International Journal of Web-Based Learning and Teaching Technologies
Volume 15 • Issue 1 • January-March 2020

Trust Decision Model and Trust Evaluation Model for Quality Web Service Identification in Web Service Lifecycle Using QSW Data Analysis

Gaurav Raj, Punjab Technical University, Kapurthala, Chandigarh, India
Manish Mahajan, CGC College of Engineering, Landran, Mohali, India
Dheerendra Singh, Chandigarh College of Engineering and Technology (Degree Wing), Chandigarh, India

ABSTRACT

In secure web application development, the role of web services will not continue if it is not trustworthy. Retaining customers with applications is one of the major challenges if the services are not reliable and trustworthy. This article proposes a trust evaluation and decision model where the authors have defined indirect attribute, trust, calculated based on available direct attributes in quality web service (QWS) data sets. After getting training of such evaluation and decision strategies, developers and customers, both will use the knowledge and improve the QoS. This research provides web-based learning about web service quality which will be utilized for prediction, recommendation and the selection of trusted web services in the pool of web services available globally. In this research, the authors include designs to make decisions about the trustworthy web services based on classification, correlation, and curve fitting to improve trust in web service prediction. In order to empower the web services life cycle, they have developed a quality assessment model to incorporate a security and performance policy.

KEYWORDS

Classification, Correlation, Curve Fitting, Decision Model, QoS, QWS Dataset, Trust Evaluation Model, Web Service Lifecycle, Web Services

INTRODUCTION

Web based learning is very helpful in all areas where ever need of training is required to be latest and updated. Our research is related to the field of information technology project management and training where developer need to work in user or customer friendly development environment. which should be vital, high quality and of a scholarly nature to advance the knowledge in the information technology project management field. Due to high economic impact of global markets and the agility of business processes, there is need to increase in the number of published web services. Web service clients having technical hitches in opting the suitable and economically effective provider with required set of web services with consumer's preference. Trust which is a degree of confidence, is required to come over this technical hitch.

In our research objective, we have presented a web-based self-learning technique for a customized provider list for selective types of web services. Web supporting WS simulator which can be design

DOI: 10.4018/IJWLTT.2020010103

on the basis of our research work, can be very helpful for making consumer aware about web service and its proper and effective utilization. This list will be automatically updated, and customer need to understand the usability and working of the list which can be done through understanding of web service lifecycle, means of trust we have focused in improving the service quality and assuring the security which can be possible through safe and secure web service communication in between users and developers. We had defined our research objectives and efforts in the fields of improving the quality of web service selection and its lifecycle. We had done literature study of various web service lifecycle and development frameworks, composite web services and their area of implementation as complex web application. Through research survey we had found that number of works are doing in the field of service prediction but there are the scope of trust based web service selection. This can be done only if we calculate trust based on available attributes from datasets of web service quality. We had selected QWS dataset for these direct attributes and use the mathematical derivation as defined for trust calculation. For validating the Trust Evaluation Model (TEM), we had used the RapidMiner tool where first we had analyse the relation between the provided attributes and come to a conclusion with a mathematical equation for calculating trust as a derived attribute from direct attributes QWS data set. To validate this trust equation, we analysis correlation metrics to observe the correlations between direct parameters and use it in TEM for evaluating trust values. In extension of that we had used classification and clustering in our research where first we compare the different classification techniques and find the suitable technique for implementation in TEM. For Classification, we have analyzed k-nearest neighbor, Naive Bayes classification model, decision tree and rule induction model techniques which is demonstrated and discussed in implementation and result analysis part of the article.

The Major focus of this paper is to evaluate trust and to design quality assessment model for QoS aware web service prediction and updating Consumer specific Customized Provider List. The existing models reflect on numerous permutations/combinations of QoS metrics to calculate the trust of web service. However, very few of them did consider the correlation among the different metrics. As per above discussion, we had also focused in correlation but within the metrics to generate improved method for calculating trust score which reduce the chances of wrong estimation of confidence in computation.

THEORETICAL BACKGROUND AND LITERATURE REVIEW

The web service trust in lifecycle has number of existing literature where quality is measured as the metrics to evaluate the trust of web services based on possibility. Web Service Lifecycle (Raj, Singh, & Bansal, 2014) has been proposed in number of literatures include the phases as defined as development, deploy, maintenance, archive and destroy. Web services are micro processes relay on open standards for communication to find the inter-operability to prevent the incidence of a long time dependency over the provider (Xenos, Stavrinoudis, & Christodoulakis, 1999).

Web Services put efforts as guided by service-oriented architecture which is an idea of service operation and execution of any application. SOA gives assured description to WS and its trust from service design and implementation (Jaiswal, Arunima, Raj, & Singh, 2014).

Web Services can be composed, self-contained and modular design. Services can describe methods called by another web application or service via putting attribute and fetching information from existing methods. Combined services can be assumed as a composite service for a large and complex web application. Web Service user request for service which is executed at web server.

Trusted Web Services which is in demand needs to describe in a vast variety of service domain. Trusted Web Services are dominating in the area of e-commerce as well as m-commerce and lots of research are developing and under process (Yao, Sheng, & Maamar, 2012; Rotter et al., 2017; Alrifai, Skoutas, & Risse, 2010).

Figure 1. Security framework for WS lifecycle

Security Framework				
Phase 1: WS Development (Locally Without Cloud Service)	Phase 2: WS Deployment (Centrally Over Cloud Using Cloud Services)	Phase 3: WS Maintenance	Phase 4: WS Archive	Phase 5: WS Destroy
1. Requirement Collection 2. Planning and Design 3. WS Development 4. WS Testing	1. Virtual Infrastructure Selection 2. Platform Selection 3. SLA Verification and Validation 4. WS Installation	1. Continuous Feedback 2. Problem Identification 3. Problem Domain Selection 4. Problem forwarding 5. Provide Solution	1. Secure Storage of WS 2. Problem Solution Synchronization 3. Disaster / WS Loss Identification 4. Start Recovery from Secure Storage	1. Uninstall the WS 2. Uninstall the Platform 3. Delete Virtual Machine 4. Free Virtual Resources 5. Update Resource Repository

In Non-functional properties, quality is the major system parameter defined for better utilization and clients satisfaction in the web service scenarios. Quality is an essential requirement to build and is an important activity for a project management at the time of planning, design, development, deployment, and maintenance. In Figure 1, the security of the web service lifecycle is the first priority as it is directly linked with the client's request and response communication. If there are any issues in quality, the business will suffer and there will be a loss of clients.

Hence, it is completely deficient to presume that the system is having predefined and set quality parameters as per standard. We need to review its implementation in between after a unique time interval and design a framework and proposed an architectural model to reuse these web services by improving on the bases of collected datasets, feedbacks and analysis done. So there is requirement of finding suitable dataset which will provide reliability, response time, latency, throughput, availability, etc., as direct attributes or properties which supports at runtime in Quality of Web Service (Roubtsov, Telea, & Holten, 2007; Sharma, Mishra, & Tiwari, 2016). To improve performance, the imperative factors for QoS are Response Time (RT), Throughput (TP), Reliability (RL), availability (AV) and Latency (LA) (Al-Masri & Mahmoud, 2007; Karimi, Isazadeh, & Rahmani, 2017; Mishra & Raj, 2017).

With large increment in demand due to scalability, not only service execution rather than performance is also expected. To deliver QoS is a decisive and significant objective of current researchers. Web services which are used in web application with different characteristics, needs to compete for limited and secure web resources which are accessed dynamically from different locations via internet. It is a big failures in delivering of web service due to DDOS attack which creates the need for improvement in QoS standards, i.e. WS- Standards (Salas & Martins, 2014; Al-Masri & Mahmoud, 2007).

ISSUES AND PROBLEM IDENTIFICATION

A model was required to answer the issues of security and quality while also making it familiar and advertised in between consumers and developers to make optimum service utilization. There is need of a recommendation system based on trust between web service and users. We have found the issue in selection of a trustworthy web service which is not possible without making consumer aware about the quality, trust and security so there is need of a learning system or information system for making latest information available to consumer by monitoring and listing the trusted web service in user reach. Web-based support is required for publishing these web services list in which cloud based trusted web service management and monitoring is not available. For understanding QoS we need to consider the latest parameters and updated datasets. In this field dataset which is available to us are not updated which need to be based on current analysis and consider it for regular improvement in dataset and training of prediction system based on ANN. There is lack of effective trust-based web service development, prediction and selection in web service lifecycle. From the quality web service perspective.

PROPOSED METHODOLOGY

Trustworthy Web Service (TWS) Selection

The monitoring is to identify issues and minimize the effort for solving them . It also reduce fixing time by effective time measurement (Govindaraj & Jaisankar, 2017; Raj, Sarfaraz, & Singh, 2014; Uusitalo, Karppinen, Juhola, & Savola, 2010). We can classify the monitoring approaches as follows:-

a. **Proactive monitoring:** A proactive strategy is desired, as opposed to reactive monitoring and performance management to avoidance strategy.
b. **Reactive monitoring:** Real time information is required to apply this monitoring model. It also requires prioritization and escalation processes.

We have designed a layout and data flow in-between processes which is an essential element of software engineering. We can bifurcate this data flow into two aspects. In Figure 2, first, the service UI/UX design which constitutes the GUI design helps in user interaction and fixes the user requirements and nature of the service response which is the gateway to service functionality and an indication of knowledge and the notion of functionality. Second, the design of service functionality of the working part and the business logic are executed inside.

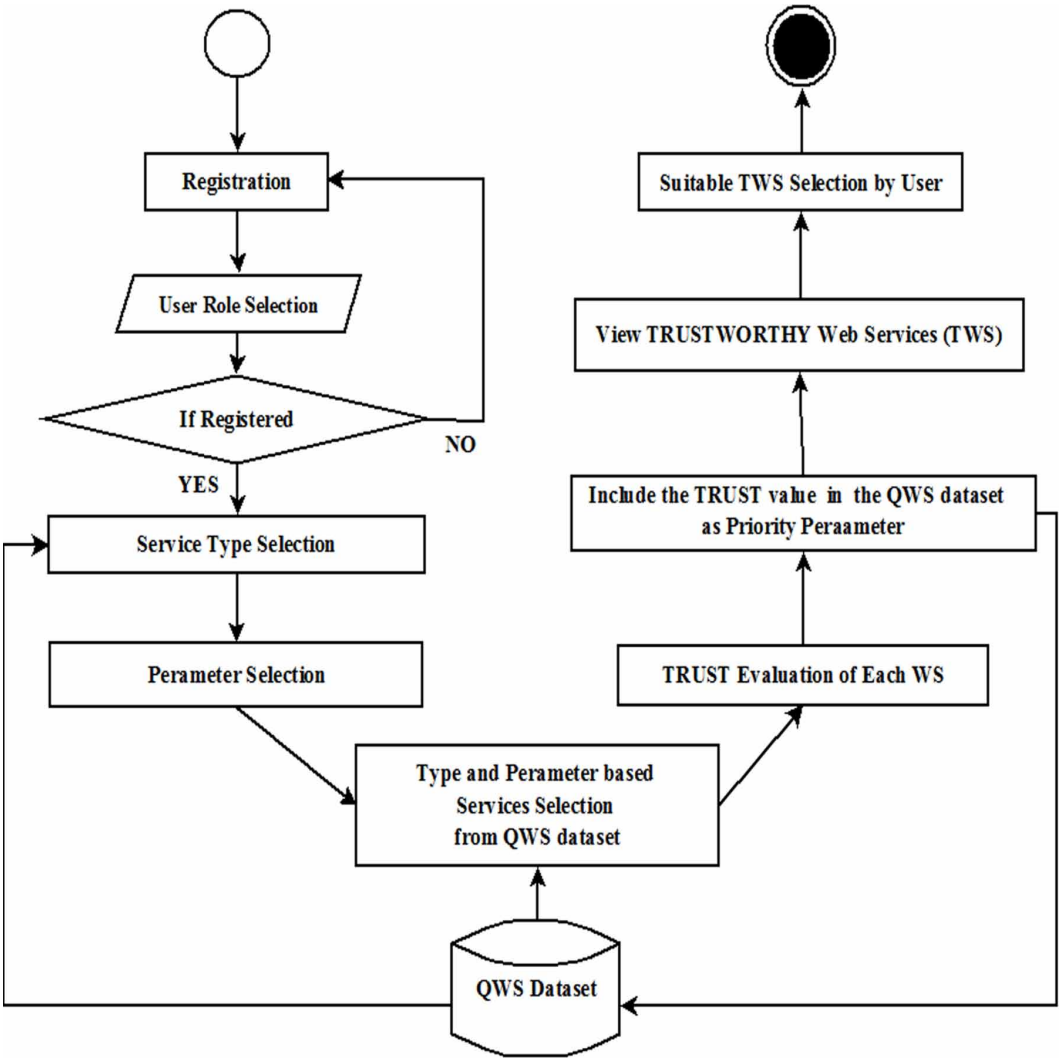
Trustworthy Web Service Request-Response Monitoring

Monitoring of request and response in TWS is a SOAP based complex communication system, which needs to be user-friendly and accessible. TWS monitoring is important to help in the recognition of information flow and performance which is useful in identifying drawbacks in service delivery to all user/client/broker/ provider and can be used to improve in web service quality provided. It will be very useful in managing service life cycle in secure manner. (Liu, Jing, & Cheng, 2016; Raj, Sarfaraz, et al., 2014; Raj, Singh, et al., 2014; Uusitalo et al., 2010)

System includes the user-controlled GUI-based application where the roles of users are classified as user, broker and provider. The purpose is to monitor the requests/response of parameter-based TWS selection.

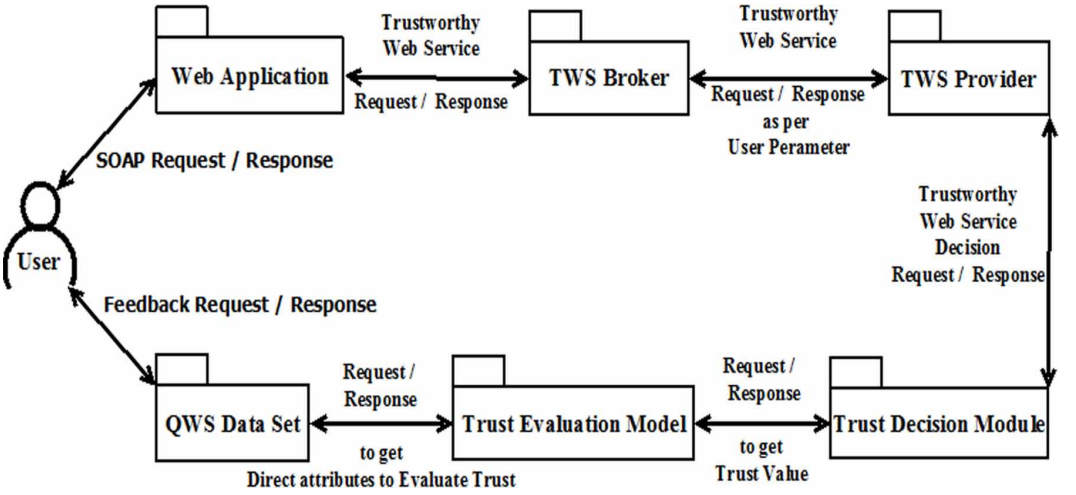
a. **User Role:** To select and use the web service characteristics in Web Application from TWS Broker and Provider which has been updated through Trust Decision Model (TDM) and Trust Evaluation Module (TEM).

Figure 2. Flow for trustworthy web service (TWS) selection



b. **Provider Role:** Web services providers can be updated by removing / adding trusted WS providers in TWS Broker. As we are including cloud there are different type of service providers as Local, Remote and Hybrid. Local which have been hosted locally as private cloud serve. Remote services are global services whose information we are having through QWS data set. Hybrid services includes the services which are customised based on user parameters and demand for effective and satisfactory use of resources by existing user as a premium paid service.
   i. *Service List*: Service list which can be used to monitor recent trends and demands in the market. It includes a description and corresponding parameters provided in the service.
   ii. *Service Feedback*: for each service, after use consumer has to fill feedback to end the process. This new option allows the user to strict in giving feedback which is most important for updating and analysing quality of web service.
   iii. *Analysis and Reports*: Useful for provider to increase satisfaction of consumers by getting analysis of recent updates and stats of the system.

Figure 3. Package design of TWS request and response monitoring in quality assessment model



Through Figure 3, we have defined the design of the research environment which will fulfil the objective of our research aims. As we decided to identify the trust evaluation model of the web service lifecycle, we had shown here complete communication between the customer, broker, provider, and other stockholders. They will deal with all aspects which are required to find the trusted web service using a quality and security policy consideration where we use WS-security and a WS-quality framework to select the policy set in a policy set selection mechanism.(Al-Masri & Mahmoud, 2007; Jaiswal, Arunima et al., 2014; Ladan, 2011; Mell & Grance, 2011; Raj, Sarfaraz, et al., 2014; Salas & Martins, 2014; Takabi, Joshi, & Ahn, 2010)

Quality Assessment Model using QWS Data Set

We have explained the detailed implementation process of quality assessment model in which includes the following models

a. QWS Based Trust Evaluation Model (TEM)
b. Trusted Decision Module (TDM)
c. Effective Policy Set Selection (PSS)

We had also shown the process of evaluation of trust through QWS dataset and its internal structure is also explain in detail in Figure 4 and 5. In our main Module where we utilize both Policy Set Selection(PSS) and Trust Evaluation Model(TEM) results, decide the trust parameter of each WSDL link based on QWS dataset analysis and its own analysis of customer dependent web service parameter selection criteria. After the decision through the model, the Customized Provider List (CPL) will be updated and provide Trusted web service information to customer to access Methods and WSDL Links of web service stored in CPL (Emelie, Mäntylä, Runeson, & Borg, 2014; Liu et al., 2016; Liu, Chu, Jia, Shen, & Wang, 2016; Manolescu et al., 2005).

Figure 4. Detailed Implementation of quality assessment model using QWS Based TEM, TDM and PSS

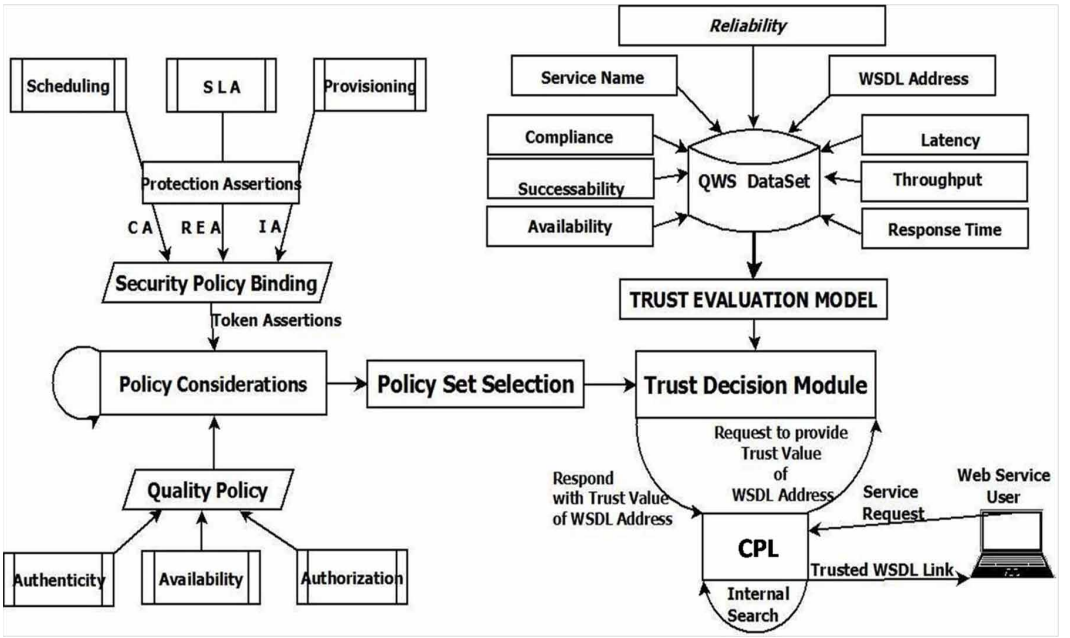


Figure 5. Trust evaluation model (TEM) for QWS dataset

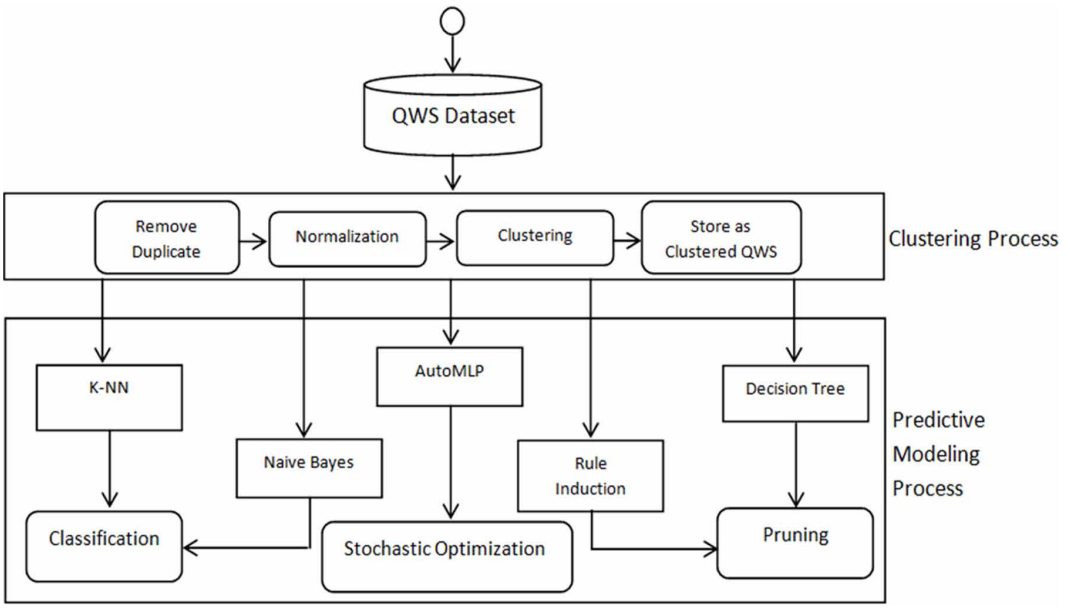
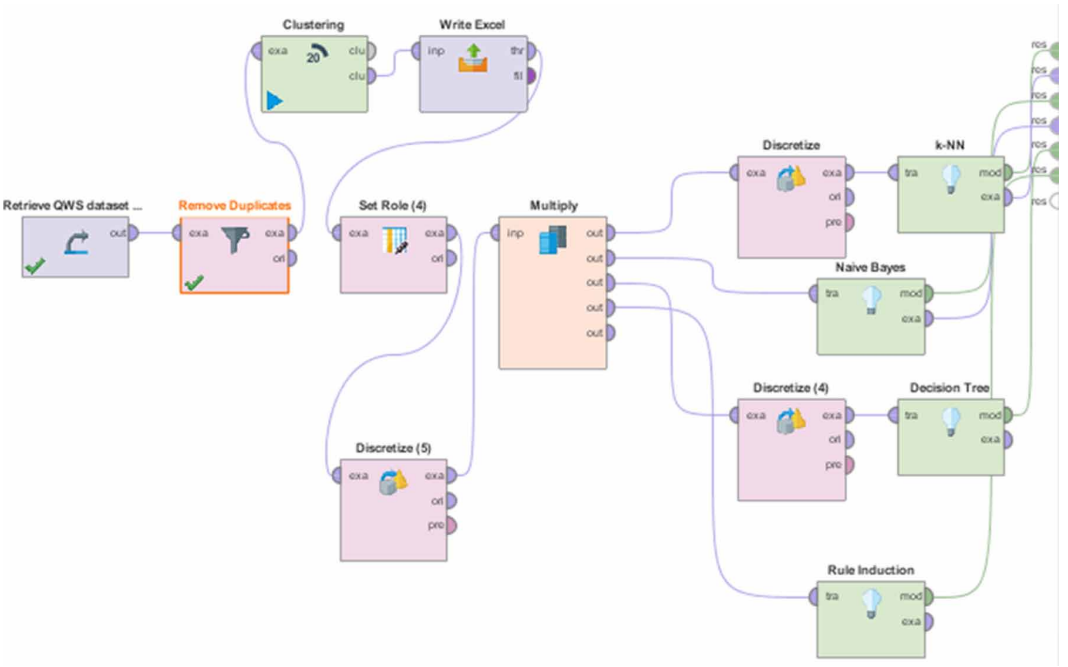


Figure 6. Clustering and Service Classification process of QWS dataset on RapidMiner for TEM using k-NN, Naive Bayes, Decision Tree and Rule Induction



IMPLEMENTATION AND RESULT ANALYSIS

QWS Dataset Analysis and Design

We have done data analysis in QWS dataset for calculating the trust-based benchmark in a WS selection which includes 2,507 WSDL web links and related QWS constraint that has been carried out in current years through WSB framework. In this research, we had selected the dataset of different Service Names, WSDL links and their measuring parameters i.e. response time (ms), throughput (invokes/sec), reliability (%), availability (%), latency (%). Our motive is to select trustworthy relation from different WSs.

The QWS, a standard dataset produced and uphold by University of Guelph, Canada, used in our research work, Clustering and Classification has been applied on QWS datasets through the RapidMiner tool. We had designed the process consist of connecting the operators in a meaningful way. We had implemented four parallel processes as per Figure 6 which will illustrate different classification and simple distributions in ranges and find facts and relational analysis between throughput, response time reliability, latency and availability which is shown as response time vs. throughput, reliability vs. throughput, reliability vs. latency, reliability vs availability in Figures 7, 8, 9, and 10, respectively. Through the relation shown in Figure 7, depicted that both of them are inversely proportionate to each other as response time increases, it's throughput surely decreases. Next Figures 8, 9, and 10, show reliability with throughput, latency, and availability. As reliability impact on trust in different aspects as you increase throughput, latency or availability, it proportionally impacts on reliability. This will be very helpful in understanding that trust which is proportionally impacted by reliability, is formulated based on these statistics. Relation in response time and throughput is as we implement decision tree, it helps in finding control for selecting range and design rules through rules Induction operator. (Govindaraj & Jaisankar, 2017; Liu et al., 2016; Mehdi, Bouguila, & Bentahar, 2016; Tang, Dai, Liu, & Chen, 2017)

Figure 7. Response time vs. throughput

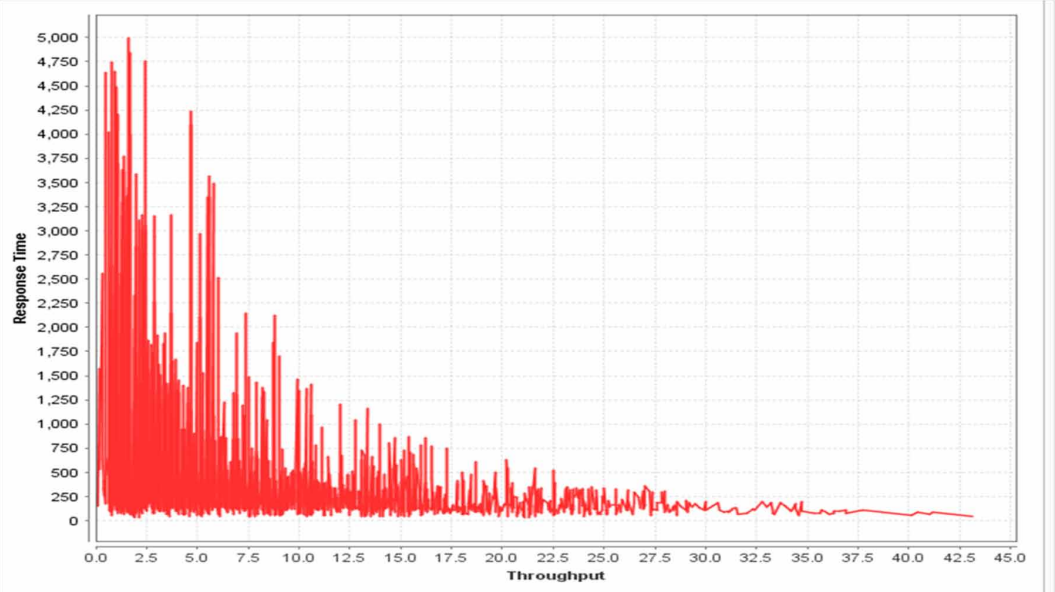
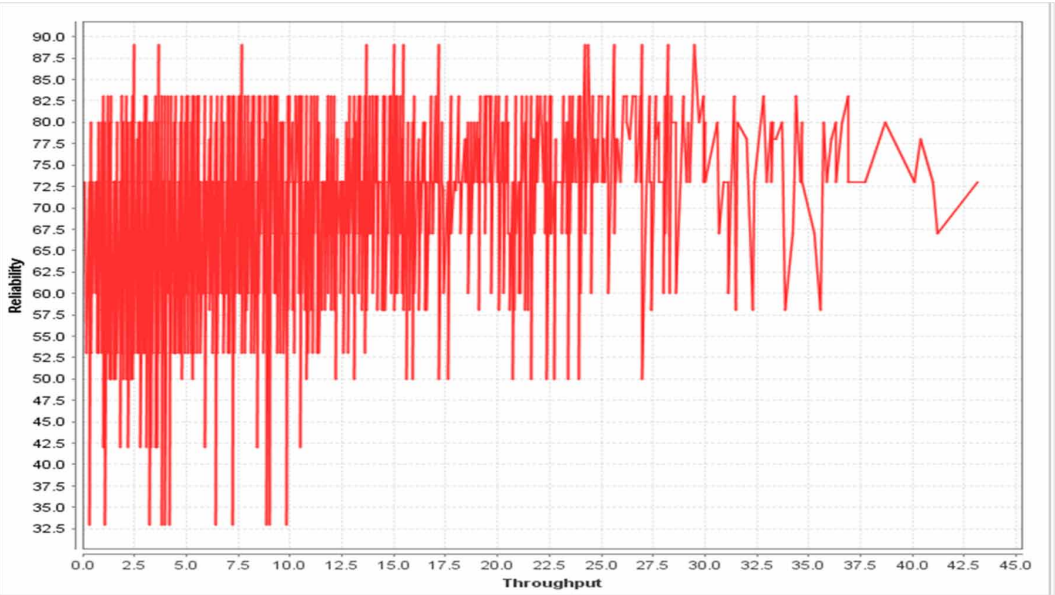


Figure 8. Reliability vs. throughput



Mathematical Model for TRUST Calculation

We have received 194 records after implementing cleaning and removing duplicate data. We have also removed the data rows with missing attributes which reduce the total size of record as 167 individual rows as it denotes different WSDL link as well as web service. We have analyzed the statistics of the parameters and found its min, max and average values and store it in table for further use. Clustered

Figure 9. Reliability vs. latency

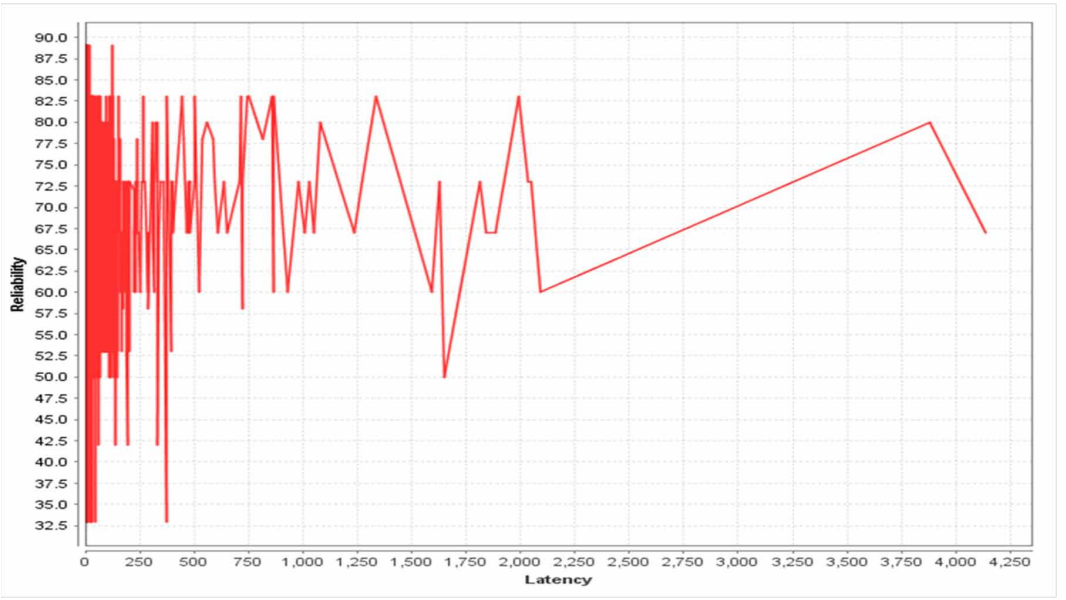
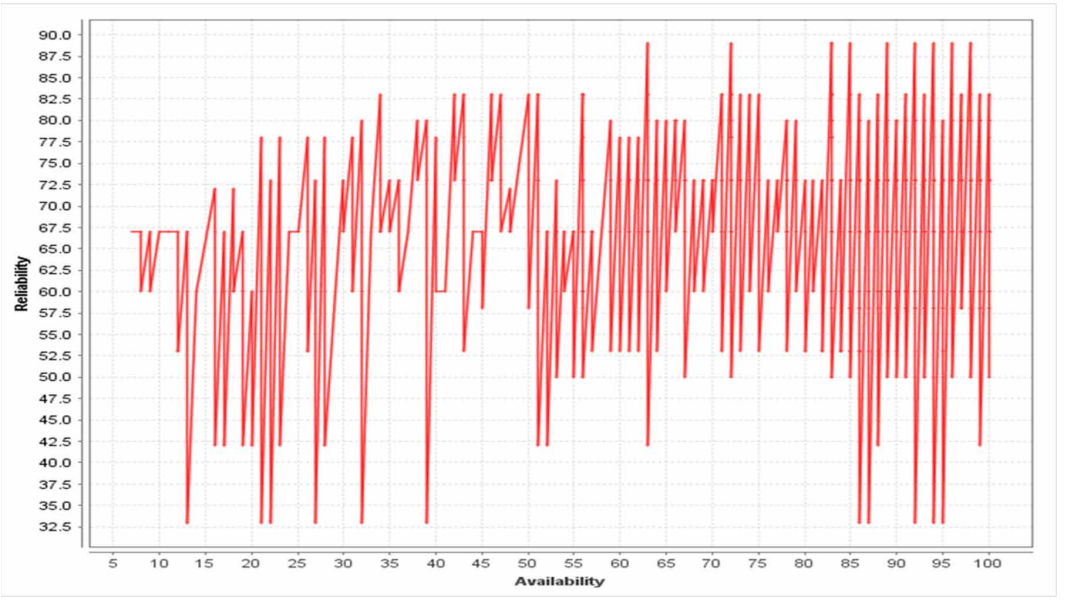


Figure 10. Reliability vs availability



QWS data set can use to classify the web services, where it is clearly stated through above graphical analysis that throughput and response time are inversely proportional. The process implemented here consists of the operators and works as follows:

The "read excel" operator fetches the data in the form of input data from the excel sheet of QWS dataset. The data pre-processing operations are performed by the next two operators, i.e., "Remove Duplicates" and the "Normalize" operators. These are used to clean the data of any repeating values

and normalise the dataset respectively. The clustering operator is used to perform clustering via k-means which is widely used and primary algorithms for clustering. The value of k defines the number of created clusters. Here, we had taken k as 4, Max Runs as 10, Measure types as Bregman Divergences, Divergence as Squared Euclidean Distance and 100 maximum optimization steps to generate the results.

In further process of analysis, we have to apply data mapping between all parameters but here we had focused on three parameters as response time, throughput and reliability. We have found the following statistics:

On the basis of statistical data in Table 1 and 2, we have classified the range in five slabs as Lowest, Above Lowest, Average, Above Average, Highest. Through this classification we had generated 4 clusters for each parameter with defined boundary limits i.e. *Cluster_1{Min: Lowest, Max: Above Lowest}, Cluster_2{Min: Above Lowest, Max: Average}, Cluster_3{Min: Average, Max: Above Average}, Cluster_4{Min: Above Average, Max: Highest}.* As there are five parameters are considering for analysis, number of test cases based on permutations and combinations are very large. For finding the correlation between *Throughput* (T_h), *Reliability* (R_e) and *Response Time* (R_t) *Availability* (A_v), *Latency* (L_c) which have a very complex relationship, we need to do analysis through correlation metrics which we had generated with the help of RapidMiner as following table. Through the Correlation metrics, In Table 3, We had stored the observations where second best correlation of each parameter is selected. Through this analysis largest correlation in between Response Time and Latency equals to 0.390, next largest is in between throughput and reliability equal to 0.255 and

Table 1. Statistical Data received for parameters in QWS data set

	Maximum	Minimum	Average
Response Time	3203	45	260.892
Throughput	29.5	0.1	10.081
Reliability	97.4	5.9	72.073

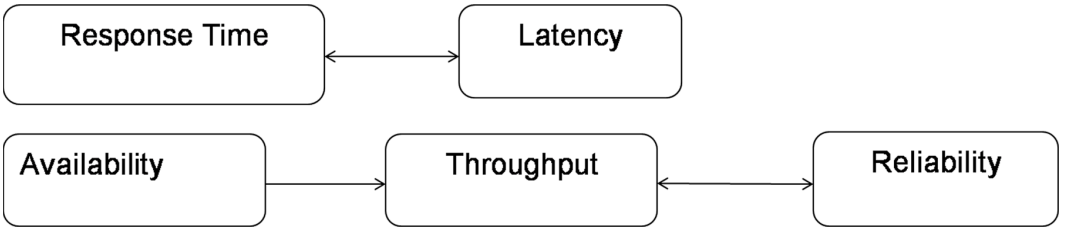
Table 2. Classification of Statistical Data of parameters in QWS data set

	Lowest	Above Lowest	Average	Above Average	Highest
Response Time	45	152.946	260.892	1731.946	3203
Throughput	0.1	5.0905	10.081	19.7905	29.5
Reliability	5.9	38.9865	72.073	84.7365	97.4

Table 3. Correlation Parameters in QWS data set

Parameter	RT	AV	TP	RL	LA
RT	1	-0.0663	-0.2530	0.0471	0.3907
AV	-0.0663	1	0.2006	0.1289	-0.0982
TP	-0.2530	0.2006	1	0.2556	-0.1450
RL	0.04711	0.1289	0.2556	1	-0.0238
LA	0.39073	-0.0988	-0.1450	-0.02388	1

Figure 11. Relations designed using correlation Metrics between QWS parameters



next largest is in between Availability and throughput equal to 0.200 (Mehdi, Bouguila, & Bentahar, 2016). This observation states the binding of parameters described in Figure 11.

Through analysis of correlation, we are testing trust based on following formulations is developed for trust calculation of each web service.

$$Trust\ \alpha\ \frac{Availabilty * ThroughPut * Reliability}{ResponseTime * Latency} \tag{1}$$

$$Trust = \mu * \frac{Availabilty * ThroughPut * Reliability}{ResponseTime * Latency} \tag{2}$$

Where μ is proportionate Coefficient which can be identified using correlations between concern attributes derived from Table. 4 as follows:

For finding the value of μ, we can use different relational correlations formulas i.e.

μ_1 = (0.2556*0.2006)/(0.3907) = 0.130
μ_2 = (0.2556+0.2006) - (0.3907) = 0.0655
μ_3 = (0.2006*0.1289*0.2556)/(0.3907) = 0.016916
μ_4 = (0.2006+0.1289+0.2556) - (0.3907) = 0.1944

μ_1, μ_2, μ_3 *and* μ_4 are different possible coefficients values which can be test and use in finalizing the trust value in TEM and further use in TDM for setting the web service trust value based on QWS dataset. In TEM, Service Classification & Clustering process of direct attributes can be analyzed using following operations in the process defined in Table 5 which shows a comparative analysis over QWS dataset and its parameters:

Distribution of range in decision tree for finding best practice to classify numerical data is shown in Figure 12 where range 1 and range 2 is described.

Table 4. Correlations between concern attributes

Proportionate correlation	Inverse proportionate Correlation
Availability ∧ Throughput → 0.2006 Availability ∧ Reliability → 0.1289 Throughput ∧ Reliability → 0.2556	Response Time ∧ Latency → 0.3907

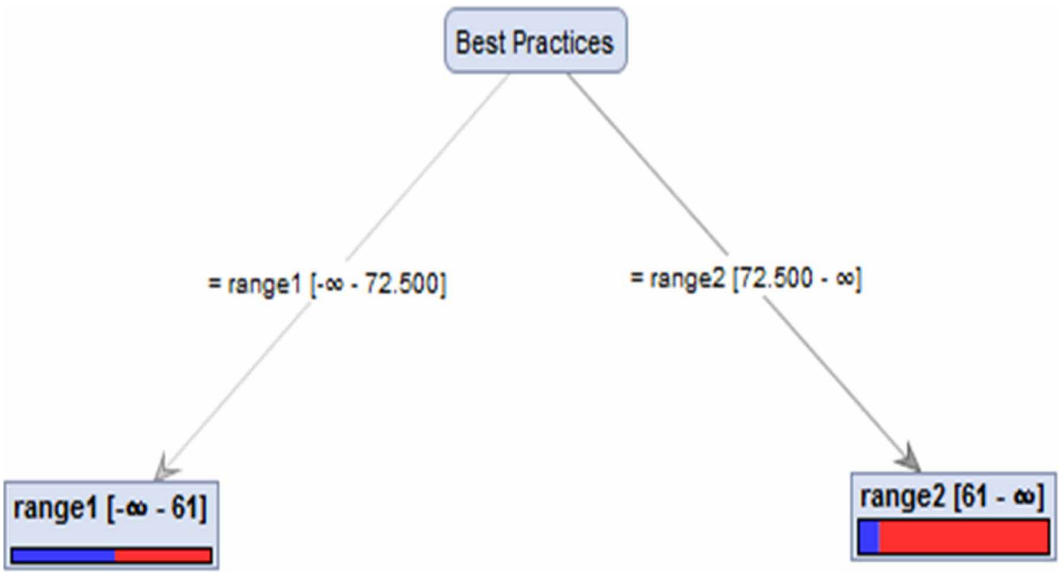
Table 5. Comparison of Classification techniques

k-Nearest Neighbor (K-NN)	Naive Bayes Classification Model	Decision Tree
This model is using for data classification and regression which is based on the input Dataset and its parameters. As implementation of K-NN model in our system, we had done classification and find the following 2-Nearest Neighboring range as in classification. The model includes 2507 samples with 11 extent as the range of classes defined by range1 [-∞ - 61] & range2 [61 - ∞]. Through this model classification, we can conclude that K-NN is reasonable and justifiable within all Considerable parameters which frequently utilized for its effortless of understanding and small computation time.	It's a probabilistic classification technique which has focused on implement Bayes' theorem. The Unique achievement of this Naive Bayes' classification technique is that it needs very tiny amount of training facts to approximation the discrepancy in parametric facts necessary for classification.	It helps in creation of a decision-based tree for nominal and numerical data classification. It's an inverted tree which has root node at the top and nodes added in downward direction in left and right. This type of data representation has number of advantages over other approaches and easy to understand. In this classification technique we predict the target based on multiple input parameter in QWS dataset. Following decision tree has been generated Best Practices = range1 [-∞ - 72.500]: range1 [-∞ - 61] {range1 [-∞ - 61] = 274, range2 [61 - ∞] = 250} Best Practices = range2 [72.500 - ∞]: range2 [61 - ∞] {range1 [-∞ - 61] = 230, range2 [61 - ∞] = 1753}

Rule Induction Model is the prediction technique using operator which works as like a learner based on propositional logic. In our simulation, rule Induction provides following rules/logics which are designed to select range using Best Practice (B_p), Throughput, Response Time and Success-ability:

1. if BP > 79.500
2. then range2 [61 - ∞] - Selection - (19 / 1552)
3. if Throughput ≤ 5.850 and BP ≤ 71.500 and Succ ≤ 88.500
4. then range1 [-∞ - 61] Selection - (144 / 2)
5. if BP ≤ 73 and BP > 71.500 and Throughput > 3.150

Figure 12. Decision tree for finding best practice for classification of numerical data



6. then range2 [61 - ∞] Selection - (2 / 95)
7. if BP ≤ 67.500 and Succ > 87.500 and Documentation > 23.500
8. then range2 [61 - ∞] Selection - (7 / 68)
9. if BP ≤ 74.500 and BP > 73
10. then range1 [-∞ - 61] Selection - (66 / 0)
11. if BP > 71.500 and BP ≤ 78.500 and BP > 76.500
12. then range2 [61 - ∞] Selection - (7 / 125)
13. if Succ > 85.500 and BP > 67.500 and Response Time > 185.335
14. then range1 [-∞ - 61] Selection - (116 / 9)
15. if BP ≤ 75.500 and BP > 71.500 and Response Time > 164
16. then range2 [61 - ∞] Selection - (4 / 78)
17. if BP > 75.500
18. then range1 [-∞ - 61] Selection - (47 / 0)
19. if Succ ≤ 90.500 and BP ≤ 71.500 and BP > 67.500
20. then range1 [-∞ - 61] Selection - (22 / 0)
21. if Response Time ≤ 151.410 and Latency ≤ 12.035 and Throughput ≤ 18.250
22. then range2 [61 - ∞] Selection - (4 / 22)
23. if Succ ≤ 93.500 and Response Time > 138.415 and Documentation ≤ 76.500
24. then range1 [-∞ - 61] Selection - (26 / 1)
25. if BP ≤ 65.500 and BP > 60.500
26. then range2 [61 - ∞] Selection - (3 / 35)
27. else range1 [-∞ - 61] Selection - (33 / 14)

It starts with the less prevalent classes where the iteration is used in algorithm and apply rule of pruning until there are no positive set of information in dataset missing or the error rate is larger than fifty percent. Number of correct data in QWS dataset as per above stated rules are 2429 out of 2501 training examples.

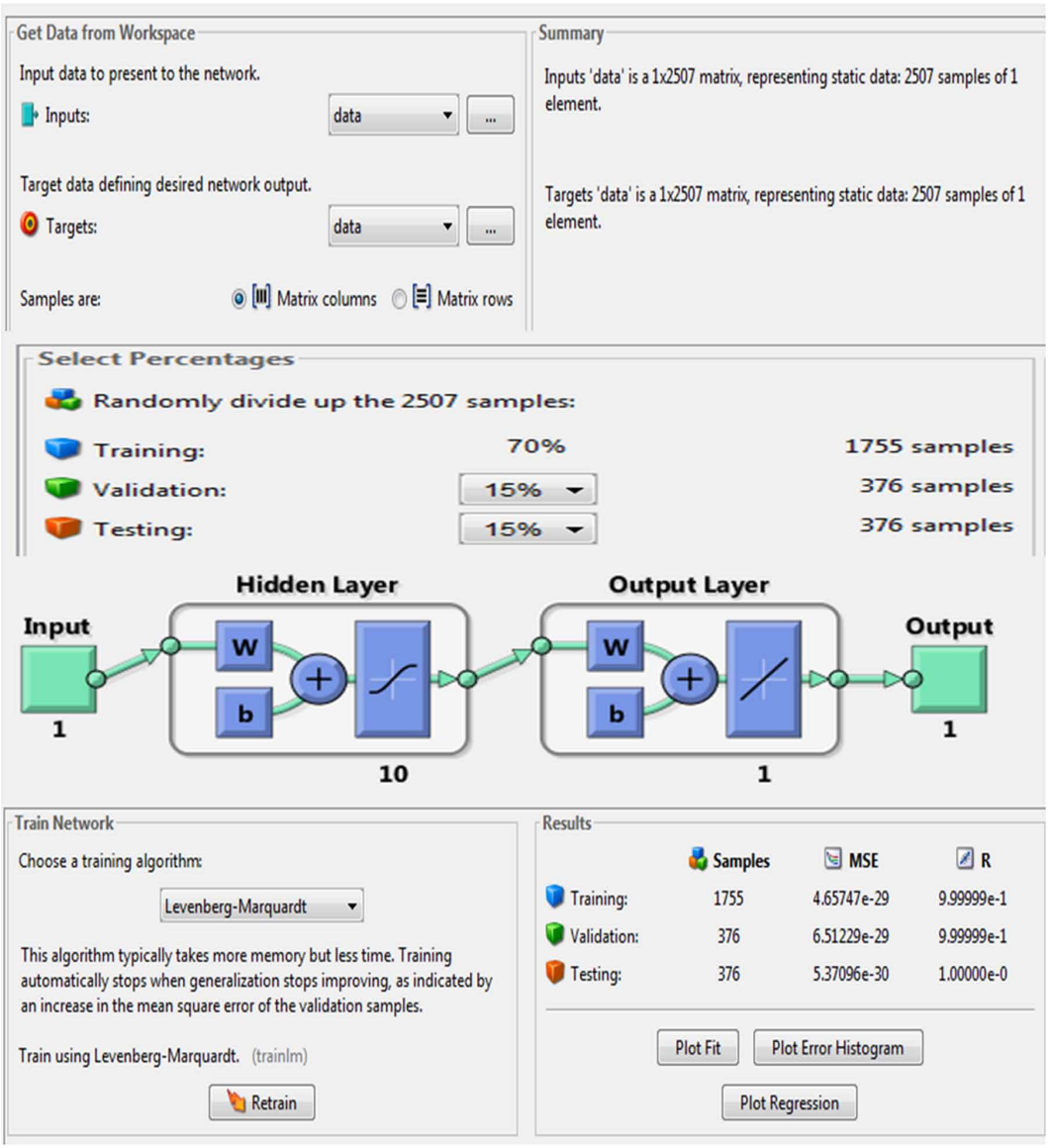
Curve Fitting Implementation

Analysis of validation and testing on QWS dataset has been done through Matlab curve fitting approach of neural network. Here, we pass all four parameters (Throughput, Response time, Availability, Reliability and Latency) as Input and find mapping of these parameter with Trust as Target or output parameter which we want to predict on the bases of formulation defined in equation number 1. Here, we want to are using this fitting approach to identify suitable scenario to predict the effective and closest Trusted value as output. Here, the size of our dataset is 2507 sample values which will be divided in training (70%), validation (15%) and testing (15%) phases.

Levenberg - Marquardt training algorithm is used which generally takes more memory but less time. This training algorithm generalize and improve process till Mean Square Error (MSE) improves. As these processes automatically stop when MSE of validation sample is increased. In results, we analyze the test output over fitting curve and error values in input parameters. We had done this testing and validating output parameter as target value. Here, we can perform exploratory data analysis over input QWS dataset and preprocess and post-process this QWS dataset. Here, we can use different library methods as linear and nonlinear methods and starting conditions used as optimized solver parameters to improve the quality of your curve fitting.

For better understanding, we had done simulation of curve fitting for training QWS parameters via fitting neural network with hidden neurons shown in Figure 13. Here, in Figure 14 fitting function describe relation between input and output values. As per QWS analysis, we find the effect of neurons in defined hidden layer which generalised the idea of selection of neurons as if we want to see the progress in performance of target data, we need to select neurons very effectively. In our customised scenario, we had analyse and find that 100 neurons help in generating less errors in generalizing

Figure 13. Fitting neural network with different hidden neurons for training QWS parameters



input data and range of output and target data is approximately -75 to 150. This range is optimised as compare to ranges in different test scenarios.

Curve fitting regularly and consistently analyzes QWS input dataset beginning from starting data to finishing data and remove poor quality values identified in curves which explained in Input-Error bar graph in Figure 15.

We have studied the error histogram in between these parameter value fitting which will be helpful to identify predicting value from target value. This error gives idea of difference between target and predicted values. Error is identified as multiplication of Target and output values. Process of identifying the error histogram is include reading data from QWS dataset which will further forwarded in next process where we identify feedforwardnet function which define hidden layer with

Figure 14. Plotting fitting function with different hidden neurons with 10,20, 40, 100, 500, 1000, respectively

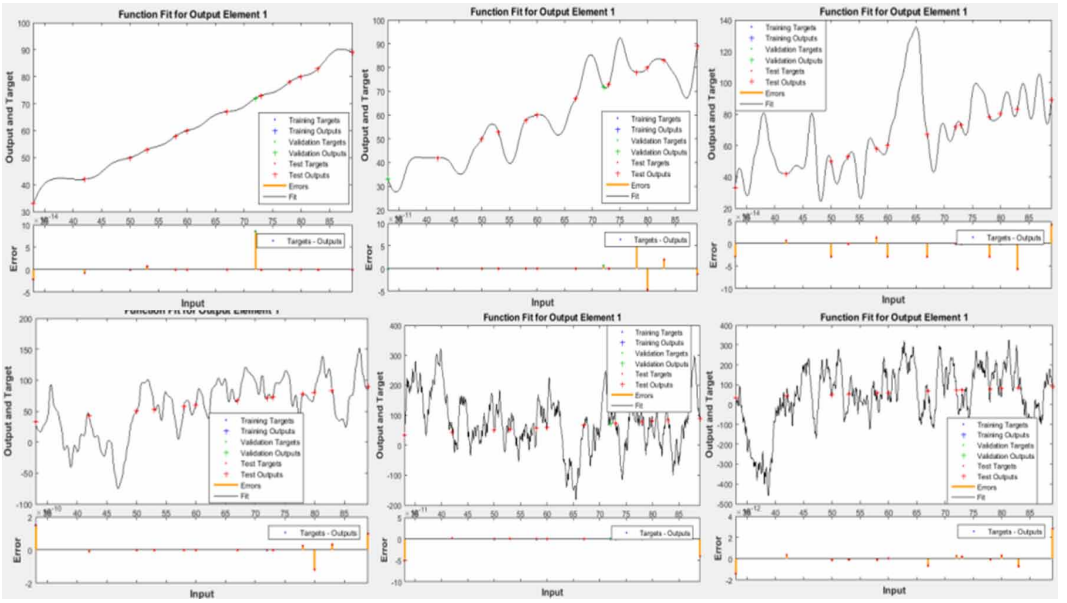
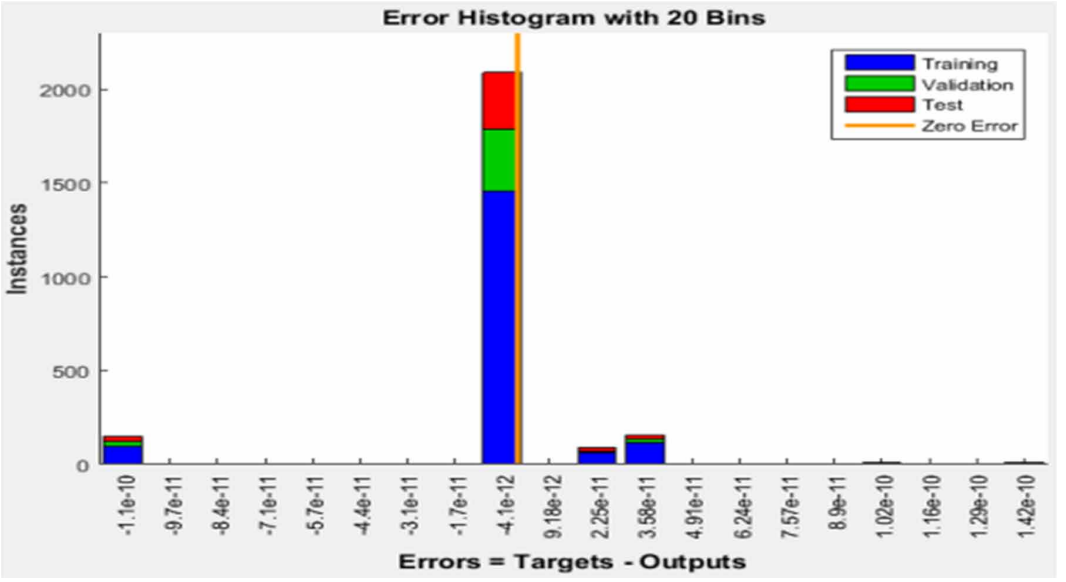


Figure 15. Plotting error histogram with different hidden neurons with 20 bins and 40 neurons



neurons and output layer. After training and validating the feedforwardnet function, we predict the responses of all input parameters over output using object of the function. In our scenario we define 6 different neurons as 10, 20, 40 100, 500, 1000 sequentially to test the optimal error histogram. By making changes in feedforwardnet function, we had analyze different scenarios and found that with 20 bin and 40 hidden nodes error is minimum and close to 1. If we increase or decrease values of neurons, this error value will increase. This minimum error value is defining that our setup is having

is trained over the scenario and now on the bases of the input parameter as response time, availability, throughput and reliability, we can predict output value as trust value.

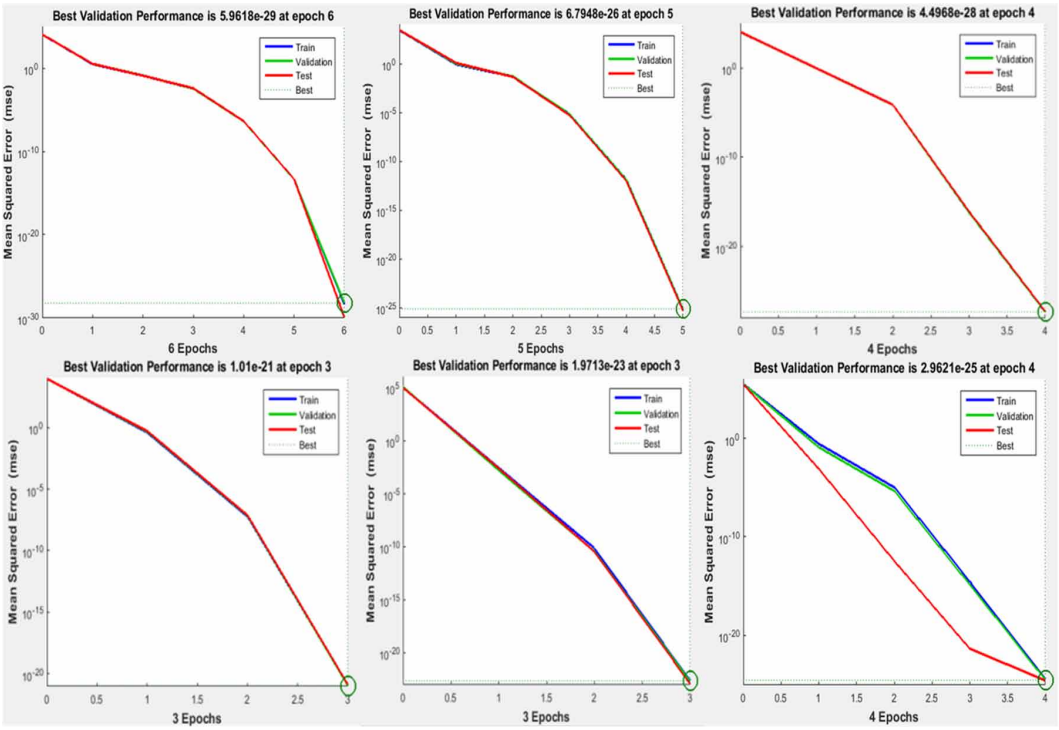
As we know that this fitting curve is a multi-layered network flow with training, validating and testing. It helps in checking and identifying the feedforwardnet performance to determine the changes need to be made in selection of neurons in hidden layer and any other training algorithm. It improves the network architecture as it contains all concerning information required to train the network as data points and test sets. If there is any need to retrain the network, we can to it using the same division of data. In our scenario, we had done testing with different number of neurons in hidden layer and summarized it as a range of 40 to 100 is suitable for getting good performance due to large quantity of neurons in hidden layer which provide better flexibility as in our simulation input parameters can optimize and best fit in target parameter.

Now, as per all discussion and analysis done in different simulations in Figure 16, we come to conclusion as to find the best suitable performance of our scenario can be find by defining the range of neurons from 40 to 100. through which very less error found in input dataset after normalization and generalization of data. It is best validation performance define in respect to mean square error in defining output as trust value using formulation defined through equation 1 and 2 with all five parameters as input i.e. *Throughput* (T_h), *Reliability* (R_e) and *Response Time* (R_t) *Availability* (A_v), *Latency* (L_c).

CONCLUSION

In concluding the research work in our paper, we have focused in designing and proposing solution for security and trust via Quality Assessment Model using TEM, TDM and PSS. TEM includes Web Service Classification & Clustering analysis (i.e. K-NN, Naive Bayes, Decision Tree Classification

Figure 16. Plotting best validation performance vs. MSE with different hidden neurons with 10,20, 40, 100, 500, 1000, respectively



Model and Rule Induction Model which is learner-based prediction technique on propositional logic). We had found the relations in between QWS dataset and calculated Trust value as direct and derived attributes used in TDM for predicting the QoS in Web Service. In order to validate the correctness of defined mathematical model, we had calculated correlation in parameters and used it in formulating Trust of WS. After Calculating Trust, we had performed Curve Fitting between direct attributes provided by QWS dataset and Trust as derived attribute using Neural Network using fitting function with different Neurons. As a testing result, we can summarize the range of 40 to 100 neurons are suitable for getting good performance which provide better flexibility and optimization into best fit values in target parameter. User will use the Customized Provider List, which is having updated list of providers with trust values as user specific priority list based on user parameter selection.

FUTURE SCOPE

This research has broad future scope as it is related to the field with uncertain growth in trusted service requirement in information and project communication management. As we are focusing on quality assessment and assurance, quality control management is also having huge scope of study in web service lifecycle. In addition, number of non-functional as well as functional issues still requires addressing and evaluate. There is a scope of improvement in cloud based virtualized environment which may be replaced with dockers and container based secure web service scanning and monitoring for service providers. In future studies our objective is to improve the monitoring of the web service lifecycle using this analysis of stated issues in terms of improvised reliability, availability and success-ability in service utilization using large dataset and deep learning implementation in designing more effectively model to calculate trust for web service. A web-supporting WS Simulator has been designed for web based self-training which train the consumer as well as developer about trusted web services and help in clustering them in a customized list of services for providers. It will be helpful in prediction, recommendation and selection of trusted web services.

REFERENCES

Al-Masri, E., & Mahmoud, Q. H. (2007). QoS-based Discovery and Ranking of Web Services. In *Proceedings of the 16th International Conference on Computer Communications and Networks,* Honolulu, HI (pp. 529–534). USA: IEEE. doi:10.1109/ICCCN.2007.4317873

Alrifai, M., Skoutas, D., & Risse, T. (2010). Selecting skyline services for QoS-based web service composition. In *Proceedings of the 19th international conference on World wide web - WWW '10*, Raleigh, NC, April 26-30 (pp. 11–20). USA: ACM. doi:10.1145/1772690.1772693

Emelie, E., Mäntylä, M., Runeson, P., & Borg, M. (2014). Supporting regression test scoping with visual analytics. In L. O'Conner (Ed.), *Proceedings of the IEEE International Conference on Software Testing, Verification, and Validation,* Cleveland, OH (pp. 283–292). IEEE. doi:10.1109/ICST.2014.41

Govindaraj, P., & Jaisankar, N. (2017). A review on various trust models in cloud environment. *Journal of Engineering Science and Technology Review*, *10*(2), 213–219. doi:10.25103/jestr.102.24

Jaiswal, A., Raj, G., & Singh, D. (2014). Security testing of web applications: Issues and challenges. *International Journal of Computers and Applications*, *88*(3), 26–32. doi:10.5120/15334-3667

Karimi, M. B., Isazadeh, A., & Rahmani, A. M. (2017). QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm. *The Journal of Supercomputing*, *73*(4), 1387–1415. doi:10.1007/s11227-016-1814-8

Ladan, M. I. (2011). Web Services: Security Challenges. In *World Congress on Internet Security* (pp. 160–163). London, UK: IEEE. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5749903

Liu, H., Jing, L., & Cheng, M. (2016). An Efficient Parallel Trust-Based Recommendation Method on Multicores. In *First International Workshop on High Performance Graph Data Management and Processing, HPGDMP '16,* Salt Lake City, Utah (pp. 9–16). ACM. doi:10.1109/HPGDMP.2016.009

Liu, Z. Z., Chu, D. H., Jia, Z. P., Shen, J. Q., & Wang, L. (2016). Two-stage approach for reliable dynamic Web service composition. *Knowledge-Based Systems*, *97*, 123–143. doi:10.1016/j.knosys.2016.01.010

Manolescu, I., Brambilla, M., Ceri, S., Comai, S., & Fraternali, P. (2005). Model-driven design and deployment of service-enabled web applications. *ACM Transactions on Internet Technology*, *5*(3), 439–479. doi:10.1145/1084772.1084773

Mehdi, M., Bouguila, N., & Bentahar, J. (2016). Trust and Reputation of Web Services Through QoS Correlation Lens. *IEEE Transactions on Services Computing*, *9*(6), 968–981. doi:10.1109/TSC.2015.2426185

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. NIST. doi:10.1136/emj.2010.096966

Mishra, T., & Raj, G. (2017). QoS implementation in Web Services selection and ranking using data analysis. In *Proceedings of the 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence*, Noida, India (pp. 537–542). IEEE. doi:10.1109/CONFLUENCE.2017.7943209

Raj, G., Sarfaraz, M., & Singh, D. (2014). Survey on trust establishment in cloud computing. In *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit*, Noida, India (pp. 215–220). IEEE. doi:10.1109/CONFLUENCE.2014.6949375

Raj, G., Singh, D., & Bansal, A. (2014). Analysis for security implementation in SDLC. In *Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit (Confluence)*, Noida, India (pp. 221–226). IEEE. doi:10.1109/CONFLUENCE.2014.6949376

Rotter, C., Illes, J., Nyiri, G., Farkas, L., Csatari, G., & Huszty, G. (2017). Telecom strategies for service discovery in microservice environments. In *Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks ICIN 2017* (pp. 214–218). doi:10.1109/ICIN.2017.7899414

Roubtsov, S., Telea, A., & Holten, D. (2007). SQuAVisiT: A software quality assessment and visualisation toolset. In *SCAM 2007 - Proceedings 7th IEEE International Working Conference on Source Code Analysis and Manipulation* (pp. 155–156). IEEE. doi:10.1109/SCAM.2007.16

Salas, M. I. P., & Martins, E. (2014). Security testing methodology for vulnerabilities detection of XSS in web services and WS-security. *Electronic Notes in Theoretical Computer Science,* 302, 133-154. Elsevier B.V.; doi:10.1016/j.entcs.2014.01.024

Sharma, T., Mishra, P., & Tiwari, R. (2016). Designite - A Software Design Quality Assessment Tool. In *Proceedings of the 2016 1st International Workshop on Bringing Architectural Design Thinking Into Developers' Daily Activities,* Austin, TX (pp. 1–4). ACM. doi:10.1145/2896935.2896938

Takabi, H., Joshi, J. B. D., & Ahn, G.-J. (2010). Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy Magazine*, *8*(6), 24–31. doi:10.1109/MSP.2010.186

Tang, M., Dai, X., Liu, J., & Chen, J. (2017). Towards a trust evaluation middleware for cloud service selection. *Future Generation Computer Systems*, *74*, 302–312. doi:10.1016/j.future.2016.01.009

Uusitalo, I., Karppinen, K., Juhola, A., & Savola, R. (2010). Trust and cloud services - An interview study. In *Proceedings - 2nd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2010* (pp. 712–720). doi:10.1109/CloudCom.2010.41

Xenos, M., Stavrinoudis, D., & Christodoulakis, D. (1999). CESM: a Perceived Software Quality Assessment Tool. In *Proceedings of the International Conference on Software Process Improvement - Research into Education and training, INSPIRE '99*, Heraklion, Crete, Greece (pp. 155–164).

Yao, L., Sheng, Q., & Maamar, Z. (2012). Achieving High Availability of Web Services Based on a Particle Filtering Approach. *International Journal of Next-Generation Computing*, *3*(2), 127–143. Retrieved from http://perpetualinnovation.net/ojs/index.php/ijngc/article/view/157/51

Gaurav Raj pursed his Bachelor of Technology from UPTU, Lucknow, India in 2006 and Master of Technology from MNNIT, Allahabad, India in year 2010. He is currently pursuing a Ph.D. from Punjab Technical University, Kapurthala, Punjab, India and currently working as Assistant Professor in Department of Computer Sciences, Amity University Uttar Pradesh, India since 2013. He has worked in Lovely Professional University as Assistant Professor from 2010 to 2013. He is a professional member of IEEE since January 2018. He has published more than 40 research papers in reputed international conferences and journals including IEEE, ACM, Springer and it is also available online. His main research work focuses on web service, software engineering, cloud security and privacy, service prediction, IoT and computational intelligence-based education. He has 10 years of teaching experience and 5 years of research experience.

Manish Mahajan pursed Bachelor of Technology and Master of technoology from MMEC, Mullana in Information Technology and Ph.D. in Computer Science from Punjab Technical University, Kapurthala in 2016 . He is currently working as Assoc. Professor and HOD in the International Journal of Computer Sciences and Engineering volume 6 issue 6, Aug 2018. He is a professional member of IEEE. He has published more than 20 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and it is also available online. His main research work focuses on Cloud security and privacy, Big Data analytics and data mining. He has 5 years of teaching experience and 4 years of research experience.

Dheerendra Singh Presently working as Associate Professor in Computer Science & Engineering Department at Chandigarh College of Engineering & Technology (Degree Wing), Sector 26, Chandigarh. He has published and presented 91 research papers in national and international journals (including the IEEE, Springer, Elsevier and they have a good impact factor) /Conferences. He is having 16 years of experience of teaching at reputed Engineering Colleges. He did his B.E. (Computer Science & Engineering), M.Tech (Computer Science & Engineering), and PhD (Computer Science & Engineering). He has guided 8 PhD theses, guided 10 M Tech theses and 19 B Tech projects. He is a member of the reviewer panel of the International Journal of Information Technology & Knowledge Management and a member of the reviewer panel of the International Journal of Research in Engineering & Technology, and a life member of the IETE. His main research work focuses on web engineering, software engineering, Cloud security and privacy, Big Data analytics and data mining. At present he is a member of the research committee of the CSE and the IT of UIET at Panjab University, Sector 14, Chandigarh.