Quality of Services

Amruta Deshpande, Amit Shah, Chirag Kular

ABSTRACT

The quality of Service (QoS) aims that network providers provide better services on the internet and adheres to web service requirements such as performance, availability, security, and reliability. The number of web services available today is huge and to identify a good quality web service is a difficult task. One way is to test these web services is based on features like performance, reliability, etc and rank them. Another approach is to use QoS prediction methods, which is an active area of research. The current techniques used in QoS prediction mainly depends on set parameters like response time, fee, throughput etc. that are prior known to the developer. With the dynamic change in the web, it is difficult to determine the value of QoS. The current approach fails to consider the end user perspective for the quality of service. In order to overcome this problem, we will concentrate on QoS feature extraction mentioned in [3], since it considers reviews and ratings submitted by users to extract features and also identifies the sentiment of the reviews simultaneously. Thereby increasing the feature usability and cost-effectiveness.

1. INTRODUCTION

Web services allow performing communication and data exchange between the two system over the network on large scale. With the rapid increase in the number of web services reliability, ranking, interoperability and quality are important aspects for selecting optimal web services. A quality of service in terms of web service is widely used to determine message persistence, avoiding duplicate messages and acknowledgment of the messages sent and received. A quality of Service has characteristics like response time, throughput, security, reliability, invocation failure rate. With the rapid change in the web, it is important to know the QoS value of service as it helps the user to select the most suitable service and also plays a major role in determining the success of the service provider.

A QoS-based approach is been widely used in service recommendation, selection, composition and discovery, monitoring and also for QoS management. Web service recommendation mainly focuses on QoS value of web service in order to recommend a service to the user based on the past experience of the service users [12]. Web service composition focuses on maximizing the Quality of service by combining services which consider the constraints and preferences of the service users [9]. Web service monitoring focuses on the monitoring service requirements like availability, reliability, service level objectives and performance and helps to evaluate the QoS testing process [8]. QoS prediction mainly tries to predict the value of Quality of Service based on user experience and traditional parameters like time, location, throughput.

The paper primarily focuses on 3 domains: QoS prediction, QoS composition, and QoS monitoring. As discussed earlier QoS prediction is mainly used to determine QoS values that use QoS features such as performance, availability, security, reliability, past data, it's location and user reviews. These features are determined by domain experts of QoS and are used by current QoS approaches. This approaches evaluates QoS features based on two resources. The first resource provides information to the users like security level and invocation fee. The second resource runs the tests and QoS values are collected. However, these approaches still have few limitations as they do not consider the interest of the user and also it's not worth to rely on the data provided by the service provided.

This paper describes different method and parameters used in QoS-predictions by various researchers. The authors of the papers [2], [6], [7], [11] and [13] have commonly used Collaborative Filtering(CF) method which is one of the most popular method being used for QoS prediction. CF uses Pearson Coefficient Similarity(PCC) for finding the similarity between the users and the services. Paper [2] uses location information of users and web service as key parameter and proposes LMF-PP method. Paper [6] uses historical data and proposes 2 phase K-means clustering approach to detect unreliable data from untrustworthy users. In addition, paper [7] tries to improve QoS performance by using CAPred prediction model whereas paper [13] tries to maintain the privacy of user information by using data obfuscation technique. Paper [3] and [4] mainly focuses on web service recommendation based on QoS prediction. Paper [3] uses feature extraction and evaluation technique for service recommendation whereas paper [4] uses multi-dimensional QoS data for the same. Researchers of the paper [5] mainly focuses on predicting the popularity of the API as it is a key factor in determining profit for service provider and the consumer. The model used for finding API popularity is the linear regression model.

The second category discussed here is QoS composition. There are many models available today to predict the QoS of composed systems but these composition fails to consider the connection between services when calculating QoS of composition. The authors of the paper [1] consider factors that influence QoS of context-aware services. This paper presents trust prediction framework that uses information like context-QoS and context-context dependencies for QoS prediction.

The third important category discussed is Quality of Service Monitoring. Monitoring of services are required to keep a record of ongoing quality of service and also to check whether the service is meeting the expected performance. [10] paper proposes a novel approach called weighted Bayesian Runtime Monitoring(wBSRM) that considers the environmental factors like position of the server, user and load at runtime is put forth thereby addressing the issues of the previous approaches.

The rest of the paper is organized as follows: Section 2 discusses the related work according to the Quality of service domains and section 3 discusses the plan of implementation.

2. RELATED WORK

2.1 QoS Prediction

2.1.1 Location-Based Web Service QoS Prediction via Preference Propagation for Improving Cold Start Problem

In [2], web Service's performance depends upon the QoS values, thus prediction of QoS values is of utmost importance. The classical technique to predict the QOS values is Collaborative Filtering (CF) but it faces the drawback of Cold Start problem. This paper proposes a Location based Matrix Factorization via Preference Propagation (LMF-PP) to overcome the cold start problem when predicting the QoS values for a web service. Service Based Software Development (SBSD) is method which combines different web services and it has recently taken over the software development paradigm. The challenge in SBSD is to predict the values of QoS. One approach is to call each individual web service and note down their QoS values. However, this approach requires the investment of time and cost.

Recently, CF has been in demand and the idea behind it is to find user and services which are similar to fill the missing values. The CF technique takes the historical invocation data and tries to predict the QoS values of web services. The CF method has two different approaches: memory-based and model-based (Matrix Factorization). The later one is more accurate than the former since it takes into consideration the additional information which are required for web service QoS prediction.

The scenarios which doesn't allow to do any prediction since data is not available is defined as a cold start problem. This paper proposes LMF-PP to avoid cold start problem by two approaches: analysis of user and web service's location and preference propagation method. In Matrix Factorization two or more sub-matrices are formed by factorizing a matrix. Missing values are predicted by performing factorization again. The results obtained are stable and of high accuracy.

The model-based approach rightly known as the matrix factorization model whose performance of prediction is staggering have been used in many studies. This model-based approach has higher efficiency in prediction when compared with memory-based approaches. Another technique known as Hierarchical matrix factorization organizes the modeled matrix geographically into local matrices which basically contains the users and services. One major issue with web services is the cold start problem which is handled by the extended model-based CF which calculates the distance between users with the help of altitude and latitude data of the users. This model has an assumption that users who are near to the web services, they are using similar QoS at the time of invocation of web services for the reason that local user may share similar network infrastructure. Using the above assumption, the QoS can be interpreted. Nonetheless the limitations of cold start and sparse matrix still exists.

In order to make dependable predictions with respect to cold start the preference propagation is used for combining users that are locational and web services having similarity computation. The preference propagation also solves the problem of sparse matrix which helps in estimating the similarity matrix correctly. The changes of the QoS of web service is depended upon the country location of users and web services.

The experiment to evaluate the LMF-PP approach contains a large-scale dataset of real word QoS values of web service and they compared nine web service which are specifically prediction techniques of QoS. The response time ranges from 0-20 seconds and 91% of dataset have response time less than 2 seconds. They also considered the cold and warm state environment in matrix densities. The performance of predicting the QoS of web service is measured by Mean Absolute Error(MAE) which is average magnitude of errors and Root Mean Square Error(RMSE) which measures the errors of the QoS predicted values. Also the parameter α helps to controls the influence of preference propagation while parameter γ controls locational terms influence for the objective function. For the better QoS prediction results, both the measures i.e. MAE and RMSE should have smaller values.

In conclusion the problem of cold start which is known as the data scarcity problem has been described as a major issue in CF and for this to solve one of the proposed technique is Location Based Matrix Factorization via preference propagation.

2.1.2 CAPred: A Prediction Model for Timely QoS

In [11], the number of web services available in the market is rapidly growing and to select a high quality web service among them is a difficult task. User evaluated QoS data is collected for a web service which helps in the selecting the optimal web service. But this data is not in timely manner and it is sparse. This challenge is solved in this paper by proposing a novel model called CAPred. This model divides the historical data into different time slices which itself is a 2D matrix. The model analyzes each time slice to fill missing values in the matrix and then use this data to predict the QoS values of web services. The historical data is evaluated in space and time dimension.

Prediction of dynamic QoS values is done with help of autoregressive-model-moving-average(ARMA) model. But this model fails under the real world QoS data since it is sparse and doesn't fit in time series analysis. This paper solves the sparsity problem by combining the data in both space and time dimension.

In CA Pred, the user asks for performance information of Web Service. Then calculation of similarity between this user and other user is computed. Then we fill the vector data's missing values through CF in each time slice. Then a data of user's invocation to web service is recognized in a time series. This time series model is used in a prediction algorithm.

The algorithm has two steps:

(a) Data is divided into single time slice which represents user-service invocation matrix at a given point of time. We generate and compute all the missing values identified in each time slice.

(b) ARMA model helps to predict QoS values since we have enough data for the time series analysis. Autoregressive model and Moving-Average model are parts of the ARMA model.

The experiment is conducted on a data set collected from WS-DREAM site. Various measures have been taken during the experiment and model is evaluated on two criterion: Mean Absolute Error(MAE) and Root Mean Square Error(RMSE). The authors have conducted one experiment on CAPred model and other three control experiment. The CAPred model has three thresholds SP, SM, SA and to study their impact, authors conducted separate experiments to select best parameters in the CAPred. The comparison of two models is conducted after testing the performance of ARMA. The results of the experiment shows that CAPred has better performance compared to other models. The results of other models have some random values of MAE and RMSE.

By combining user-based CF method and ARMA model, CAPred does the prediction of QoS performance of web service. The model can still be improvised for practical usage and model can also combined by considering different ways apart from time and space.

2.1.3 QoS Prediction of Web Services Based on Two-Phase K-Means Clustering

In [6], the unreliable data from the untrustworthy users leads to select inappropriate web service from a set of services thereby reducing the accuracy of the QoS prediction. The previous methods like Collaborative Filtering(CF), Reputation system founded the missing QoS values, but did not take into account about data credibility. The researchers of [6] proposed a new method called credibility-aware QoS prediction method (CAP) which detects the unreliable data and also the source of this unreliable data which is untrustworthy users in order to keep credibility of the data. Further the method also predicts the missing QoS values. The dataset is from the WS Dream collected by Zheng.

The Credibility-aware QoS prediction(CAP) method mainly uses the unsupervised k-means algorithm in 2 phases. In the first phase, untrustworthy users are identified by clustering similar QoS values and by keeping a track of index calculation made by the untrustworthy users. In the second phase, a set of groups are formed based on the index calculated thereby identifying the untrustworthy users. Further, the missing QoS values are predicted based on the cluster to which they belong.

K-means algorithm is the unsupervised machine learning algorithm that classifies similar group of data into one single cluster where K defines the number of clusters formed. In the credibility-aware QoS prediction method, it mainly tries to find outliers i.e the clusters that are have very few elements and are far away from other clusters. The outliers are considered as the untrustworthy users.

The dataset is taken from WS Dream collected by Zheng which consist of 339 users and 5825 Web Service. The data set mainly consist of two matrices. First one is response-time Matrix(rtMatrix) and throughput matrix(tpMAtrix). Normalized Mean Absolute Error (NMAE) metric is mainly for common basis of comparison between two methods. Further it discusses the impact k value in K-means clustering. The experiments shows that value of K has a remarkable impact on prediction accuracy. Higher value of K is not suitable for the CAP method. The CAP method outperforms when the K value is between 5 and 8.

The performance measure of the CAP outperforms other methods like UPCC (User-based collaborative filtering method using the Pearson Correlation Coefficient), IPCC (Item-based collaborative filtering method using the Pearson Correlation Coefficient), UIPCC (combination of UPCC and IPCC) and RAP. The metric used here for comparison is the Normalized Mean Absolute Error (NMAE) which measures the prediction accuracy. The smaller the value of NMAE the higher the prediction accuracy. CAP has the very less value of NMAE and thus outperforms as compared to other CF-based approaches.

The approach of predicting QoS values based on two phase k means performs well as compared to other CF based approach and also helps to identify the untrustworthy users. Thus, we can say that this is one of the best prediction approach of Web services.

2.1.4 A Privacy-Preserving QoS Prediction Framework for Web Service Recommendation

[13] comprises of two appendixes: Collaborative filtering and mathematical basics. *Collaborative filtering (CF):* It is commonly used in commercial recommender systems, wherein CF works for the rating prediction problem. Users tend to rate the items that they know about, the movies they watched or the book they read about. Generally, each user tends to rate only a small set out of all the items available, as the number of items is quite large. Thus, the user-item rating matrix is very sparse. The basic idea of CF is to build a model that can predict the unknown values, based on the perception that similar users may have similar preferences of the same item, and thus provide similar ratings.

For this purpose, two types of CF techniques have been studied:

(a) Neighbourhood-based CF approach: This approach is a fusion of user-based approaches that uses similarity between users and item-based approaches that make use of similarity between items. The shortcoming of this approach is that it is incapable of handling the data sparsity problem and it also has a high time complexity.

(b) Model-based CF approach: This approach uses a predefined compact model to fit the training data that can be used to predict the unknown values. Matrix factorization model, a popular model-based CF approach, is known to achieve better performance than neighborhood-based CF approach.

Mathematical Basics:

(a) Euclidean Norm: It is a vector norm. It is the squared root of the summation of each element in the vector squared.

(b) Gradient Descent: It is used to find a local minimum of a function in an iterative manner. For the experiments in the paper, the approach PMF is implemented by using gradient descent algorithm.

The input to this algorithm is the collected QoS matrix, the indication matrix and the model parameters. Gradient descent works on batch-mode, thereby needing all the data to be available. The latent factors move iteratively by a small-step of the average gradient.

(c) Stochastic Gradient Descent: The idea behind stochastic descent gradient (SGD) is to update stochastically or randomly using the sequentially incoming data. On every step, the model is updated based on the current data sample. This means SGD provides for an online algorithm that adjusts the model based on each data sample from the data stream in an online manner.

For the experiment, the authors have used two QoS attributes: Response Time(RT) and throughput(TP). The experiment uses a real-word Web services dataset and to measure the outcome of the experiment, a standard error metric , MAE(Mean Absolute Error) is used. The accuracy of QoS prediction is higher if the MAE value is small. The experiment takes into consideration the effect of data obfuscation (RQ1) which preserves the user privacy pertaining to QoS data, effect of prediction accuracy (RQ2) which requires the users true QoS data, tradeoff between accuracy & privacy (RQ3) which takes into account the effect of noise range α and effect of distribution of random noises (RQ4). The results of experiments show that random noises have performance impact on the privacy-preserving approaches. The objective of this paper is to address the privacy issues for Web Service Recommendation and they have presented an impressive initial attempt in their two approaches. This paper encourages that there is a lot of scope in the future for web service recommendation even by ensuring the privacy.

2.1.5 *Predicting QoS Values via Multi-dimensional QoS Data for Web Service Recommendation*

in [4] services are software components designed to support interoperable machine-to-machine interactions. Quality of Service (QoS) is an important basis to judge if a service is suitable to recommend or not. Historical multi-dimensional QoS data needs to be exploited as much as possible when predicting necessary unknown QoS values to help recommend the best candidate services for each client's service selection request. The drawbacks of lack of an integrated approach to analyzing multi-dimensional QoS data includes less accurate QoS prediction, unsatisfactory service recommendations and complex QoS prediction systems.

The paper proposes an integrated QoS approach to unify the modeling of multi-dimensional QoS data via a multi-linear-algebra based concept of tensor and enables accurate prediction via tensor decomposition and reconstruction optimization algorithms.

The High Dimension Oriented QoS Prediction (HDOP) aims to enable predicting the unknown QoS values in any dimension easily. The problems to achieve the goal is: (1) how to model the multi-dimensional QoS data? and (2) how to use the model to make predictions? The paper discusses the HDOP QoS prediction algorithms as follows:

(a) Multi-Dimensional QoS data as a Tensor

The paper introduces concepts of tensor and its decomposition.

- Concepts of Tensor: Tensors in HDOP are multi-dimensional arrays. Some useful tensors operations make it easy to predict QoS values. The paper talks about the five dimensions based on a figure represented in the paper and how these dimensions can be useful for predicting the value of multi-dimensional arrays in terms of each user.
- Tensor Decomposition: Based on tensor decomposition formulation, decomposition is performed on a specific tensor to find its component matrices and estimate unknown values. The biggest problem is determining R (rank of tensor) accurately.

(b) QoS Prediction using tensor model:

Computing the component matrices is essential for which two things need to be determined:

- Optimization Goal: To avoid over fitting issue, the optimization goal is considered just a modification of the loss function obtained from tensor decomposition.
- Optimization Algorithm: The first step involves computing partial derivatives of the loss function. The second step involves updating the iteration step size and doing the iteration. After each round of iterations of all component matrix entries, the loss function is computed. The computing would be finished once the loss function is less than or equal to precision requirement of tensor decomposition and all component matrix are got.

The paper then talks about performing experiments to validate the HDOP, for which they select two datasets which includes WSDream dataset and Beijing taxi passengers (BTP) dataset. The two main parts to the experiment include: (1) comparing their algorithm with other three well known prediction methods; (2) studying the optimal parameter settings. A brief description of the datasets is also provided. In terms of the experimental setup, the details of the hardware and software are also described. The metrics used for comparison is the Metric Absolute Error(MAE) and RMSE.

There are two methods used for comparison:

1. Special characteristic based methods: Makes prediction based on special characteristics like context-aware, time-aware and location-aware and so on. Time-aware method is compared with HDOP considering WSDream dataset while location-aware method is compared with HDOP considering BTP dataset.

2. Matrix factorization based method: Makes prediction by factorizing the user-service matrix wherein each entry is the QoS value of a user-service invocation. The method is compared with HDOP considering both dataset.

On basis of the prediction accuracies of HDOP and the comparisons with other methods, it is found that HDOP is more accurate than all the other methods for the two chosen datasets.

For good performance, the setting of the tensor decomposition precision, the proportions of two parts of loss function and rank of tensor is determined mathematically and from the accuracy comparisons.

Thus, the approach proposed in the paper first models the multi-dimensional QoS data as a tensor, then finding the component matrices by decomposing the QoS tensor. The component matrices thus obtained helps to reconstruct the QoS tensor accurately. Finally, the reconstructed tensor helps in the prediction of the unknown values of QoS data. We can say that the advantage of this approach is that it can used to predict any dimensional QoS data easily and accurately.

2.1.6 Time-Aware API Popularity Prediction via Heterogeneous Features

API's are becoming prominent web services and they are gaining popularity in recent times. In [5], authors have proposed an approach of predicting API's popularity. This will help the developers and consumers to select the best API which will suit their purposes. The prediction of API popularity is made with help of heterogeneous features which includes time series features, API's self-features and API provider ranking and description features. When developers combine more than one API's for programming web, it results in a Mashup. APIs popularity is decided upon the number of mashups an API is used. However, time series factor is not considered when deciding upon the API's popularity. This paper proposes a time-aware model to predict API's popularity by considering features of different data structures such as time-series features, its numerical features, categorical features, and textual features as well.

For the dataset, the authors have considered the Programmable Web platform which consist of API's and mashups. The authors did the statistical analysis on this dataset and laid down certain factors like provider rank, information integrity, category and API's followers which contribute towards the popularity of API.

The authors have used relative squared error(RSE) and mean relative squared error(mRSE) as an evaluation metric for evaluating the models in the paper. The model has better performance if it has less mRSE.

There are previous models like Szabo-Huberman (S-H) Model, Linear Regression (L-R) Model and LR-HF model which predicts the popularity of APIs considering only one features and neglecting the others. The author proposes LR-HF2 model which uses all the heterogeneous features for the prediction.

To evaluate the performance of all the models, the author divided the dataset into junk, popular and full dataset and performed the experiment on every one of them. The results show that author's time-aware model showed better result which helps in recommending popular APIs to the developers and the users.

2.1.7 A Learning Approach to the Prediction of Reliability Ranking for Web Services

In [7], Reliability ranking for web service is yet another important feature that helps to select the useful web service from the set of services. The researchers of the paper [7] propose a learning approach for predicting reliability ranking of web services called LAPRR which is mainly based on preference function. For reliability ranking, history logs are used as a source. They are obtained by evaluating the web services at the client side. In the previous studies, failure probability of services observed by the users are used to determine the reliability ranking for web services. The information gathered in the log were sparse and to fill the missing value in the failure probability was one of the issue. Also, to get preference function was yet another issue. The approach provided in this paper overcomes issues that existed in the previous papers.

(a) Filling in Incomplete Failure probability matrix.

    To fill the missing values in the failure probability matrix, similarity between users and services is computed using the Pearson Correlation Coefficient(PCC). PCC is mainly adopted by a User based collaborative filtering and Item based collaborative filtering.

    For similarity Computation, User based collaborative filtering(UPCC) helps to find the similarity between the 2 users using the service and is in the interval of -1 to 1. The higher the value of the PCC higher is the probability of the 2 users to be similar.

    Item based collaborative filtering using PCC(IPCC) helps to find the similarity between the 2 services and has interval similar to that of the UPCC i.e between -1 and 1.

    To predict the missing value of failure probability we can use the similar neighbor selection. Here it finds the set of similar services and set of similar user

(b) Missing Values prediction: Using the user based collaborative filtering (UPCC) we can predict the missing failure probability for a user. Similarly, we can predict the missing value using the item based collaborative filtering(IPCC) for the services. The values obtained for the missing values can have different performance measure and can be balanced by a parameter lambda.

(c) Framework for learning the preference function: The learning approach uses the past service logs to train the preference function. Suppose a user has given ratings of web service reliability on a set of items, then we tell preference of user from given set of items.

WSDream dataset is used to perform prediction of reliability ranking and consist of service invocations on 3568 Web services from 102 service. The data is converted to user-service matrix. Normalized discounted cumulative gain (NDCG) is used as a performance matrix. The larger value of NDCG indicates better accuracy ranking. The denser the user-item matrix, the higher the value of NDCG giving high accuracy prediction.

Comparison is being done with greedy approach, Cloud Rank1, CloudRank2 where the approach proposed by the researcher i.e. LAPRR gave the highest prediction accuracy as it contains highest NDCG value.

The advantage of this approach is that it did not invoke any additional service and have simply used the preference function between the pairs of service in order to predict the reliability ranking for web service.

2.1.8 Extracting, Ranking, and Evaluating Quality Features of Web Services through User Review Sentiment Analysis

In [3] traditional methods QoS of web services is based on a fixed set of parameters such as response time, fee and availability. This fails to consider the actual features that matter to the user. This paper proposes an approach to extract QoS features of a domain, ranking the features based on how interesting they are and evaluating the value of these features based on sentiment analysis of the reviews posted by the users. In contradiction to the traditional methods where the QoS features are provided by data published by the service provider and third party agents, this paper proposes a new approach where user reviews are considered to rate the features of a service. This has several advantages over the traditional methods. Each user review is categorized in positive or negative based on augmented logistic regression (l1).

The raw user reviews go through preprocessing steps such as review refining, word tokenizing, conjunction word removal, Part of Speech Tagging (POS Tagging). After refining the user reviews, sentiment analysis is done on the reviews to rank them as positive and negative and identify the features that matter to the user. The paper then implements a small modification to the regularized Logistic Regression model to consider QoS related sentiment analysis to extract the features. When performing the sentiment analysis over reviews every distinct word becomes a feature, this makes the model highly likely to over fit the training data. To improve the model prediction regularization needs to be applied, the paper choses *l1* regularization. The paper then evaluates how their proposed approach performs by considering the accuracy in two ways. Accuracy of features extracted and accuracy of sentiments. It is then evaluated against other models to rate its performance.

The paper then conducts an experiment by selecting user reviews from an online website and then running the *l1*-RLR model on it and then quantifies the results. First the results were cleaned by removing the less frequently occurring data and invalid data. Then the reviews were manually given a sentiment since the reviews were inconsistent with the rating. Then the features extracted by this model were compared with the features extracted by other models by comparing the top K features extracted using the recall@K as a measure to rate the results. It was observed that *l1*-RLR was much more efficient than the previous approaches. To verify the authenticity of the results Sentiment Classification Evaluation was done. It was observed that primary due to regularization the accuracy of *l1*-RLR was much higher than the previous approaches since it effectively reduces the feature space.

Most of the previous approaches for evaluating the Quality Features of Web Services had a different focus than this paper since this paper focuses on user feedback which is the most oriented towards what features a user would care about in a service. In most cases of previous papers to rate the features, either predefined QoS parameters were considered or the QoS values were considered which is not as oriented to the users as the proposed approach in this paper.

For predicting the QoS values, the data is crawled from sitejabber site. The data retrieved mainly consist of computer, business and mobile-phones domain reviews. This reviews are further used to determine positive and negative reviews and for freature evaluation. In order to calculate the accuracy of the features extracted, recall @K is used where top K features are used for comparison. These top K features are compared with the ground truth feature set.

Comparison of *l1* Regularized Logistic Regression(*l1*-RLR) is performed with the Logistic Regression and frequency based method. *l1*-RLR out performs as compared to that of LR when there is a large size of the feature space. Also, *l1*-RLR overcomes the issue addressed by frequency based method of selecting words that are irrelevant.

The approach presented by this paper effectively identifies quality feature and surpass present approach for QoS prediction. Thus, we can say that the approach presented in this paper is the novel approach for feature extraction and evaluation for predicting quality of service value.

2.2 QoS Composition

2.2.1 A QoS and Trust Prediction Framework for Context Aware Composed Distributed Systems

In [1], it's necessary for Web Services to perform efficiently in a context-free way for the fact that over a few years there has been evolution of pervasive computing along with broad endorsement of smart phones. In such type of web Services, two major aspects rely on context of services which are the functional attributes and the QoS attributes which is the response time and availability. The system composed of services such as trust which is the level of consent of a service that agrees to its specification and QoS interpretation of service have considered these services under context dependencies. The discussion in the paper revolves around the proposed model that makes use of data related to individual services with respect to context-QoS dependency as well as the interaction pattern which is an inter-service for QoS and composition of trust of services that are required at the phase of designing.

One of the efforts predicting the QoS of a composed system at an initial design stage described in paper was by Jaeger et al. This effort takes into account the QoS composition properties along with the interaction patterns which helps in calculating the QoS of compositions by making use of QoS of individual services. Several composition attributes such as time, throughput etc have been analyzed along with the frequent interaction pattern such as sequences and loops. Nonetheless the concussion of environment and their interrelationships which are elements that alter the QoS of context-aware services have not been taken into consideration by these operators.

The context-aware nature of the services makes them adaptable to their ever changing environment. Distributed Systems making use of these services play an important role in applications that are built over the Internet.

The model discussed in this paper is about predicting the trust and quality of service at an initial stage of the context aware distributed environment. The context of service has classified into 5 different categories such as the attributes physically affecting the Quality of service (e.g. Temperature, location), associative context which is the ramification occurring the QoS due to the presence of other services (e.g. synchronizing, web sessions), several inputs and configurations affecting the QoS and consequences of hardware on QoS which runs the service. The result of the analysis on QoS services indicate that problems which are fixed at a later stage of the software development life cycle requires much more time than those that are fixed early.

The model approach have been experimented with the help of a case study of the bayesian network of the camera service. It's context attributes consists of distance, angle, resolution width, height and QoS attributes such as response time and tracking error. The metric used for tracking error associated with interferences results is relative absolute error.

The trust prediction framework normally composed of four phases which are individually collecting dependency data of the Context-QoS, collecting the data with regards to interaction pattern in a composed system environment, interpreting the Bayesian network that is required to identify the dependencies for the context-QoS and lastly answering the relevant trust queries using the interface techniques. An indoor tracking system has been used to identify the potency of the proposed framework with the help of empirical validations.

The data to train the Bayesian network is obtained through execution traces once the service starts running. The data consist of context attributes and the QoS attributes. Relative absolute error(RAE) metric is used to calculate the absolute error corresponding to results of inference. The error values are less for those algorithm that use context in the predictions.

When compared with other prevalent approach that do not consider context, Bayesian network approach has relatively low absolute error of predictions of response time. Thus, it can be inferred that trust predictions which include context in it have a higher accuracy than to those algorithm that do not consider context.

2.3 QoS Monitoring

2.3.1 A Novel QoS Monitoring Approach Sensitive to Environmental Factors

In [10] Quality of service relies on third-party services. Due to this reliance service-oriented system requires runtime monitoring technique. Influence of environmental factors like position of server, user and load at runtime should also be considered. They should not be ignored and to solve that problem a novel approach is suggested by the authors of this paper.

Service Oriented Approach is widely used by internal and external enterprises. The success of the SOA depends on the third-party services which require runtime quality assurance. QoS properties can be expressed by probability quality properties. As an example, we can say that Service reliability i.e the average running time without failures of a service in one year is 95%. Response time can be described as the probability of response time is within 8s after sending invoking operation to this service is 90%. QoS heavily use monitoring approaches towards probability quality property. One such approach is ProMon (Probabilistic Monitor) based on SPRT (Sequential Probability Ratio Test). This approach is based on Wald's hypothesis test. It makes two hypothesis H1 and H2. Then judges the hypothesis whether it should be accepted or rejected. They also have some drawbacks like a significance level 'alpha' is needed to calculate the rejection area of the hypothesis which can at times give opposite conclusions and sometimes it is invalid to check with the refusing area given by 'alpha'.

However, none of the Bayesian algorithms can solve the drawback where one hypothesis of Bayesian does not suit all cases. In QoS monitoring, fluctuation of running environment and context can affect the execution environment of service. These factors will affect the impact of samples which will in turn affect the overall determination. Failures brought by Bayesian algorithm, thus cannot be trusted and may result in wrong monitoring result. To solve the problem of quantizing the environmental factors on QoS, this paper applies TF-IDF algorithm which calculates the impact and proposes a QoS monitoring approach called wBSRM (weighted Bayesian Runtime Monitoring) consisting of training stage and the monitoring stage. During training samples are set which satisfy QoS property as sort c0 and ones which do not satisfy as sort c1. The proposed novel approach can calculate weights and using the TF-IDF algorithm, thus helping to consider the impact of the environmental factors. Naive Bayesian network classifier considers the weight of each sample as the final monitor results. The approach was validated by set of experiments performed on open source network data and randomly generated data. Results displayed that the proposed approach is more efficient as compared to the current QoS techniques.

Several approaches for probabilistic quality property have been proposed before. But each of them had some or the other problem like statistical analysis was not done due to which the actual and the expected values were far apart, continuous monitoring was not supported, prior probability of a service could not be predicted etc. Moreover, none of the approaches worked upon previously had any consideration for environmental factors.

Naive Bayesian classifier:

Based on the sample data that is to be classified the probability of occurrence of different sorts is computed, out of which the sort with the largest probability is considered for classification. The approach in this paper uses the Bayesian formula to estimate a random variable based on implementing samples. Term Frequency-inverse document frequency (TFIDF) technique is applied in this paper.

Weigh Bayesian Runtime Monitoring Approach wBSRM:

The wBSRM modules include: Bias classifier being constructed, where the samples which are lacking information are removed, Computing the weight of factors via TF-IDF where the primary factors considered are network performance, user location and server location, matching the environmental factors where a weight is assigned to environmental factors, monitor the system, where all the modules are considered to report the monitoring results. The paper then talks about the theoretical description of wBSRM.

Algorithm Description:

wBSRM works in three steps as follows:

(a) Compute the prior probability

(b) Compute the weight of environmental factors via TF-

IDF algorithm.

(c) Apply the Bayesian decision algorithm by taking in consideration the environmental factors as well as the prior probability

Experimental Evaluation:
The paper then tests the wBSRM algorithm against the previous approaches to check if it has achieved its purpose. The data set used was a dataset provided by the Chinese University of Hong Kong which includes data of 339 users and 5825 services in the real world. The three primary questions that were addressed are:

(a) Is the monitoring result affected by environmental factors?

When the environmental factors considered are IP address, servers position and testing time it was discovered that except the service itself the person supplying the service will also impact the monitoring result. At different times when the service was used, it was discovered that the response time for a service was different for the same user. Due to the 0-1 classification of the web service quality the sample is classified as 0 even though there were some outliers in the data due to environmental factors, such issues are not considered by other approaches.

(b) Is wBSRM more efficient than the previous approaches?

It was observed that some of the previous approaches could not even track the violations tracked by wBSRM, others could track but at a slower rate.

(c) What is the time requirement of wBSRM?

Time taken by wBSRM can be quantized in two parts: monitoring time and training time. Training time is short in comparison to previous approaches. The time taken in monitoring is not the shortest compared to previous approaches but it is still very good considering the amount of computation that is done.

The paper proposes a novel approach for QoS monitoring which also considers the impact of the environmental factors on the QoS. By testing the algorithm on real and stimulated data sets it was concluded that the wBSRM is a very effective approach. Though the paper puts a novel point in the picture to consider environmental factors for QoS monitoring, we personally feel that it puts an additional burden on the user about the choice of prior property which can in turn affect the entire efficiency of wBSRM which needs to be worked upon and give proper guidelines as to how to assign it.

3. PLAN OF IMPLEMENTATION

After considerable survey of the work done in QoS services, we intend to work on the Feature Extraction domain of the QoS. We will implement the paper [3], since it implements a novel way to identify the QoS of a web service based on the user reviews. The QoS ranked by users will be the most accurate information if it's from legitimate users. We plan to proceed with implementing the research paper as follows:

1. Research the data sources to implement this algorithm.
2. Cleaning the reviews to remove invalid data.
3. Identify the sentiment of a review and a rating to check if they are in sync, either manually or by using an algorithm such as bag of words and then categorizing the sentiment.
4. We will then research about implementing the logistic regression model to identify the key features
5. The next step will be implementing the *l1* regularization to reduce the feature space by eliminating the weak features.
6. We will then test the accuracy of the features with comparison to previous approaches and will also test the sentiment evaluation in comparison to previous approaches

4. WORKLOAD DISTRIBUTION

1. Amruta Deshpande
   (a) Read the papers [7] [6] [4] and summarized them
   (b) Read and implemented the usage of Logistic Regression Algorithm. A learning model is designed to determine whether a review is positive or negative. Further,*l1* Regularized Logistic Regression is implemented. The confusion matrix and classification report is generated for both the algorithm in order to compare the two algorithm.
   (c) Worked on the abstract and Introduction part of the paper.
2. Amit Shah
   (a) Read the papers [3] [10] [13] and summarized them
   (b) Performed crawling on `www.sitejabber.com` to collect the user reviews (i.e. data) using a online crawler `www.apifier.com` and also cleaned the data by removing punctuations and attributes which were invalid for the sentiment analysis
   (c) Integrated the entire program code viz data collection, preprocessing and *l1*-RLR result.
3. Chirag Kular
   (a) Read the papers [5] [2] [11] and summarized them.
   (b) Performed preprocessing of data using python's nltk module and also created the frequency matrix of all the nouns and adjective for all the reviews
   (c) Created the project report using online latex `www.overleaf.com`

5. EXPERIMENT AND RESULT

In our experiment, We are considering only one section of dataset i.e. user reviews of computers. The different parameters considered from our dataset is shown in Table 1.

Table 1: Experiment Dataset

Parameter	value
# of services	15
# of reviews	1000
# of positive reviews	554
# of negative reviews	446
# of Features Extracted	554

We are evaluating the effectiveness of l1-RLR by comparing the results with frequency based method and logistic regression.

Table 2: Extracted Features: Frequency based method

Positive		Negative	
frequency	feature	frequency	feature
149	product	397	order
120	computer	215	computer
117	service	216	service
101	customer	208	ship

Table 3: Extracted Features : l1-RLR

Positive	Negative
feature	feature
good	bad
great	repair
price	refund
best	ship

As shown in the table 2 and 3, frequency based method contains terms that are not relevant to the quality of service. It contains terms like service, product, customer, order etc. which do not define the quality measure of the service whereas l1-RLR algorithm extract those feature that are more sentiment oriented. The features extracted here are good ,bad, refund, ship, price that directly correlates with quality. Thus, from above extracted features we can concluded that l1-RLR gave more relevant feature as compared to that of the frequency-based features.

Table 4: Recall Comparison of Feature Extraction Methods

Methods	K=30	K=40	K=50
l1-RLR	0.31	0.40	0.5
Logistic Regression	0.27	0.29	0.34
Frequency based method	0.27	0.31	0.31

In this paper, for common basis of comparison recall@K is used. l1-RLR is compared with other feature extraction based approaches like frequency-based approach and Logistic regression. Recall @k compares the features extracted with the ground truth features which are computed manually. As shown in the table 4, as the value of K increases, the accuracy of other three approach increases. With the variation in K value, l1-RLR always outperforms as compared to that logistic regression and the frequency based models. l1-RLR computes only those feature that have low p-values implying coefficient value to be zero thereby making feature more significant. Due to this reason,l1-RLR always outperforms.

Table 5: Accuracy Comparison for Sentiment Analysis

Methods	Accuracy	F-score
l1-RLR	0.76	0.77
Logistic Regression	0.61	0.67

We can calculate the accuracy and F-score of the feature extraction algorithms and can determine their performance effectiveness by comparing the results. Table 5 shows the comparison results and it can be seen that the performance of l1-RLR is better than logistic regression and thus l1-RLR can be used for feature space having large size.

6. CONCLUSION

We implemented the approach given in [3] and verified that l1-RLR extracts the most significant features and also performs better than logistic regression and frequency based methods.

7. REFERENCES

[1] D. U. Gamage, L. S. Gallege, and R. R. Raje. A qos and trust prediction framework for context-aware composed distributed systems. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 41–48, 2015.

[2] K. Lee, J. Park, and J. Baik. Location-based web service qos prediction via preference propagation for improving cold start problem. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 177–184, 2015.

[3] X. Liu, A. Kale, J. Wasani, C. C. Ding, and Q. Yu. Extracting, ranking, and evaluating quality features of web services through user review sentiment analysis. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 153–160, 2015.

[4] Y. Ma, S. Wang, F. Yang, and R. N. Chang. Predicting qos values via multi-dimensional qos data for web service recommendations. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 249–256, 2015.

[5] Y. Wan, L. Chen, J. Wu, and Q. Yu. Time-aware API popularity prediction via heterogeneous features. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 424–431, 2015.

[6] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang. Qos prediction of web services based on two-phase k-means clustering. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 161–168, 2015.

[7] W. Xiong, B. Li, X. Cui, Y. Ma, R. Yang, and P. He. A learning approach to the prediction of reliability ranking for web services. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York,*

NY, USA, June 27 - July 2, 2015, pages 169–176, 2015.

[8] C. Zeginis and D. Plexousakis. Monitoring the qos of web services using slas, 2010.

[9] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. *IEEE Transactions on software engineering*, 30(5):311–327, 2004.

[10] P. Zhang, Y. Zhuang, H. Leung, W. Song, and Y. Zhou. A novel qos monitoring approach sensitive to environmental factors. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 145–152, 2015.

[11] Y. Zhao, Q. Pi, C. Luo, and D. Yan. Capred: A prediction model for timely qos. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 599–606, 2015.

[12] Z. Zheng, H. Ma, M. R. Lyu, and I. King. Qos-aware web service recommendation by collaborative filtering. *IEEE Transactions on services computing*, 4(2):140–152, 2011.

[13] J. Zhu, P. He, Z. Zheng, and M. R. Lyu. A privacy-preserving qos prediction framework for web service recommendation. In *2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015*, pages 241–248, 2015.