Machine Learning Methods for Trust-based Selection of Web Services

Muhammad Hasnain[1], Imran Ghani[2], Muhammad F. Pasha[1], and Seung R. Jeong[3*]
[1] School of Information Technology, Monash University
Subang Jaya, 47500 Malaysia
[e-mail: muhammad.malik1@monash.edu, muhammad.fermipasha@monash.edu]
[2] Computer and Information Sciences Department, Virginia
Military Institute, Lexington, VA, USA
[e-mail: ghanii@vmi.edu]
[3] Kookmin University, Seoul 136, South Korea
[srjeong@kookmin.ac.kr]
[*]Corresponding author: Seung R. Jeong

Abstract

Web services instances can be classified into two categories, namely trusted and untrusted from users. A web service with high throughput (TP) and low response time (RT) instance values is a trusted web service. Web services are not trustworthy due to the mismatch in the guaranteed instance values and the actual values achieved by users. To perform web services selection from users' attained TP and RT values, we need to verify the correct prediction of trusted and untrusted instances from invoked web services. This accurate prediction of web services instances is used to perform the selection of web services. We propose to construct fuzzy rules to label web services instances correctly. This paper presents web services selection using a well-known machine learning algorithm, namely REPTree, for the correct prediction of trusted and untrusted instances. Performance comparison of REPTree with five machine learning models is conducted on web services datasets. We have performed experiments on web services datasets using a ten k-fold cross-validation method. To evaluate the performance of the REPTree classifier, we used accuracy metrics (Sensitivity and Specificity). Experimental results showed that web service (WS1) gained top selection score with the (47.0588%) trusted instances, and web service (WS2) was selected the least with (25.00%) trusted instances. Evaluation results of the proposed web services selection approach were found as (asymptotic sig. = 0.019), demonstrating the relationship between final selection and recommended trust score of web services.

Keywords: Users' trust, machine learning, REPTree, fuzzy rules, trust prediction.

1. Introduction

Web services are loosely coupled, independent, and distributed services that operate on the web infrastructure. Web services are language and platform-independent, making it easier for users to access them in a heterogeneous environment. With the rapid development in web technology, many researchers focused on the functional and interfacing aspects of web services [1]. The first aspect of web services has given birth to industrial standards. For example, 'extensible markup language' (XML) related technologies are used to standardize messages and documents of web services. Web services description language (WSDL) is another standard that is widely used as guidelines for web services. Moreover, 'single object access protocol' (SOAP) is an accepted standard for messages in web services [2].

Web services technology is becoming prevalent and popular due to numerous advantages, including, but not limited to, interoperability, composability, and reusability [2-3]. This has led the world's leading companies, such as Facebook and Google, to provide services and applications in the form of web services [3]. However, the selection of web services is a real challenge for web service users. Among non-functional attributes, RT, TP, and reliability have been widely used in primary studies [4]. Before the recent research, Mao et al. [5] used reliability, RT, and reputation as 'quality of service' (QoS) metrics for selecting web services. Even then, researchers were concerned about the invocation timing of web services. They realized that it was less likely for users to invoke all web services at the same time.

Web services selection is the cornerstone for increasing users' trust in web services; however, sometimes, selection cannot meet the users' expectations regarding the QoS metrics values [6]. Web services selection is based on distinguishing the non-functional attributes [7]. A web service with the same functional requirements is promoted as the best candidate for web service selection, making it difficult for users to select the best web service when more than one web services are equally meeting the functional requirements.

A considerable body of web services selection has examined the link between QoS metrics and consumers' choice to select web services [6-8]. However, service selection based on the QoS metrics has conflict regarding web services selection criteria and performance evaluation methods. In research [9], authors have ranked web services from QoS metrics values using four service selection techniques. In addition, researchers in [10] used QoS metrics and reputation values to select newcomer services. In contrast, research work [11] recommend automated services selection techniques for a better recommendation of web services. Authors in [12] explore web services, geographic location information integrated with the functionality of web services. Latter mentioned studies prefer using the information other than QoS metrics in selecting the web services. Researchers in studies [13-14] explore the trust-based selection of cloud services providers by using their proposed algorithms. They did not use QoS metrics nor geographic location features of web services. Instead, they exploit the objective and subjective data regarding the selection of web services.

Web services are platform-independent and are widely used for better convergence of disparate business functionalities [15]. Similar services compete with each other and can create difficulty for users to select the best performing web services. While the quality of web services is guaranteed in the 'web service level agreement' (WSLA), it is sometimes not reliable for the users [16] because the quality metrics' values guaranteed by service providers are lower than the values obtained at the users' end. Subsequently, this decreases users' trust in web services, which can be illustrated by the following process of web services subscription load.

A web service X is showing the regular operation of TP and RT, according to WSLA. It is due to constant or decreasing load on web service X. Since subscriptions on web service X have expired, TP and RT values are not decreasing. In contrast, another web service Y has a combined load and prescriptions increase. Therefore, web service Y is not meeting the TP and RT goals. User trust in two web services, X and Y, is not the same in the described situations. To meet the increasing load of existing and new prescriptions, web service providers add new servers to fulfill the supply of services. Newly added services, together with the existing web services, present a real challenge for researchers to select the best web services that meet users' expectations.

Moreover, integration with partners' web services and modifications in services are also undertaken to meet the TP and RT values agreed with the customers in the WSLA document. Due to these actions, web service providers ensure that service Y, with its existing functionalities, meets the users' requirements. To ensure that the addition of services is not impacting service Y, and other modifications, performance testing is performed. The purpose of performance testing is to identify the issues in web service Y after adding additional resources. Automated test case suite is developed to measure the peak, normal and exceptional loads on web service Y. Subsequently, a pool of web services can be ranked in a list, and the priority sequence of each web service for the regression testing in a pool is determined that helps users to select web services from a pool of existing and newly added web services.

To cope with this mismatching of quality metrics values attained at two levels (one from a service provider and another from a service user), we need to investigate trust on web services from users' feedback values. Delay in response to a user request and low TP values of web services have never been addressed by training classifiers on web services metrics values [1, 3]. Consequently, we aim to use classification models to examine instances of web services datasets for the selection of web services, which qualify the high trust score from users.

Our contributions are summarized as follows:

1. We propose to perform the selection of web services, using calculated trust score from the classification of web services instances.
2. We analyze web services quality metric values attained at the users' end.
3. We construct and implement fuzzy rules to compute the trust of web services users from each instance value.
4. We perform a comparative analysis of REPTree with the other five classifiers via sensitivity and specificity measures.

In this paper, Section 2 presents a literature review of the existing studies. Section 3 offers the proposed approach to select web services from users' feedback on the invoked instances of web services. Section 4 presents the experimental setup and detailed implementation of the proposed approach on web services datasets, while section 5 and 6 present results and discussion. Section 7 presents threats to validity, and section 8 concludes this study with future works.

2. Literature Review

In recent years, machine learning algorithms have been widely employed in different research domains. There is a growing literature on the use of machine learning algorithms due to their strength in classification. In this section, we present how other researchers have used classification approaches in web services selection.

Yahyaoui et al. [17] have proposed an approach specific to web services and their quality attributes. The quality attributes are extracted from using the set theory. Fuzzy classification rules are extracted that help in matchmaking of web services. A matchmaking aspect needs further improvement by using hybrid techniques. Before this study, Mohanty et al. [1] used well-known classification models to predict web services classification. The TreeNet and J48 classification models were found to be outstanding compared to other models in terms of the accuracy of results. This study found that reliability, RT, TP, documentation, and success ability were the essential features of web services.

To predict the most critical quality features is not precisely the solution for selecting web services. That is why Liu et al. [18] used semantic features of web services and used the accuracy of results to select web services. The accuracy of results based on the classification of web services was different from the semantic information-based classification. Naive Bayes algorithm could provide better results of web services using semantic information. In the latter given study, the authors used interfacing and usability features to classify web services. Chen et al. [19] stated that their proposed approach outperformed the other methods of measuring the similarities in web services discovery. The performance of the proposed plan was calculated by using F-measure, recall, and precision accuracy metrics.

To address the class imbalance data, Maratea et al. [20] proposed adjusted F-Measure that weighed recall more than the precision to be able to strengthen false negative (FN) values. As a result, a new confusion matrix was built by the conversion of positive values into negative values.

Chen et al. [21] used recall, precision, and F-Measure in assessing the predictive capabilities of model-based 'Best-first tree' (BFtree), 'Naive-Bayes tree' (NBtree), and 'Random-forest tree' (RFtree) classifiers. They found that higher values of quality metrics resulted in a better model. Among the three classifiers, RFtree performed better than the other two classifiers in the validation of datasets.

2.1 Trust-based Web Services Selection

To address the challenges of web services selection, researchers have used QoS based approaches. Mallayya et al. [22] proposed a strategy of using the preferences given by users over QoS parameters. The proposed algorithm involves the feedback given by users that helps in determining the reputation of web services. They implemented the proposed approach to a travel application and found that the proposed method took low execution time. Before the proposal of the latter proposed approach, Elfirdoussi [23] also involved the web services and pertinent QoS aspects to select web services. A user's request was matched to the best web service that reduced time in the composition and selection of web services. The common point between these two primary studies is the utilization of user preference, and QoS aspects of web services and user feedback remained an essential component of the proposed approaches.

In a recent study, Lu and Yuan [24] emphasized trustworthiness selection of cloud web services having the same functionalities. They used TOPSIS as a ranking algorithm to assess web service users' trust by combining objective and subjective aspects. The objective aspects of cloud services were undertaken for monitored values of QoS features, i.e., reliability. Entropy weights were assigned to QoS features, which helped reduce the impacts of false and artificial sources of information. Trust's subjectivity was reflected through the trust performance. Weights of objective and subjective aspects were integrated and evaluated by the TOPSIS algorithm. Therefore, it is evident that TOPSIS is an efficient approach for examining multiple objects with the same attributes.

In order to normalize data of QoS features and entropy weights, vector normalization was selected as a better choice than the linear normalization. Before this study was undertaken, Saoud et al. [25] assessed the rating given by end-users to examine the trust approach. The proposed strategies were known as deterministic and probabilistic. The main aim behind the strategy's proposal was to allow the users to select web services using the trust levels. The probabilistic approach was found to be more robust than the deterministic one.

An increased number of web services makes it harder for users to select web services that efficiently meet their requirements according to services level agreement (SLA). To address the challenges of web services ranking, Wong et al. [26] used the WS-Dream dataset of web services with TP and RT quality metrics. They determined the web services violation by matching TP and RT values beyond the threshold to users' observed values. They proposed sixteen fuzzy rules coupled with the machine learning model to decide whether the web service worked within the agreed document or crossed the threshold values. The first limitation of the lateral mentioned approach is that web services datasets have not been explicitly mentioned. The second limitation of the lateral mentioned study is that it does not explicitly reveal the web services selection using TP and response metrics.

Data mining is an emerging research field with many applications to deal with knowledge discovery. Data mining helps in visualizing, understanding, and analyzing extensive data collected from various sources. Data mining study includes the summarization, association, classification, clustering, and prediction [27]. In our proposed approach, we use data mining for the selection of web services. It is necessary to choose the most appropriate classifiers for web services' numerical data. Therefore, we provide an overview of classification techniques as follows:

2.2 Advantages and Disadvantages of Classification Techniques

2.2.1 Naive Bayes Approach

Advantages: This classification technique has been used for the classification of web applications based on defects prediction. It provides precision and accuracy when it is applied to a large size dataset.
Disadvantages: This classification technique provides zero values based on probability when an occurrence has no class label and attribute values. It also requires a solid assumption to shape the data [28].

2.2.2 J48 Approach

Advantages: This classification technique has been used as a predicting model to find valuable information from massive data. This technique also accounts for missing values and ranges.
Disadvantages: It works only on the linearly available separable data. It involves N-P complete problems. It ignores the correlation between attributes. It has difficulty in dealing with the missing information. It favors a method that has more values [29].

2.2.3 Bagging Approach

Advantages: Bagging algorithm is usually applied to increase the prediction accuracy of other classifiers. The central concept behind bagging is to aggregate the several complex classifiers and then average the output of different models. The bagged trees, which are the main parameters, are not pruned, and all features are considered when the tree looks for the best split from each node [30].

Disadvantages: Sometimes, this classifier mildly degrades the performance of the best classifier like K-Nearest Neighbors.

2.2.4 Multi-logistic Regression Approach

Advantages: This classification technique has been used to calculate probabilities of various classes based on a linear combination of observed features
Disadvantages: It does not solve the non-linear problem. It requires the identification of independent and dependent variables [31].

2.2.5 OneR Approach

Advantages: A simple classification technique which generates a set of rules, and a practice with the smallest error ratio is chosen. For numerical data classification, the OneR technique divides the values into many ranges.
Disadvantages: OneR classifier randomly selects one of the rules, and it makes use of inefficient information when several rules have the same information [32].

Moreover, a few tree-based classifiers have been used on numerical data. One of these classifiers is 'Reduced Error Pruning Tree' REPTree classifier, which is used to build a decision or a regression tree. According to Mesaric and Sebalj [33], REPTree works as a fast decision tree learner using information gain as the splitting criterion, and prunes it using the reduced error pruning. Gonzalez-Robledo1 et al. [34] used REPTree and J48 (advanced version of C4.5) classifiers in a primary study. The former classifier is based on the information gain and uses reduced error-pruning for back-fitting. Both classifiers are tree-based. Hussain et al. [35] stated that J48, along with other classifiers, was effectively applied to numerical and categorical attributes. Subsequently, we propose to use REPTree as a base classifier and validate its performance using the additional five classifiers.

3. Proposed Approach

This section presents the explanation of the proposed approach and its consequences on the web services classification: The proposed web services selection approach exploits a binary classification of TP and RT metrics. Users give both the TP and RT values as feedback about the invoked web services. The feedback values are preprocessed and normalized to make useful classification of users' invoked web services instances. In **Fig. 1**, we present the algorithm of the proposed approach.
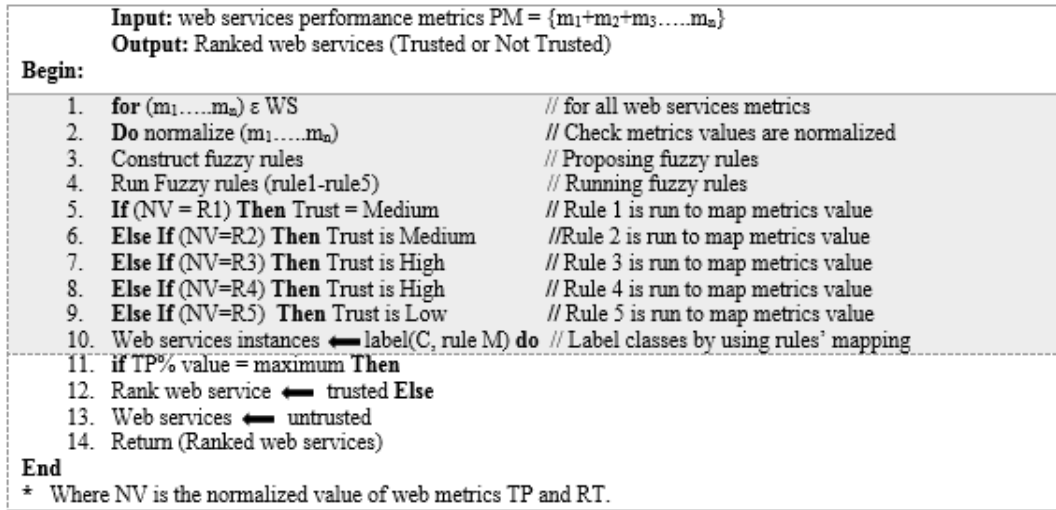
Input: web services performance metrics PM = $\{m_1+m_2+m_3\ldots.m_n\}$
Output: Ranked web services (Trusted or Not Trusted)
Begin:

1. **for** $(m_1\ldots.m_n)$ ε WS   // for all web services metrics
2. **Do** normalize $(m_1\ldots.m_n)$ // Check metrics values are normalized
3. Construct fuzzy rules   // Proposing fuzzy rules
4. Run Fuzzy rules (rule1-rule5)   // Running fuzzy rules
5. **If** (NV = R1) **Then** Trust = Medium   // Rule 1 is run to map metrics value
6. **Else If** (NV=R2) **Then** Trust is Medium   //Rule 2 is run to map metrics value
7. **Else If** (NV=R3) **Then** Trust is High   // Rule 3 is run to map metrics value
8. **Else If** (NV=R4) **Then** Trust is High   // Rule 4 is run to map metrics value
9. **Else If** (NV=R5) **Then** Trust is Low   // Rule 5 is run to map metrics value
10. Web services instances ← label(C, rule M) **do** // Label classes by using rules' mapping
11. if TP% value = maximum **Then**
12. Rank web service ← trusted **Else**
13. Web services ← untrusted
14. Return (Ranked web services)

End
* Where NV is the normalized value of web metrics TP and RT.

Fig. 1. An overview of the proposed algorithm

Fig. 1 is the overall demonstration of web services instances classification and selection by using fuzzy rules. A user trust-based approach is explained with several loops and control statements. For instance, values of web services metrics ($m_1...m_n$) are normalized for further processing. As fuzzy rule function is executed on normalized web services metrics values, we find a very precise class of metric for web service instances. Furthermore, we label classes as per our findings from the latter discussed function. We provide fuzzy rules construction in section 3.2. As fuzzy rules' mapping through the function completes, we classify the web services instances by machine learning methods. Thus, fuzzy rules provide the ground truth for accurate classification of trusted and untrusted instances of web services. Our proposed TP% method (explained in section 3.3) is used to calculate each web service's trust score. The trust score of web services determines the trust-based selection of web services.
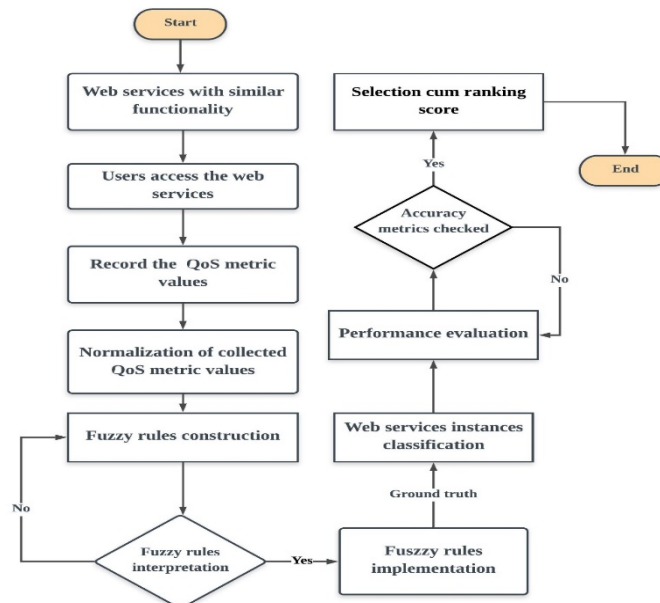

Fig. 2. Flowchart of trust-based selection of web services approach

Based on the above discussion and algorithm of the proposed approach, we present an overview of the trust-based forecast of web services selection as shown in **Fig. 2**, which demonstrates the flow process of proposed trust-based selection of web services. Given schema is aligned with the algorithm of the proposed approach. As shown in **Fig. 2**, trust-based prediction uses web services web services with the similar functionality as a starting point, and after several processes, it ends in the final predicted selection of web services.

3.1 Data Preprocessing

Data preprocessing has become an essential technique to extract new information from scenarios of a large dataset. This technique is aimed to reduce the complexity inherent in real-world datasets. Thus, a dataset can be easily processed using classification methods. A precise and faster learning process is set for the logical structure of raw information [36]. Data preprocessing techniques involve preprocess of the imbalanced dataset by altering the data's distribution as per the goals of users. Then any of the standard algorithms is used to preprocess the dataset [37]. We use the data normalization technique for data preprocessing, as given in the subsequent section.

3.2 Data Normalization

Data normalization, in general, is applied to numerical data when the range of raw data has a variance. This variance in the data can decrease the performance of machine learning algorithms. Several normalization methods have been used in the research, i.e., min-max normalization and standard score [38]. Min-max normalization scales the data features between 1 and 0, and this method is suitable for our chosen dataset because positive values are required. As shown in the following equation (1), the normalized value Zi is computed for an attribute x from the best values considered as the highest one.

$$\text{Zi} = \frac{xi - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where xi is the value of an attribute, and min and max denote the minimum and maximum values on all value of the respective attribute.

3.3 Fuzzy Rules Construction

Construction of fuzzy rules is employed to learn the structure of web services instances. The rationale behind proposing fuzzy rules is to improve the binary classification for the correct selection of web services. We cannot assign a class to web service instances without any evidence of handling the sparse values of web services instances. Fuzzy logic increases the power of decision making to label the information consistently. Before our proposed work, Oliff and Liu [39] developed a decision support model after extracting the fuzzy rules. Since we propose to use machine learning models on our emphasized and correctly labeled values of web services instances, we require to train on the correctly labeled information of web services [40]. We construct fuzzy rules with the objectives of labeling normalized web services instances data and keeping web services transactions into two classes. We propose to use users' feedback in terms of TP and RT values that users report after they invoke web services.

Furthermore, we interpret the users' given feedback by constructing rules as given in **Table 1**. We interpret statements of the rules to decide the mode of a user. As stated in the following table about the description of fuzzy rules, we propose to keep fuzzy rules into two

classes to perform binary classification. We have grouped various observations into two classes by splitting the data based on the similarities [41]. In **Table 1**, we present the examples of our proposed IF-THEN rules.

Table 1. IF-THEN rules

Rule No.	Rule Structure
1.	If throughput value is acceptable OR response time value is non-acceptable, Then a user's trust will be medium.
2.	If the response time value is acceptable, Then a user's trust will be medium.
3.	If the response time value is amazing OR throughput value is acceptable, Then a user's trust will be high.
4.	If the response time value is amazing OR throughput value is amazing, Then a user's trust will be high.
5.	If response time is non-acceptable OR throughput value is non-acceptable, Then a user's trust will be low.

We have constructed fuzzy rules from our observations of the dataset used in this study (see **Table 1**). For the construction of fuzzy rules, we have combined TP and RT values. The purpose of using the values of TP and RT together is to achieve better construction of rules. Subsequently, we state how these rules can be interpreted and used to label instances or transactions.

Table 2. Linguistic variables and their ranges for TP instances

Linguistic Value	Numerical Ranges
Non-acceptable	[0:00; 0:30]
Acceptable	[0:30; 0:70]
Amazing	[0:70; 1:00]

Table 3. Linguistic variables and their ranges for RT instances

Linguistic Value	Numerical Ranges
Amazing	[0:00; 0:30]
Acceptable	[0:30; 0:70]
Non-acceptable	[0:70; 1:00]

Table 4. Linguistic variables and their ranges for output membership function

Linguistic Value	Numerical Ranges
Low	[0:00; 0:30]
Medium	[0:30; 0:70]
High	[0:70; 1:00]

As we know that TP and RT are inversely proportional to each other, we keep them together to avoid a long listing of rules. The first rule aims to identify the users with medium-level trust in web services regarding RT and TP instances and their values. An instance with the acceptable TP metric value and non-acceptable RT values makes a user medium-level trusted in respective web services. The second rule aims to show that a user's trusted on a web service

is medium-level since RT value is acceptable. In comparison to TP, and RT values stated in the SLA, a user with the medium trust level cannot be kept in the C1 class of web services instances. Therefore, we follow rule 1 and rule 2 in class C0. The third rule mentions that a user has given feedback by referring to RT's amazing value and an acceptable value of TP. Therefore, trust-level of a user for such kind of web service is high. The fourth rule mentions that a user's trust level is high as RT and TP metrics have amazing values. In case a user gives feedback for a web service by saying non-acceptable RT and TP metrics values, then a user's trust level is low on web services.

Tables 2-4 show us the linguistic values, numerical ranges for each RT and TP instances, and the output membership function. The implementation of proposed fuzzy rules has been conducted in Python Notebook 3.0 with the help of scikit-fuzzy libraries [42], as given in the following Table 5.

Table 5. Proposed fuzzy rules' implementation

Rule No.	Rule's Implementation
1.	rule1 = ctrl.Rule(tput['Acceptable'] | rtime['Non-acceptable'], trust['medium'])
2.	rule2 = ctrl.Rule(rtime['Acceptable'], trust['medium'])
3.	rule3 = ctrl.Rule(rtime['Amazing'] | tput['Acceptable'], trust['high'])
4.	rule4 = ctrl.Rule(rtime[' Amazing '] | tput['Amazing'], trust['high'])
5.	rule5 = ctrl.Rule(rtime[' Non-acceptable '] | tput[' Non-acceptable'], trust['low'])

We propose the application of Takagi-Sugeno fuzzy inference system (FIS) modeling due to its decisive role in optimization techniques [43]. The inference method derived from this modeling, and typical fuzzy rules, based on the assumption that every rule has two inputs with the logical operator 'OR' operation, are written in Table 1. For instance, our input variables RT and TP use three fuzzy sets: non-acceptable, acceptable, and amazing. Membership functions of three fuzzy sets can be triangular. A crisp value is the conclusion of Takagi-Sugeno FIS.

3.4 Trust Prediction (Tp')

We propose a web services selector using confusion matrix measures. For this, we aim to select web services on correctly predicted 'true positive' TP instances. To compute the trusted instances of web services, we propose a TP% age method as given in equation (2). We aim to use the true positive measure of confusion matrix results. Based on the TP %age, we calculate the trust score of web services users on the chosen web services.

$$TP \, \%age \; = \frac{TP \, value}{Total \, Instances} x \, 100 \qquad (2)$$

where TP value is expressing the number of correctly predicted TP instances, and Total instances are the sum of web services instances. We propose TN %age from a true negative measure of confusion matrix results, as shown in equation (3).

$$TN \, \%age \; = \frac{TN \, value}{Total \, Instances} x \, 100 \qquad (3)$$

where TN value expresses the true negative correctly predicted web services instances, although we use TP percentage method to calculate the trust score of web services, we may apply TN %age method wherever we find the web services with duplicate TP %age. If two or more than two web services obtain the same trust score, TN %age is alternatively applied to measure the trust score of web services. In such a case, a web service with a least TN %age is deemed most trusted by users.

4. Experimental Setup

This section presents experiments on the dataset of web services to evaluate the proposed web services selection approach. Moreover, the description of the dataset, cross-validation method and accuracy metrics are given as follows:

4.1 Dataset

We propose using the quality of services (QoS) dataset known as the WS-DREAM dataset of web services. This dataset contains 339 user invocation records for 5258 web services. This dataset has been used in many primary studies [13, 44-45]. The original statistics of this dataset are given in **Table 6**.

Table 6. Dataset statistics

Item	Value
Users	339
QoS Records (5 web services)	1695
Throughput Range (Kbps)	1~1000
Response Time Range	0~20s

4.2 Cross-Validation

We select TP and RT as a QoS metric and use a 10-fold cross-validation technique. The training set is divided into 10-fold of equal size, where one set is kept for testing and the remaining nine sets for training.

4.3 Accuracy Metrics Determination

We determine precision, F-measure, sensitivity, and specificity accuracy metrics values from 10-fold cross-validation. We provide formal definitions of accuracy metrics used in this study, as given follows:

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

$$Accuracy\ Acc. = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

F-measure metric, along with precision and recall, has been recently used by Chen et al. [9] as an evaluation and discovery method for web services. Both the precision and recall have been integrated, as shown in equation (5).

Sensitivity or recall is computed as a ratio between truly classified 'true positive' TP instances and the sum of TP and 'false negative' FN instances.

$$Sensitivity = \frac{TP}{TP + FN} \qquad (7)$$

Specificity is computed as a ratio between truly classified 'true negative' TN instances and the sum of the TP and 'false positive' FP instances.

$$Specificity = \frac{TN}{TN + FP} \qquad (8)$$

Before classifying the instances of web services, we perform the labeling of our two proposed classes C1 and C0. Hence, we use binary classification and organize the web services instances either within the users' trust level or beyond their trust level.

4.4 Results

This section presents the classification of web services instances and the evaluation results of our proposed approach. First, we show the web services classification results before applying fuzzy rules to label the web services instances from web services datasets. Rather than using the random labeling for web services instances, we propose to use labeling of web services instances in equal numbers (C1 = 34), and (C0 = 34); where C1 is the class of trusted web services instances, and C0 is the class of untrusted web services instances. It is done to avoid the labeling bias in the training of our chosen classifiers. Non-trivial classification accuracy is ensured by balancing the true positive and true negative instances [38]. Furthermore, we computed sensitivity and specificity accuracy metrics results and presented them with the same metric results in the lateral part of this section.

Logistic classifier showed a high accuracy (Acc = 0.5147) following by OneR with (Acc = 0.4705) from WS1 dataset; J48 classifier showed accuracy (Acc = 0.6911) followed by OneR with (Acc = 0.6764) from WS2 dataset; J48, and BayesNet showed accuracy (Acc = 0.4411) followed by OneR with accuracy (Acc = 0.4264) from WS3 dataset; J48, OneR, and BayesNet showed accuracy (Acc = 0.4411) followed by REPTree with accuracy (Acc = 0.4264) from WS4 dataset; and Logistic showed accuracy (Acc = 0.5735) followed by REPTree with accuracy (Acc = 0.5441) from WS5 dataset. It is revealed that Logistic, OneR, J48, BayesNet, and REPTree target the trusted and untrusted instances of web services with high accuracy in comparison with the Bagging classifier.

We exploit the confusion matrix to interpret the results from the classification of web services instances after using fuzzy rules. This interpretation of results allows us to conclude the instances into many ranks for the chosen web services. For example, we have a confusion matrix of WS3 with a 20% density. We find the following items of the confusion matrix, as given in **Table 7**.

Table 7. Confusion matrix of WS3 dataset

N = 68	Predicted No	Predicted Yes	
Actual No	TN=44	FP=01	45
Actual Yes	FN=03	TP=20	23
	47	21	

Table 7 shows the confusion matrix measures results of the WS3 dataset from the REPTree model. Similar to results given in **Table 7**, we obtain the confusion matrix measures results of other web services datasets. A confusion matrix provides us with an idea of how accurate instances are classified. As shown in **Table 8**, a total of 68 instances from two classes, C1, and C0, have been classified. Out of 21 instances from class C1, 20 instances were correctly predicted as instances of the WS3 dataset. Out of 47 instances labeled as C0, 44 instances were correctly predicted.

Table 8. Web services two classes correct classification from REPTree classifier

Dataset	C1 class instance	C0 class instance	TP	TN	FP	FN
WS1	34	34	32	33	2	1
WS2	19	49	17	46	2	3
WS3	21	47	20	44	1	3
WS4	28	40	28	34	0	6
WS5	29	39	22	37	7	2

Results presented in **Table 8** give us an indication of the correct classified instances within the user trust level and beyond the trust level. It was helpful for us to correctly predict the trust of users on the invoked web services. In **Table 9**, we present the %age values of each of the four measures from the respective confusion matrixes of web services datasets. The %age values shown in **Table 9** are calculated from REPTree confusion matrix measures.

Table 9. Confusion matrix measures based on two classes' values

Dataset	Trusted Class (TP %)	Untrusted Class (TN %)	Selection cum Ranking
WS1	47.0588	48.5294	1
WS2	25.0000	67.6400	5
WS3	29.4117	64.7058	4
WS4	41.1764	50.0000	2
WS5	32.3529	54.4117	3

Table 9 shows the percentage distribution of values from four confusion matrix measures used for the selection of web services. FP % and FN %, as shown in **Table 9**, express the false positive and false negative percentage values. Users' trust based on the prediction of instances within the users' trust level (FP %) was demonstrated in terms of selection cum ranking score, which is calculated by using equation (2). Furthermore, we computed untrusted users using equation (3). Due to accumulated and correctly predicted instances within the users' trust level, we established a selection of web services. It interprets that every web service shows a particular correct classification of instances within the users' trust level.

4.5 Performance Comparison of Classifiers

To evaluate the performance of the chosen REPTree classifier, we use other fives classifiers on the same web services dataset. The purpose of using J48, OneR, Bagging, Logistic, and BayesNet classifiers is to observe whether the former classifier performs better than the rest of classifiers. Moreover, we compare classifiers' performance for the correct classification of instances within the user trust level and beyond the trust level in terms of sensitivity and specificity metrics values.

Table 10. Sensitivity and specificity results before and after using fuzzy rules

Dataset	Classifier	Sensitivity II	Specificity II	Sensitivity I	Specificity I
WS1	REPTree	0.9696	0.9428	0.4000	0.4211
	J48	0.8618	0.9677	0.4500	0.4285
	OneR	1.0000	1.0000	0.4750	0.4643
	Bagging	0.9090	0.8857	0.4411	0.4412
	Logistic	0.9142	0.9393	0.5238	0.5106
	BayesNet	0.9090	0.8857	0.4500	0.4285
WS2	REPTree	0.8500	0.9583	0.6086	0.7272
	J48	0.8095	0.9574	0.6382	0.8595
	OneR	0.8261	1.0000	0.6250	0.8000
	Bagging	0.8181	0.9782	0.6000	0.6428
	Logistic	0.8667	0.8868	0.5800	0.7222
	BayesNet	0.8261	1.0000	0.4411	0.4411
WS3	REPTree	0.8695	0.9777	0.4047	0.3461
	J48	0.9090	0.9782	0.4500	0.4286
	OneR	0.8000	0.9767	0.4390	0.4074
	Bagging	0.9524	0.9787	0.3666	0.3947
	Logistic	0.7727	0.9130	0.3636	0.2500
	BayesNet	0.9524	0.9787	0.4500	0.4285
WS4	REPTree	0.8235	1.0000	0.4545	0.3076
	J48	0.8387	0.9459	0.4500	0.4285
	OneR	0.6842	0.9333	0.4500	0.4285
	Bagging	0.8065	0.9189	0.3448	0.3846
	Logistic	0.7333	0.8421	0.3902	0.3333
	BayesNet	0.8235	0.9787	0.4500	0.4285
WS5	REPTree	0.9166	0.8409	0.5283	0.6000
	J48	0.9166	0.8409	0.4564	0.4091
	OneR	0.8519	0.8537	0.5161	0.5135
	Bagging	0.9565	0.8444	0.4705	0.4706
	Logistic	0.6923	0.2619	0.5757	0.5714
	BayesNet	0.9565	0.8444	0.4500	0.4285

Sensitivity is the proportion of true instances within the users' trust of web services datasets. On the other hand, specificity is the proportion of instances beyond the users' trust level. **Table 10** shows classification results for each type of machine learning classifier when trained on web services instance data: trusted instance of web services and cases untrusted of web services. For qualitative characterization, we emphasize the sensitivity results. In this context, *sensitivity* g 0:90 indicates an excellent accuracy while a classifier having f 0:60 is considered

a weak classifier [46]. Sensitivity I and Specificity I represent results before using fuzzy rules, while Sensitivity II, and Specificity II represent results after using fuzzy rules (see **Table 10**)

For WS1 dataset, OneR achieved the highest performance metrics: a sensitivity II of 1.000, and a specificity II statistics of 1.000. For the WS2 dataset, the Logistic classifier outperformed the rest of classifiers in terms of sensitivity II (0.8667), while the OneR classifier achieved the highest performance metric in terms of specificity II (1.000). For the WS3 dataset, Bagging and BayesNet classifiers reached the highest sensitivity II: 0.9524, while J48, Bagging, and BayesNet attained the highest specificity II: 0.9787. For the WS4 dataset, J48 classifier achieved the highest sensitivity II: 0.8387, while the REPTree classifier achieved the highest specificity II: 1.000. For the WS5 dataset, Bagging and BayesNet achieved the highest sensitivity II: 0.9565, while OneR achieved the highest specificity II: 0.8537.

Before using fuzzy rules for labeling of web services instances, we achieved lower sensitivity I and specificity I value from WS1-WS5 web services datasets. Statistics shown in **Table 10** indicate that none of the classifiers attained excellent sensitivity I for WS1-WS5 datasets. However, REPTree along with J48, OneR, and Bagging classifiers achieved sensitivity I greater than poor one from WS2 dataset.

4.6 Performance Evaluation of the Proposed Approach

To evaluate the performance of the proposed approach, we consider the following points.

1. Computed TP %age of each web service
2. Initial selection and
3. The final selection of web services

We apply p-value to see the relationship between the three factors. We discuss nonparametric tests and then select the one that is more appropriate for our case. Wilcoxon signed-rank test is suitable for a small number and repeating measures. Karaboga and Kaya [47] used the Wilcoxon signed-rank test for comparison of two extended versions of Artificial Bee Colony algorithms. P-Value was taken as a pivotal factor in reporting the better algorithm. Before our proposed work, Kaur and Kaur [48] also used the Wilcoxon test to see the differences between the performances of various classifiers. They found that 'instance-based learner' IBk classifier was the most accurate classifier by Friedman test. In recent work, Kitchenham et al. [49] used the Wilcoxon test to convert data into ranks of two datasets of n measures. Wilcoxon (W) test was represented as:

$$W = U + \frac{(N+1)N}{2} \qquad (9)$$

Both W and U were used based on the assumption that there were no duplicate values. Besides the Wilcoxon signed-rank test, Kitchenham et al. [49] reported that nonparametric tests, particularly with 'nonparametric two way repeated measures' ANOVA, were other data analysis approaches which are more appropriate for ordinal data. Since the final trust score (TP %age) is derived from the subjective assessment of web services profiling, we consider that ANOVA analysis is more suitable than the Wilcoxon test to evaluate our proposed approach of web services selection. Before reporting the ANOVA two-way variance results and their interpretation, we present the original Friedman Formula in equation (10).

$$M = \frac{12}{NK(K+1)} \sum (Ri)^2 + 3N(K+1) \qquad (10)$$

where K is the number of columns in the treatment, and n is many rows. Ri expresses the number of ranks.

We performed (ANOVA) in SPSS 23.0. We find that M is > than the critical value, so the null hypothesis is rejected. It indicates that initially selected web services are dissimilar in their new positions. In our case, we have two measurements of our dependent variable: initial selection/position and final selection/position of web services. For example, we collect initial selection and a final selection of web services for our calculated TP %age values. The purpose of using ANOVA is to statistically compare similarity or dissimilarity before and after TP % distribution. We use the same web services, and two samples are taken where the first sample implies the initial selection and TP %age values, and the second sample suggests the relationship between final selection and TP %age values. We proceed with the construction of null and an alternative hypothesis as given follows:

H0: The distributions of TP\% values, Initial Position, and Final Position are the same.

HA: The distributions of TP\% values, Initial Position, and Final Position are not the same.

Table 11. Hypothesis test summary

Hypothesis Test Summary				
	Null Hypothesis	**Test**	**Sig.**	**Decision**
1	The distributions of TP% values, Initial Position, and Final Position, are the same.	Related-Samples Friedman's Two-Way Analysis of Variance by Ranks	.019	Reject the null hypothesis.
Asymptotic significances are displayed. The significance level is .05.				

We present ANOVA results in **Table 11**. From **Table 11**, we can see that p-value (sig.) is less than 0.05, indicating that the alternative hypothesis is accepted. Moreover, the initial selection and final selection of web services are significantly different after the computation of TP %age values.

Table 12. Hypothesis testing results

Test	Related-Samples Friedman's Two-Way Analysis of Variance by Ranks
Sig.	0.0193054387731441
Decision	Reject the null hypothesis.
H_0 Null Hypothesis	The distributions of TP % values, Initial Position, and Final Position, are the same.
H_A Hypothesis	The distribution of TP % values, Initial Position, and Final Positions are not the same.

Table 12 shows the summary of the hypothesis testing results. P-value (sig) allows us to make a precise statement about an event that has not occurred. Moreover, the decision, H_0, and H_A hypothesis statements depict the summary of our hypothesis testing in this paper.

Table 13. Two related samples output summary

Names	Statistics
Total N	5
Total Static	7.895
Degree of Freedom	2
Asymptotic Sig. (2 sided test)	0.019

To see whether the finding is statistically significant, we performed Friedman's Two-way ANOVA test. As shown in **Table 13**, the difference observed across 2 degrees of freedom is significant (Asymptotic Sig. (2 sided = 0.019) as value is less than 0.05, so the null hypothesis is rejected. Subsequently, we can conclude that TP%, initial selection, and final selection are closely related to each other.

5. Discussion

The performance of machine learning classifiers evaluated by the 10-fold cross-validation using different statistical accuracy metrics, including precision, F-Measure, sensitivity, and specificity, is determined. In terms of accuracy metrics, Bagging, and BayesNet classifiers showed the highest predictive performance for WS3 dataset, followed by J48, and REPTree classifiers. Generally, the accuracy results given by five metrics indicate that all classifiers showed robustness and high-performance results to be used for the classification of web services instances.

Based on the trust prediction of web services datasets, six classifiers were evaluated. The analysis of all six classifiers revealed that each classifier reflected the data it contained. For instance, WS1 dataset exhibited a relatively balanced prediction performance of trusted instances and untrusted instances of web services with high sensitivity and specificity rates (**Table 10**). The WS2 dataset enclosed more untrusted web services instances than trusted instances; hence, it showed a higher specificity. WS3 also showed a higher performance prediction for untrusted instances than trusted instances and revealed a higher specificity.

Furthermore, WS4, and WS5 web services datasets contained a higher number of untrusted web instances than the trusted instances of web services. Therefore, the specificity of WS4 dataset remained higher from REPTree classifier in comparison to other classifiers. On the other hand, the sensitivity for WS5 dataset remained higher for WS5 dataset from REPTree classifier.

We noticed the most striking result that emerged from web services datasets was the correlation between the initial accuracy metrics (sensitivity I and sensitivity II), and (specificity I and specificity II). Big improvement in the performance of REPTree and other five classifiers is worth mentioning because all experiments result showed sensitivity I, and specificity I changed to highly significant sensitivity II, and specificity II. Mainly, fuzzy rules were used to correctly label the instances that resulted in improving the classifiers' performance. We further explain that improvement in REPTree classifier is dependent mainly on the accuracies gained before and after using the fuzzy rules.

6. Threats to Validity

There are some limitations to this work. First, binary classification is restricted to classifying the instances within two classes. Alternatively, we can employ multiple classifications of web services to achieve better accuracy in the selection. Although we trained multiple classifiers on our chosen web services datasets and used REPTree and J48, OneR, Bagging, Logistic, and BayesNet classifiers, we can improve the results accuracy by selecting the most advanced classification techniques. Alternatively, we can propose our classifier for the accurate classification of web services instances. We applied ANOVA two-way variance as a statistical method to evaluate the proposed web services selection approach. To assess and compare our proposed plan, we have not considered other evaluation metrics for the selection of web services.

7. Conclusion and Future Implications

This paper presents web services selection using the feedback for TP and RT instances from web services users. A classifier, namely REPTree, has been used for the correct prediction of web services instances (trusted or untrusted). Correctly predicted Users' trusted instances had been further used to select the web services. The WS1 dataset has been selected as the most trusted web service and WS2 as the least trusted web service. Trust-based selection of web services through classification is one of the continuous developments for better regression testing of web services. J48, OneR, Bagging, Logistic, and BayesNet classifier were used to compare the performance of REPTree classifier using precision, F-Measure, accuracy, sensitivity, and specificity metrics. We used a nonparametric (Two-way ANOVA analysis) test to evaluate the web services selection approach and found that the distribution of TP %age value, initial selection, and final selection were not the same. We plan to perform regression testing of web services by using our ranking based selected web services in the future work.

Acknowledgement

References

[1]   R. Mohanty, V. Ravi, and M. R. Patra, "Web-services classification using intelligent techniques," *Expert Syst. Appl.*, vol. 37, no. 7, pp. 5484-5490, Jul. 2010. Article (CrossRef Link).

[2]   L. Bahri, B. Carminati, and E. Ferrari, "Privacy in web service transactions: A tale of more than a decade of work," *IEEE Trans. Serv. Comput.*, vol. 11, no. 2, pp. 448-465, Jun. 2017. Article (CrossRef Link).

[3]   Y. Syu, J. Y. Kuo, and Y. Y. Fanjiang, "Time series forecasting for dynamic quality of web services: an empirical study," *J. Syst. Softw.*, vol. 134, pp. 279-303, Dec. 2017. Article (CrossRef Link).

[4]   P. Zhang, H. Jin, Z. He, H. Leung, W. Song, and Y. Jiang, "IgS-wBSRM: A time-aware Web Service QoS monitoring approach in dynamic environments," *Inf. Softw. Technol.*, vol. 96, pp. 14-26, Apr. 2018. Article (CrossRef Link).

[5]   C. Mao, J. Chen, D. Towey, J. Chen, and X. Xie, "Search-based QoS ranking prediction for web services in cloud environments," *Futur. Gener. Comp. Syst.*, vol. 50, pp. 111-126, Sep. 2015. Article (CrossRef Link).

[6]   W. Serrai, A. Abdelli, L. Mokdad, and Y. Hammal, "Towards an efficient and a more accurate web service selection using MCDM methods," *J. Comput. Sci.*, vol. 22, pp. 253-267, Sep. 2017. Article (CrossRef Link).

[7]   W. Wang, Z. Huang, and L. Wang, "ISAT: An intelligent Web service selection approach for improving reliability via two-phase decisions," *Inf. Sci.,* vol. 433-434, pp. 255-273, Apr. 2018. Article (CrossRef Link).

[8]   M. Yaghoubi, and A. Maroosi, "Simulation and modeling of an improved multi-verse optimization algorithm for QoS-aware web service composition with service level agreements in the cloud environments," *Simul. Model. Pract.Theory*, vol. 103, p. 102090, Sep. 2020. Article (CrossRef Link).

[9]   S. Maheswari, and G. Karpagam, "Performance evaluation of semantic based servicevselection methods," *Comput. Electr. Eng.*, vol. 71, pp. 966-977, Oct. 2018. Article (CrossRef Link).

[10] O. Tibermacine, C. Tibermacine, and F. Cherif, "Estimating the reputation of newcomer web services using a regression-based method," *J. Syst. Softw.*, vol. 145, pp. 112-124, Nov. 2018. Article (CrossRef Link).

[11] N. Almarimi, A. Ouni, S. Bouktif, M. W. Mkaouer, R. G. Kula, and M. A. Saied, "Web service API recommendation for automated mashup creation using multi-objective evolutionary search," *Appl. Soft, Comput.*, vol. 85, p. 105830, Dec. 2019. Article (CrossRef Link).

[12] K. A. Botangen, J. Yu, Q. Z. Sheng, Y. Han, and S. Yongchareon, "Geographic-aware collaborative filtering for web service recommendation," *Expert Syst. Appl.*, vol. 151, p. 113347, Aug. 2020. Article (CrossRef Link).

[13] N. Somu, G. R. MR, K. Kirthivasan, and S. S. VS, "A trust centric optimal service ranking approach for cloud service selection," *Futur. Gener. Comp. Syst.,* vol. 86, pp. 234-252, Sep. 2018. Article (CrossRef Link).

[14] U. Noor, Z. Anwar, J. Altmann, and Z. Rashid, "Customer-oriented ranking of cyber threat intelligence service providers," *Electron. Commer. Res. Appl.*, vol. 41, p. 100976, Jun. 2020. Article (CrossRef Link).

[15] P. Hung, *Web Service Composition and New Frameworks in Designing Semantics: Innovations: Innovations*, Hershey, Pennsylvania, USA: IGI Global, 2012.

[16] A. Dan et al., "Web services on demand: WSLA-driven automated management," *IBM Syst. J.,* vol. 43, no. 1, pp. 136-158, Jan. 2004. Article (CrossRef Link).

[17] H. Yahyaoui, M. Almulla, and H. S. Own, "A novel non-functional matchmaking approach between fuzzy user queries and real world web services based on rough sets," *Futur. Gener. Comp. Syst.*, vol. 35, pp. 27-38, Jun. 2014. Article (CrossRef Link).

[18] J. Liu, Z. Tian, P. Liu, J. Jiang, and Z. Li, "An approach of semantic web service classification based on Naive Bayes," in *Proc. of 2016 IEEE Int. Confer. (SCC)*, San Francisco, CA, USA, pp. 356-362, 2016.

[19] F. Chen, C. Lu, H. Wu, and M. Li, "A semantic similarity measure integrating multiple conceptual relationships for web service discovery," *Expert Syst. Appl.*, vol. 67, pp. 19-31, Jan. 2017. Article (CrossRef Link).

[20] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," *Inf. Sci.,* vol. 257, pp. 331-341, Feb. 2014. Article (CrossRef Link).

[21] W. Chen, S. Zhang, R. Li, and H. Shahabi, "Performance evaluation of the GIS-based data mining techniques of best-first decision tree, random forest, and naïve Bayes tree for landslide susceptibility modeling," *Sci. Total Environ.*, vol. 644, pp. 1006-1018, Dec. 2018. Article (CrossRef Link).

[22] D. Mallayya, B. Ramachandran, and S. Viswanathan, "An automatic web service composition framework using QoS-based web service ranking algorithm," *Sci. World J.*, vol. 2015, Oct. 2015. Article (CrossRef Link).

[23] S. Elfirdoussi, Z. Jarir, and M. Quafafou, "Ranking web services using web service popularity score," *Int. J. Inf. Technol. Web Eng.*, vol. 9, no. 2, pp. 78-89, Apr. 2014. Article (CrossRef Link).

[24] L. Lu and Y. Yuan, "A novel TOPSIS evaluation scheme for cloud service trustworthiness combining objective and subjective aspects," *J. Syst. Softw.,* vol. 143, pp. 71-86, Sep. 2018. Article (CrossRef Link).

[25] Z. Saoud, N. Faci, Z. Maamar, and D. Benslimane, "A fuzzy-based credibility model to assess Web services trust under uncertainty," *J. Syst. Softw.*, vol. 122, pp. 496-506, Dec. 2016. Article (CrossRef Link).

[26] T. S. Wong, G. Y. Chan, and F. F. Chua, "A machine learning model for detection and prediction of cloud quality of service violation," in *Proc. of Int. Conf. (ICCSA)*, *Saint Petersburg*, SPB, RU., pp. 498-513, 2018.

[27] M. Phankokkruad, "Classification of file duplication by hierarchical clustering based on similarity relations," in *Proc. of 13th Int. Conf. (ICNC-FSKD)*, Guilin, China, pp. 1598-1603, 2017.

[28] Ö. F. Arar, and K. Ayan, "A feature dependent Naive Bayes approach and its application to the software defect prediction problem," *Appl. Soft. Comput.*, vol. 59, pp. 197-209, Oct. 2017. Article (CrossRef Link).

[29] P. Chandrasekar, K. Qian, H. Shahriar, and P. Bhattacharya, "Improving the prediction accuracy of decision tree mining with data preprocessing," in *Proc. of Conf. (COMPSAC)*, Turin, Italy, pp. 481-484, 2017.

[30] Y. Alsouda, S. Pllana, and A. Kurti, "Iot-based urban noise identification using machine learning: performance of SVM, KNN, bagging, and random forest," in *Proc. of (COINS)*, Crtete, Greece, pp. 62-67, 2019.

[31] O. Miguel-Hurtado, R. Guest, S. V. Stevenage, G. J. Neil, and S. Black, "Comparing machine learning classifiers and linear/logistic regression to explore the relationship between hand dimensions and demographic characteristics," *PLoS One*, vol. 11, no. 11, pp. 1-25, Nov. 2016. Article (CrossRef Link).

[32] O. N. Al Sayaydeha and M. F. Mohammad, "Diagnosis of the Parkinson disease using enhanced fuzzy min-max neural network and OneR attribute evaluation method," in *Proc. of Int. Conf. (ICOASE),* Zakho-Duhok, Iraq, pp. 64-69, 2019.

[33] J. Mesarić, and D. Šebalj, "Decision trees for predicting the academic success of students," *Croat. Oper. Res. Rev.*, vol. 7, no. 2, pp. 367-388, Dec. 2016. Article (CrossRef Link).

[34] J. González-Robledo, F. Martín-González, M. Sánchez-Barba, F. Sánchez-Hernández, and M. N. Moreno-García, "Multiclassifier systems for predicting neurological outcome of patients with severe trauma and polytrauma in intensive care units," *J. Med. Syst.,* vol. 41, no. 9, pp. 1-8, Jul. 2017. Article (CrossRef Link).

[35] M. Hussain, W. Zhu, W. Zhang, and S. M. R. Abidi, "Student engagement predictions in an e-learning system and their impact on student course assessment scores," *Comput. Intell. Neurosci.*, vol. 2018, pp. 1-22, Oct. 2018. Article (CrossRef Link).

[36] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39-57, May. 2017. Article (CrossRef Link).

[37] A. Roy, R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "A study on combining dynamic selection and data preprocessing for imbalance learning," *Neurocomputing*, vol. 286, pp. 179-192, Apr. 2018. Article (CrossRef Link).

[38] S. Bagui, X. Fang, E. Kalaimannan, S. C. Bagui, and J. Sheehan, "Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features," *Journal of Cyber Security Technology*, vol. 1, no. 2, pp. 108-126, Jun. 2017. Article (CrossRef Link).

[39] H. Oliff, and Y. Liu, "Towards industry 4.0 utilizing data-mining techniques: a case study on quality improvement," *Procedia CIRP*, vol. 63, pp. 167-172, Mar. 2017. Article (CrossRef Link).

[40] H. A. Nguyen, and D. Choi, "Application of data mining to network intrusion detection: classifier selection model," in *Proc. of (APNOMS)*, Beijing, China, pp. 399-408, 2008.

[41] H. Erol, B. M. Tyoden, and R. Erol, "Classification performances of data mining clustering algorithms for remotely sensed multispectral image data," in *Proc. of 2018 (INISTA)*, Thessaloniki, Greece, pp. 1-4, 2018.

[42] SciKit-Fuzzy. [Online]. Available: https://pythonhosted.org/scikit-fuzzy/overview.html

[43] T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.,* vol. SMC-15, no. 1, pp. 116-132, Feb. 1985. Article (CrossRef Link).

[44] H. Ma, Z. Hu, L. Yang, and T. Song, "User feature-aware trustworthiness measurement of cloud services via evidence synthesis for potential users," *J. Vis. Lang. Comput.,* vol. 25, no. 6, pp. 791-799, Dec. 2014. Article (CrossRef Link).

[45] S. Wang, L. Huang, L. Sun, C.-H. Hsu, and F. Yang, "Efficient and reliable service selection for heterogeneous distributed software systems," *Futur. Gener. Comp. Syst.*, vol. 74, pp. 158-167, Sep. 2017. Article (CrossRef Link).

[46] R. Salla, H. Wilhelmiina, K. Sari, M. Mikaela, M. Pekka, and M. Jaakko, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behav. Processes*, vol. 148, pp. 56-62, Mar. 2018. Article (CrossRef Link).

[47] D. Karaboga, and E. Kaya, "An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training," *Appl. Soft. Comput.*, vol. 49, pp. 423-436, Dec. 2016. Article (CrossRef Link).

[48] A. Kaur, and K. Kaur, "Statistical comparison of modelling methods for software maintainability prediction," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 23, no. 06, pp. 743-774, Oct. 2013. Article (CrossRef Link).

[49] B. Kitchenham et al., "Robust statistical methods for empirical software engineering," *Empir. Softw. Eng.*, vol. 22, no. 2, pp. 579-630, Jun. 2017. Article (CrossRef Link).

Muhammad Hasnain completed his PhD from Monash University Malaysia Campus. He received his Master of Computer Science (MCS) degree from (UAAR) and Master of Science in Computer Science (MSCS) degree from Abasyn University Pakistan. His research focus is on regression testing, fuzzy logic, machine learning, web services performance testing, and web services selection and ranking. He has published several papers in reputed journals and conferences.

Imran Ghani is an Associate Professor at Computer and Information Sciences Department, Virginia Military Institute, Lexington, VA, United States. He received his Master of Information Technology Degree from UAAR (Pakistan), M.Sc. Computer Science from UTM (Malaysia) and Ph.D. from Kookmin University (South Korea). His research focus includes agile software development practices, semantics techniques, cyber security, web services, and software testing.

Muhammad Fermi Pasha received his PhD degree in computer science from Universiti Sains Malaysia in 2010. After his PhD degree, he worked as a research fellow at Universiti Sains Malaysia. Currently he is working as a Senior Lecturer in the School of Information Technology in Monash University Malaysia. His research interest is focused on computational neuroimaging, intelligent network security traffic analysis, and healthcare and radiology IT with emphasis on big data. He is an IEEE member and currently supervising PhD students in the latter mentioned research areas

Seung Ryul Jeong is a Professor in the Graduate School of Business IT at Kookmin University, Korea. He holds a B.A. in Economics from Sogang University, Korea, an M.S. in MIS from University of Wisconsin, and a Ph.D. in MIS from the University of South Carolina, U.S.A. Dr. Jeong has published extensively in the information systems field, with over 100 publications in refereed journals like Journal of MIS, Communications of the ACM, Information and Management, Journal of Systems and Software, among others. Dr. Jeong's areas of interest are Systems Implementation, Process Management, and Machine Learning.