of rows in x .
- A string indicating the type of ensemble is the model.
- Numberens is the sum of all the elements of learners who are deficient in ens.
- Therefore, ens is equal to numberens multiplied by the numerical value of learners for the number of elements.
- Learners might represent a string describing a poor student's performance in class, or it could be a template or collection of templates for that student.

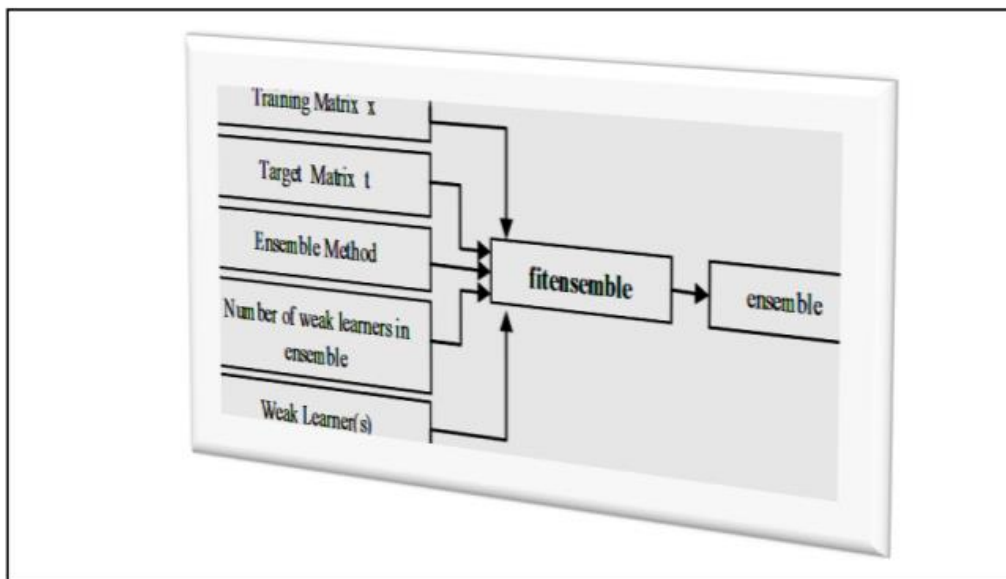


Figure 1: Steps to Create an Ensemble learner

LSBoost

LSBoost, which stands for "Least Squares Boosting," adjusts regression ensembles—as proposed by Breiman (1984), Hastie (2008), or Polikar (2006)—based on the difference between the observed response & total of all learners' predictions. To reduce the MSE, the ensemble fits here.

Bag

Regression Bagged ensemble method uses a combination of predictions from its weak learners and trained weak learner models to anticipate the ensemble's response (Breiman, 1996). Estimating the generalization error does not necessitate further cross-validation.

Lasso

One way to do linear regression is using the Lasso, which stands for Least Absolute Shrinkage and Selection Operator (Tibshirani, 1996).

The lasso algorithm fits a linear model given a collection of input dimensions $X_1, X_2 \dots X_n$ and a result dimension Y as:

$$\hat{Y} = b_0 + b_1 * X_1 + b_2 * X_2 + \dots + b_n * X_n \quad (1)$$

The decisive factor, it uses is:

$$\text{Minimize } \sum ((Y - \hat{Y})^2) \text{ subject to } \sum (|b_j|) \leq s \quad (2)$$

The parameter s serves as a tuning factor and is a boundary condition. If s is greater than zero, the output will be a modified version of the least square estimates. If s is less than zero, the output will be the standard multiple linear least squares regression of Y on $X_1, X_2 \dots X_n$.

Stepwise Forward Regression Algorithm:

- Start with coefficients $b_j = 0$.
- Estimate the predictor X_j mainly correlated with Y , and append it into the model.
- Obtain residuals $r = Y - \hat{Y}$.
- Keep adding the predictor most strongly connected with f to the model at each step until all predictors have been estimated.

MISSING QOS-VALUE PREDICTION

For real-world web service requests to the cloud, missing value prediction is a must-have duty. The prediction accuracy is significantly affected by the sparseness of the user-web services matrix, which often pertains to the reaction time & throughput of QoS attributes. Active user

prediction accuracy can be enhanced by forecasting missing values for the QoS matrix. For this reason, we employed MATLAB's regression-based missing value prediction models to increase the density of the matrix. To forecast the values, eight different kinds of QoS matrices were used, with three user groups being used: all, normalized, & continent-wise. For the purpose of missing value prediction, all user types will make use of these excessive preprocessing QoS datasets.

WEB SERVICE RECOMMENDATION

The problem of quality-of-service (QoS) aware online service recommendation has grown in importance due to the meteoric rise of cloud-based web services over the past decade. Due to the fact that gathering QoS-information about web services is a laborious and, at times, impracticable process, the values of certain services' QoS-value are often absent. Traditional methods relying on the user-item matrix static model to forecast the missing QoS-value are inefficient and prone to errors. Then, employing new models whose performance is significantly better than some classic forecasting models allowed for the proper and efficient suggestion of web services for users. The experimental comparison section details these newly used models.

CONTINENTS DEFINITIONS AND FEATURES

A $u \times w$ user-item matrix can be used to represent the relationship between online services & users in a recommender system where there are w web services and u users. Each row in this matrix reflects a quality of service (QoS) metric that user i has measured for web service j ; for example, r_{uij} stands for response time and t_{uij} for throughput. A group of users from the same continent—North America, South America, Europe, or Asia—is called a user continent. The term "web service continent" refers to a collection of interconnected online services that originate in one of the five major North American, South American, European, or Australian continents. Every single user and every single web service is located on one specific continent.

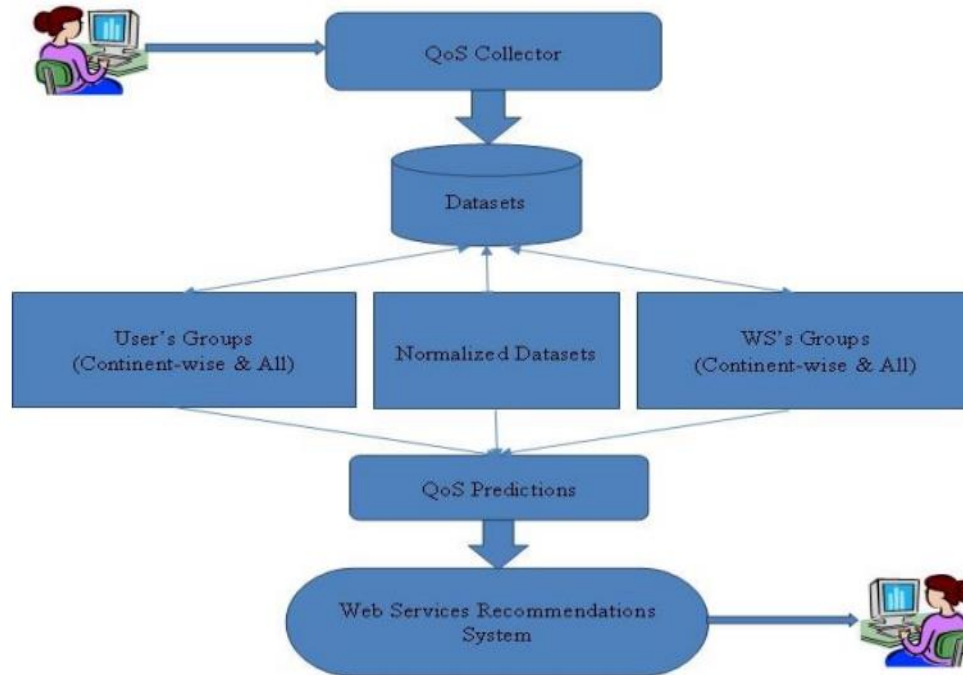


Figure 2: User-Continent Interaction with Web Services

PROPOSED METHODS

The purpose of the Experiment was to evaluate several state-of-the-art methodologies and important regression approaches for prediction accuracy. The following questions were meant to be addressed by the Experiments: 1) Is it feasible to utilize it to propose web services to consumers on the same continent? 2) Effects of dataset density.

Dataset

Data collection pertaining to commercial Web services from all corners of the globe is a challenging task in the actual world. Consequently, the dataset that was shaped by (Zhang, 2011; Dinget, 2014; Zheng, 2013) was utilized, which is related to web services. In order to gather these online invocations of Web services by active users in a cloud environment, they established a lab named PlanetLab (Chun, 2015). Table-4.1 provides information regarding the dataset. They developed two matrices, *rtmatrix* & *tpmatrix*, that provide throughput and reaction time values for various users across various web services. Where $RT_{i,j}$ in the *rtmatrix* represents the user's reaction time for Web service j and $TP_{i,j}$ in the *tpmatrix* represents the user's throughput for Web service j , respectively.

Preprocessing

The QoS attributes were preprocessed and eight new matrices were generated for each, labelled dataset1 through dataset8, utilizing a MATLAB tool that removes columns with missing values. In mathematics, matrices can be of varying sizes. First, the original dataset was used to extract dataset1, and then datasets 2–8 were also extracted from dataset 1. In both datasets, there were 43,798,8 QoS-value invocations, and the corresponding complete QoS dataset matrices were 339 by 1,292. These matrices are perfect for training MATLAB learning models to predict the target class for each column because they don't have any missing values. Table 1 provides in-depth statistical characterization of the preprocessed dataset 1.

Table 1: WS QoS-Dataset Statistics

Statistics	Response Time	Throughput
Scale or Range of values	0 - 20 seconds	0.06 - 1000 kbps
Mean value	0.6525	27.3271
Standard Deviation	0.7061	30.6155
Maximum value	19.9520	1000
Minimum value	0.0020	0.0600
Number of user's countries	31	31
Number of web services countries	29	29
Number of Users	339	339
Number of Web Services	1292	1292
Number of Invocations	437988	437988

- **Dataset1:** This information was extracted straight from PlanetLab's initial dataset. It has been preprocessed from the original data by decreasing or deleting the columns with missing values; its size is 339*1292. Among the most extensive databases, it is quite vast.
- **Dataset2:** The dataset has been normalized and has dimensions of 105*320. Each of the five continents was represented by a minimum of 64 online services, & minimum of 35 users were selected from each of the three main continents (35*3=105).

- **Dataset3:** All web services were recommended to only Asian customers. It measures 35 by 1292.
- **Dataset4:** For the benefit of Asian users, only Asian web services were utilized. The dimensions are 35 by 245.
- **Dataset5:** The survey exclusively polled European users & only recommended web services to them. It measures 123 by 1292.
- **Dataset6:** We exclusively used European web services for our European users. It measures 123 by 472 inches.
- **Dataset7:** All web services were suggested to users in North America. The measurements are 175 by 1292.
- **Dataset8:** We exclusively gathered web services for North American users. The dimensions are 175 by 419.

Table 2: Dataset that has been preprocessed

Attributes	Values
Number of users	339
Number of web services	1292
Number of user's countries	31
Number of web services countries	29
Matrix Size of datasets	339*1292

Table 3: Distribution of users by continent

Continent's name	# of users
Asia (AS)	35
Europe (EU)	123
North America (NA)	175

South America (SA)	6
--------------------	---

Table 4: Distribution of Web Services by Continent

Continent's name	# of web services
Asia (AS)	245
Europe (EU)	472
North America (NA)	419
South America (SA)	92
Australia (AU)	64

Metric for Evaluating Performance

To compare the quality of the predictions made by different regression algorithms, the MSE metric was utilized. Better performance is indicated by a smaller MSE value.

MSE is characterized as

$$MSE = \frac{\sum_{i,j}(U_iW_j - \sigma_i\bar{W}_j)^2}{N}$$

This is where U_iW_j stands for the QoS value that user i has seen for web service j , U_iW_j for the expected QoS value, and N for the total number of forecasted values. MSE is a statistical method that involves squaring and averaging the discrepancy between predicted or actual values.

RESULT AND DISCUSSION

Performance Metric for Predictions on the Dataset using 10-fold cross-validation

- The lasso approach consistently produces reduced MSE values under all default settings, suggesting improved prediction accuracy.

- As the number of online services increases, the MSEs of the lasso, fitensemble (Bag), fitree, & fitensemble (LSBoost) techniques decrease, suggesting that adding additional QoS metrics improves prediction accuracy.
- The prediction model performs best when users are grouped by continent. Preferring local services improves the accuracy of predictions.
- Compared to the prediction approaches recommended by (Chen, 2013), the lasso & fitensemble (Bag) models outperformed IPCC, UPCC, WSRec, CBRec, RegionKNN, and LoRec.

Table 5: MSE values for regularized dataset (105*320) and entire dataset (339*1292)

Mean (calculating MSE values for all columns)			
QoS properties	Regression methods	Full Preprocessed Dataset1 (339*1292)	Normalized Dataset2 (105*320)
Response time	Fitree	0.8755	0.8681
	Fitensemble (LSBoost)	0.8702	0.9162
	Fitensemble (Bag)	0.5444	0.5820
	Lasso	0.6097	0.6644
Throughput	Fitree	508.0189	383.97
	Fitensemble (LSBoost)	643.5024	482.2403
	Fitensemble (Bag)	365.0933	311.4466
	Lasso	304.9751	234.8782

Table 6: Based on the user's continent, recommend web services.

QoS properties	Regression methods	Asia		Europe		North America	
		Dataset3 (35*1292)	Dataset4 (35*245)	Dataset5 (123*1292)	Dataset6 (123*472)	Dataset7 (175*1292)	Dataset8 (175*419)
Response time	Fitrtree	1.4462	0.9167	0.9011	1.2011	0.8147	0.7233
	Fitensemble (LSBoost)	1.6343	1.0689	0.9309	1.2736	0.8499	0.8102
	Fitensemble (Bag)	1.2545	0.8917	0.5708	0.8266	0.5267	0.4995
	Lasso	0.9971	0.7267	0.6885	1.0389	0.5674	0.5358
Throughput	Fitrtree	165.8760	16.6473	339.2962	133.4694	726.7981	292.6088
	Fitensemble (LSBoost)	149.2656	29.2060	383.4500	137.0153	937.6143	445.2717
	Fitensemble (Bag)	130.7990	12.8699	248.3225	99.0320	534.0454	240.0586
	Lasso	102.9153	12.2483	197.9621	93.8626	426.7096	168.1700

Effects of distribution by continent

The values of certain QoS attributes, such as web service throughput and response time, differ greatly depending on user location & internet connection. As a result, users are categorized according to continents and web services. Using smaller datasets with improved precision will be done if web services are available at the local data center. It is preferable to suggest online services with local data centers to consumers in a certain location; for instance, Google offers data centers in Asia, Europe, the Americas, etc. Giving Asian users recommendations for web services is one example.

In proposing a service from the web services of data centers across the continent, the current work's output would significantly improve the prediction accuracy for customers distributed across the entire continent. When compared to response times predicted by the best algorithm fitensemble (Bag), recommendations made utilizing all online services (large datasets) for Asian customers had the lowest accuracy. Throughput QoS-properties also had the lowest accuracy predictions.

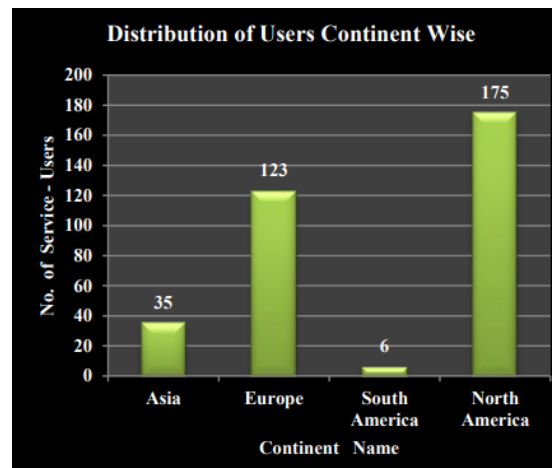


Figure 3: Users' Distribution by Continent

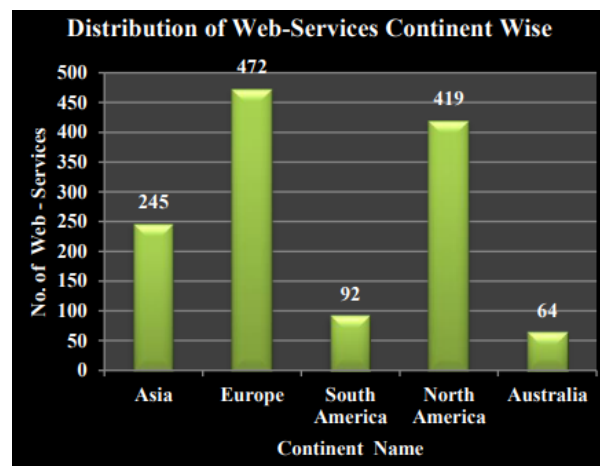


Figure 4: Continental Web Service Distribution

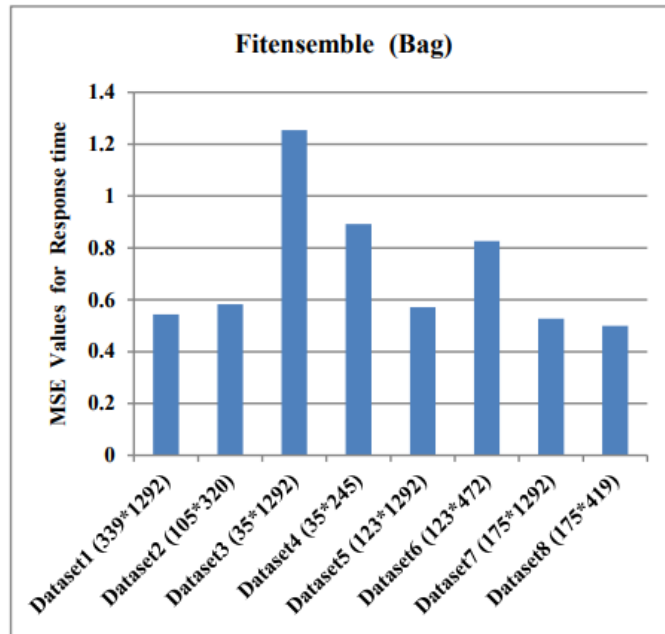


Figure 5: MSE Distribution for Response Time across all datasets using Fitensemble (Bag)

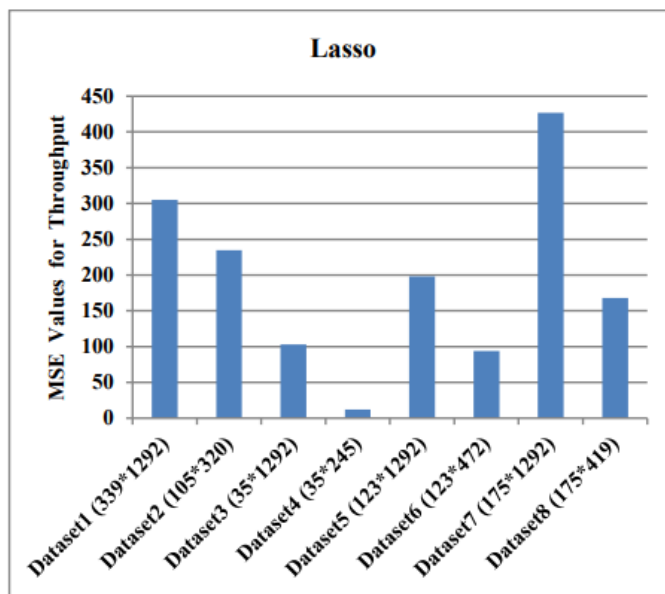


Figure 6: Lasso distribution of MSE for throughput across all datasets

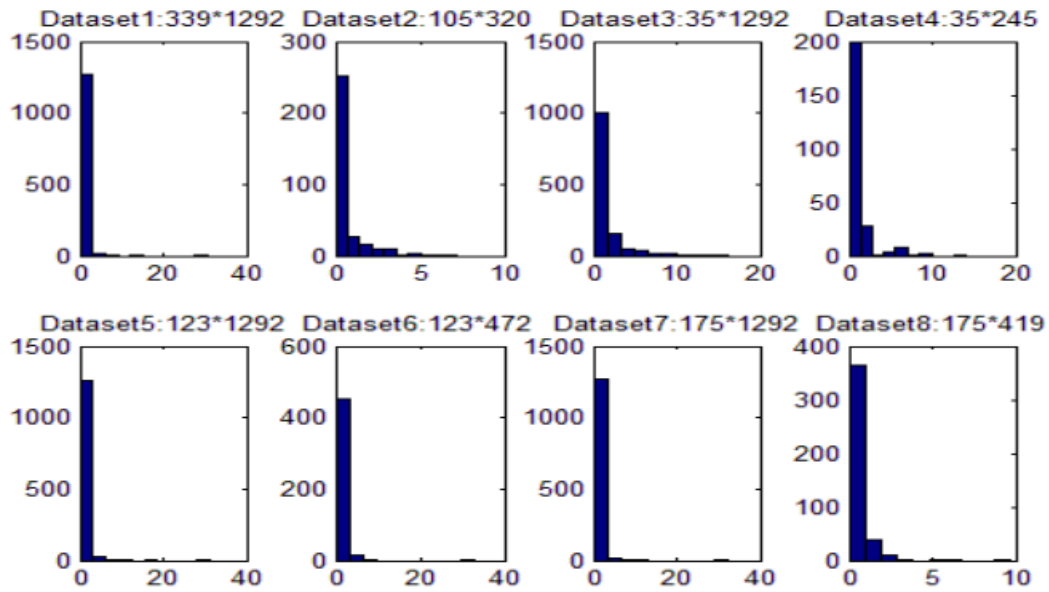


Figure 7: The fitensemble (Bag) distribution of MSE for response times across all datasets

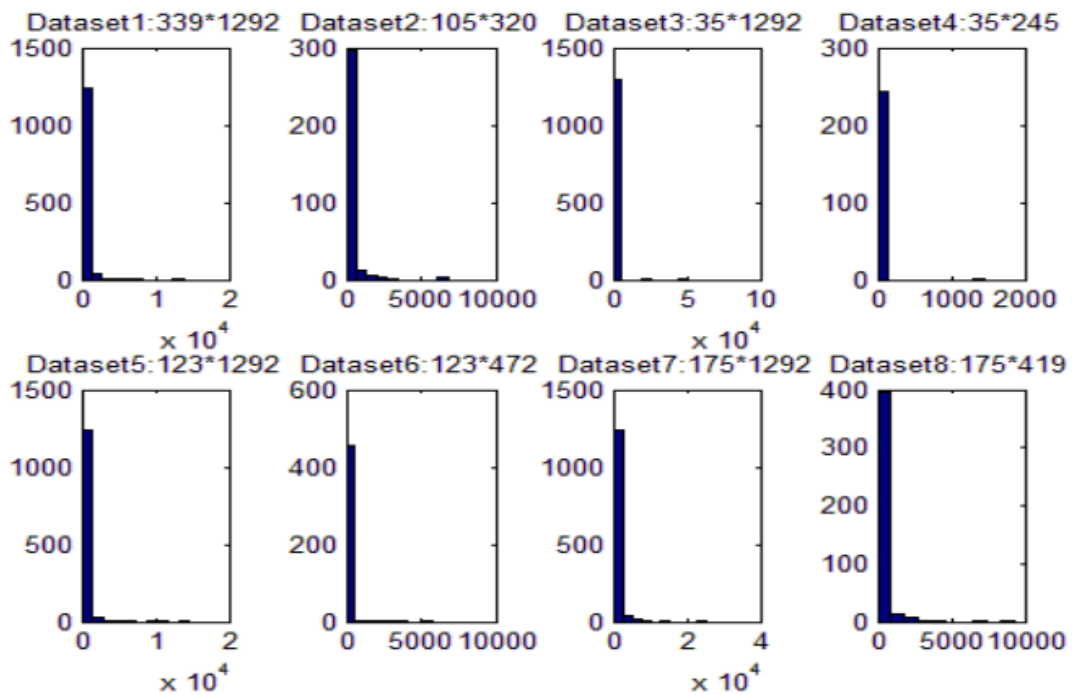


Figure 8: The Lasso method for distributing MSE for throughput among all conceivable datasets

Effects of predicting a missing value

To anticipate the missing values in the training matrix and make it denser, the missing value prediction uses similar web services & users. A training matrix and a target value vector were utilized in the suggested prediction model. The size of the training set matrix is $(u*w-1)$ & size of the target set matrix is $u*1$. Full web services and each user's web services column were both included in the projection. We used a linear regression model to forecast web service QoS values, assuming that customers do not have access to these metrics.

In order to determine which linear regression method provides the most accurate recommendation of web services, four different kinds of algorithms were used in a comparison fashion to examine the effects of missing value prediction. As part of the trials, the most active users from the real dataset were selected, and then, from both the QoS characteristics datasets, the denser and more prevalent web services were chosen. Each of the two QoS datasets—response time & throughput—had 1445 and 1347 full web services, respectively. However, for the sake of this study, we will only be using 1292 common web services from both datasets, as we have previously eliminated those with missing values. Consequently, for the two QoS datasets, a full denser matrix measuring $339*1292$ was employed. In the first step, the prediction model was applied to all eight types of datasets using the QoS parameters, which are throughput and reaction time. Utilizing missing value prediction on four logically distributed datasets, we were able to forecast user values based on continent and entire web services. The following datasets were used:

- There are three types of missing value predictions: (1) for full datasets, (2) for normalized datasets, where each continent has an equal number of users or web services involved or employed, and (3) for users on a single continent who access distributed web services globally.
- The inability to forecast values for customers on the same continent who utilize the similar web services.

CONCLUSION

Regarding the Suggestion for Customized Online Services, It was first discovered that, when it came to offering throughput, Lasso regression algorithms performed the best across all datasets. For all dataset versions except dataset3 & dataset4, which yielded the best results

when employing the lasso approach, Fitensemble (with Bag ensemble method) outperformed other algorithms in terms of reaction time. Secondly, it was discovered that ASIAN web service recommendations for ASIAN users are more accurate & appropriate. It was determined that users on the same continent would benefit most from continent-wise service recommendations in the event that services were available at the local data center of that continent. This is because customers on the same continent would have to travel less distance and spend less money to access the services. It was suggested that in the event that services were not available at the local data center on the continent, services from the main data center or data centers on other continents may be used instead.

References

1. Ardagna D. Casale G. Ciavotta M. Pérez J.F. and Wang W. (2014) 'Quality-of-service in cloud computing: modeling techniques and their applications,' *Journal of Internet Services and Applications*, Vol.5, No.1, pp. 11.
2. Babu K.D. and Kumar D.G. (2012) 'Allocation Strategies of Virtual Resources in Cloud-Computing Networks', *Int. Journal of Engineering Research and Application*, Vol. 4, No. 11, pp. 51-55.
3. Ergu, D., Kou, G., Peng, Y., Shi, Y., & Shi, Y. (2013). The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment. *The Journal of Supercomputing*, 64, 835-848.
4. Kumar N. and Saxena S. (2015) 'A preference-based resource allocation in cloud computing systems', *Procedia Computer Science*, Vol. 57, pp. 104-111.
5. Moura J. and Hutchison D. (2016) 'Review and analysis of networking challenges in cloud computing', *Journal of Network and Computer Applications*, Vol. 60, pp. 113-129
6. Nema, P., Choudhary, S., & Nema, T. (2015). Vm consolidation technique for green cloud computing. *Int J Comput Sci Inf Technol*, 6, 4620-4624.
7. Praveen S.P. Rao K.T. and Janakiramaiah B. (2017) 'Effective Allocation of Resources and Task Scheduling in Cloud Environment using Social Group Optimization', *Arabian Journal for Science and Engineering*, pp. 1-8

8. R. N. Calheiros, R. Buyya, C.A.F.D. Rose, A heuristic for mapping virtual machines and links in emulation testbeds, in Proceedings of the 38th International Conference on Parallel Processing, Vienna, Austria, 2009.
9. Sharkh M.A. Ouda A. and Shami A. (2013) 'A resource scheduling model for cloud computing data centers', In Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, pp. 213-218, IEEE, 2013.
10. Xiong A. and Xu C (2014) 'Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center', Mathematical Problems in Engineering, 2014
11. Ye, K., Jiang, X., Huang, D., Chen, J., & Wang, B. (2011, July). Live migration of multiple virtual machines with resource reservation in cloud computing environments. In *2011 IEEE 4th International Conference on Cloud Computing* (pp. 267-274). IEEE.
12. Zibin Zheng, Hao Ma, Michael R. Lyu, and Irwin King, Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization, IEEE Transactions on Services Computing, Vol. 6, No. 3, 2013, 289-299.