

Received April 23, 2020, accepted May 2, 2020, date of publication May 12, 2020, date of current version May 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994222

Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking

MUHAMMAD HASNAIN¹, MUHAMMAD FERMI PASHA¹, (Member, IEEE), IMRAN GHANI², MUHAMMAD IMRAN³, MOHAMMED Y. ALZHRANI⁴, AND RAHMAT BUDIARTO⁵

¹School of IT, Monash University Malaysia, Subang Jaya 47500, Malaysia

²Department of Mathematics and Computer Sciences, Indiana University of Pennsylvania, Indiana, PA 15705, USA

³Next Bridge (Pvt.) Ltd., Lahore 54000, Pakistan

⁴Information Technology Department, College of Computer Science and IT, Albaha University, Al Bahah 65527, Saudi Arabia

⁵Computer Engineering and Science Department, College of Computer Science and IT, Albaha University, Al Bahah 65527, Saudi Arabia

Corresponding author: Muhammad Hasnain (muhammad.malik1@monash.edu)

ABSTRACT To accurately rank various web services can be a very challenging task depending on the evaluation criteria used, however, it can play an important role in performing a better selection of web services afterward. This paper proposes an approach to evaluate trust prediction and confusion matrix to rank web services from throughput and response time. AdaBoostM1 and J48 classifiers are used as binary classifiers on a benchmark web services dataset. The trust score (TS) measuring method is proposed by using the confusion matrix to determine trust scores of all web services. Trust prediction is calculated using 5-Fold, 10-Fold, and 15-Fold cross-validation methods. The reported results showed that the web service 1 (WS1) was most trusted with (48.5294%) TS value, and web service 2 (WS2) was least trusted with (24.0196%) TS value by users. Correct prediction of trusted and untrusted users in web services invocation has improved the overall selection process in a pool of similar web services. Kappa statistics values are used for the evaluation of the proposed approach and for performance comparison of the two above-mentioned classifiers.

INDEX TERMS Web services, trust prediction, web services selection, binary classification, fuzzy rules, confusion matrix.

I. INTRODUCTION

Trustworthiness of web services has a significant role in the ranking of web services. Web services can be ranked based on the requesters' demands [1]. For example, two web services have similar functionality, one web service is more used as compared to another one; then possibly the selected web service is usually a more trusted. Web services selection and ranking is a problem that can be addressed through the classification mechanism of non-functional quality attributes. Quality attributes such as response time, throughput, availability, and security have different weights, which are principal for ranking of web services [2]. In the latter study, three categories of web services ranking techniques objective, subjective and hybrid were discussed. Expert opinion is not considered in the objective category of ranking approaches. On the other hand, the subjective category considers expert opinion or subjective judgment. However, lack of experience

The associate editor coordinating the review of this manuscript and approving it for publication was Zhangbing Zhou¹.

may affect the results of a subjective category of techniques. Subsequently, the hybrid of objective and subjective categories of technique can be useful to overcome the limitations of the existing techniques. Our proposed fuzzy-based users' trust prediction approach involves the end-users values of quality attributes and then rank web services by calculating the trust score of web services.

The confusion matrix is widely used in machine learning for supervised classification or determination of the behavior of classification models [3]. The square structure of a confusion matrix is represented through rows and columns, where rows are the actual classes of the instances, and columns are the predicted classes [4]. For the binary classification, a confusion matrix is represented as 2×2 matrix. For a confusion matrix, four measures, namely, 'true positive' (TP), 'true negative' (TN), 'false positive' (FP), and 'false negative' (FN), have been reported. For the multiclass problem, a confusion matrix with the k class has a $k \times k$ confusion matrix [5].

Confusion matrix is applied to evaluate the performance of classifiers on datasets. Font *et al.* [6] used the confusion

matrix to distinguish the predicted values and real values of model elements in software engineering. Four confusion matrix measures such as TP, FP, TN, and FN were used for the classification of faulty and non-faulty classes of Java programs.

Although multiple classification has an extensive background, but studies with regards to multiple web services instances classification are relatively scarce [50], [51]. Existing studies on the multiple classifications show that classifiers used for multiple classification are relatively low in performance accuracy. Due to this reason, we observed the restricted applications of existing multiple classifiers. We have found in [51] that multiple classification models do not outperform the single classifier. The authors in the lateral mentioned study proved their claim by using statistical analysis of multiple classification and binary classification. Both standard deviation and coefficient of variations for testing multiple classifiers remained higher for the single classifiers. Based on the findings of these studies, we can reveal that still, the classifiers can handle efficiently the binary classification problem rather than handling the issue of multiple classifications of web services instances.

The concept of trust prediction for web services is not new in the research domain of web services and estimation of 'quality of services' (QoS). Su *et al.* [7], proposed a trust-awareness approach for the prediction of reliable and personalized QoS features. Users' reputation was determined by clustering the information obtained from similar users to identify the clusters of users and invoked web services. Web service trustworthiness is dependent on users, and it may be maintained in the inappropriate clusters. As a result, this inappropriate clustering affects the trustworthiness of certain web services. To address this issue, we propose an approach with the use of a confusion matrix measures. Our focus is on the binary classification of web services from invoked web services by using the obtained feedback in terms of the throughput and response time metrics values. We measure users' trust from the performance evaluation of quality metrics. Both, response time and throughput come under performance category of quality metrics.

The well-known fact regarding the performance of web service is its reflection from functional and non-functional quality attributes. Response time and throughput are two vital considered attributes in studies [64]. QoS based ranking of web services is appropriate, employing the quality attributes as mentioned earlier. Moreover, Mao *et al.* [65] considered throughput and response time as quality attributes to conduct the experiments for QoS based ranking of web services. Evaluation of the most web services ranking approaches is performed on the real-world dataset that is composed of two QoS attributes (throughput and response time) [64]. Somu *et al.* [66] also performed trust centric ranking of web services by using the throughput and response time quality attributes. Based on the existing literature and understanding of the QoS attributes, it is appropriate to use throughput and response time as the most popular quality attributes

because web services users mostly expect low response time and high throughput from service providers [67]. Therefore, the trustworthiness of a web service is more relevant to the performance evaluation of a web service, which is derived by using QoS attributes.

The proposed approach exploits the values of both monitored QoS metrics and mentioned in the 'service level agreement' SLA document [39]. Untrustworthy users were identified with the assumption that the majority of users were honest as their majority opinions were consistent. In contrast, dishonest users provided a low rating without any consistency in their opinions. This assumption can be further discussed in future research works because no web services QoS metric has been used for the evaluation of the proposed approach.

Trust is defined in different contexts. Trust on eBay, and Amazon has been measured by using the users' past interaction because trust is relational [8]. For instance, two users of web services interact with each other, as a result of the interaction, their relationship strengthens, and trust evolves from their mutual exchange. In addition to it, trustworthy and reputed web services have been defined as services which are inherently secure, reliable, and available despite disruption from the environment, and human errors [41]. The author points out the requirements of secure web services that ensure the users' trust in web services. A trusted web service is reliable as well as provides high throughput and low response time [42].

Suppose a web service consumer asks for the best services that meet requirements Re as $(r_1, r_2, r_3, \dots, r_n)$. Standard attributes such as response time, cost, and availability along their levels are well defined in the SLA document by service providers. Trust reputation model proposed in [43] is evaluated on the latter mentioned three quality attributes where they find that web services consumers are more interested in completing their transactions with the low response time rather than focusing on the high availability, and cost attributes. It means, a web service consumer is more oriented towards the short response time to complete transactions and shows his trust as feedback. Web services consumers rate their invoked web services differently in terms of QoS properties. For instance, users a and b rate high throughput, and low response time, while another user ' c ' rates the same services with the low performance (throughput) and high response time. Subjective perception of QoS attributes may cause the differences in rating by users [44]. As users, a , and b may think that it is good if their invoked service responds within one second. On the other hand, user c may have not a high requirement, and he would like those web services which respond within 20 seconds. We can specify that users a , b , and c have provided their trust values by differently rating web services.

Web services selection approach proposed in [40] is aimed to evaluate the security as a big challenge of web services. Researchers mentioned that the security of web service is further related to confidentiality and privacy aspects. This is because a web service is more reliable than another web

service in confidentiality, and the same web service may be weaker in security in comparison with another web service. Therefore, web services users find it hard to resolve the selection and ranking of web services as they lack expertise. Other than confidentiality and privacy features can be used to address the security issue of web services that, in turn, helps in determining the trust of users in web services.

Trust prediction of web services can be approached as a ranking problem. Ranking encompasses several issues, such as selection, recommendation, and testing of web services. The main objective of trust prediction is to calculate the users' trust in the invoked web services. Then, the calculated users' trust is used to rank web services from a pool of web services accessed by the same users of various regions. Our ultimate goal of ranking is to identify the web services with the high trust score of users and prioritize them for better future selection of web services.

Contributions of this paper are as follows:

- This paper proposes a trust prediction binary classification approach by using QoS attributes of web services.
- This paper proposes fuzzy rules to provide ground truth for training and evaluation of binary classifiers.
- This paper proposes an application of the confusion matrix measures to evaluate the ranking of web services.

In the remainder of this paper, section 2 presents the relevant literature on the existing trust and confusion matrix topics. Section 3 presents the proposed approach; section 4 presents results and discussion; section 5 presents the impact of dataset size on the trust prediction precision; section 6 presents threats to validity; section 7 concludes the proposed work along with future research implications.

II. LITERATURE REVIEW

In this section, we present a review of the existing primary studies on the classification with regards to the confusion matrix. We also discuss a few significant approaches proposed for QoS prediction in the literature.

Polat *et al.* [9] used four measures, namely, TP, TN, FP, and FN, of the confusion matrix, to determine whether patients have optic nerve disease or not. These researchers exploited TN for patients with optic nerve and TP for healthy individuals, reported the results with the confusion matrix, and used TP and TN to classify individuals as either diseased individuals or healthy individuals. For binary classification, the use of TP and TN can accurately predict instances. Choudhury and Bhowal [10] used confusion matrix measures to predict the true and false instances of the network attacks. Binary classifiers were used to represent the attacked and normal classes for network intrusion detection. Based on the confusion matrix measures, these researchers developed a 'false positive rate', (FPR), 'false discovery rate', (FDR), and 'negative prediction rate' (NPR) measures. To predict the possibility of the accurate and inaccurate classification of network attack and normal instances, three developed measures were used for accuracy metrics. To increase the accuracy of the anomaly detection system, Aljawarneh *et al.* [11] used

a hybrid approach of classifiers to address the issue of high percentages of the FP instances. Along with the proposed hybrid approach, feature selection and reduction are required to find the maximum number of attacks on a network system.

Al-Obeidat and El-Alfy [12] proposed an approach to address the space issue between yes and no in binary classification. Their decision tree generates rules, which have incredibly crisp intervals, and using the fuzzy membership to an object of a class can address marginal space issues between yes and no. The main objective behind the proposal of the hybrid approach is to classify internet traffic through classification and interpretation.

In the literature, the trust prediction of web services is presented in various means and names. Ding *et al.* [13], combined QoS prediction and estimation of customer satisfaction in their proposed approach known as CStrust, which is used to release the customer satisfaction information on web services. The main difference between CStrust approach and our proposed approach is that the CStrust evaluates the cloud web services. In contrast, our proposed approach focuses on web services that use open standards, such as 'extensible markup language' (XML), 'web services description language' (WSDL), and 'simple object access protocol' (SOAP).

To manage the "system-level agreement" (SLA), QoS prediction is a significant tool. To know the behavior of services consumers, Hussain *et al.* [14] compared the results of ML approaches to time series approaches. With the objectives of knowing the services violation and avoidance of penalties, service providers could benefit from the ML-based QoS prediction. Somu *et al.* [15] proposed the web services' ranking algorithm to identify the most trustworthy web services. The proposed approach employed hyper-graph partitioning and time-varying mapping method to identify the similar services providers. Moreover, the use of "hyper-graph-binary fruit fly algorithm" (HBFFOA), which employs hypergraph partitioning, and time-varying function for the identification of similar services, helped in determining the optimal ranking of web services.

Trust assessment of web services through fuzzy-based credibility was undertaken by Saoud *et al.* [16], and they pointed out the limitations of those trust-based web services selection approaches that involved the end-users rating. Uncertainty and bias were the concerns of researchers that affected the end-users ratings for web services. A fuzzy-based model was proposed to address the uncertainty and biases of end-users' ratings of web services. The proposed trust-approach was evaluated on a number of experiments. Results indicated that the proposed approach improved trust quality and robustness.

To address the problem of the accurate prediction of unknown QoS values, Ma *et al.* [17] proposed the collaborative filtering that outperformed the existing approaches in the accurate prediction of missing values. The main difference between the collaborative filtering approach and our proposed approach is that the former considers the missing

values, while the latter uses throughput and response time values as feedback given by users. None of the two studies mentioned above show trust prediction via classification. Therefore, the proposal of our approach is mainly based on the binary classification along with the confusion matrix and k-fold cross-validation (CV) method, which divides data points into a fixed number of folds of the data. K-fold cross-validation ensures that data in each fold is at least once tested.

Wang *et al.* [45] proposed a trustworthiness of the web services selection approach that involved collaboration reputation in social networks. This study offers a web services selection process with the aims of including and excluding the services with the high and low reputations, respectively. The reputation of a web service is increased due to more interaction rounds of web services. This study also defines the reliability levels of web services which are given as follows:

- Level 1: Good web service (0.9-1)
- Level 2: Normal web service (0.3-0.9)
- Level 3: Bad web service (0.0-0.3)

The findings of this study indicated that web services' reputation was fairly computed that distinguished web services from the selection process by using the defined three levels. This study showed the scalability issue because the proposed approach was less effective or even was unable to work on a small community of web services. Therefore, a web service ranking approach is required to address the scalability issue within the small community of web services. A ranking approach can be proposed to rank web services for a small number of web services.

Mehdi *et al.* [46] proposed a trust and reputation-based web services selection approach. This proposed approach used correlation information among various QoS metrics, which resulted in estimating the trustworthiness of web services. Researchers exploited two statistical distributions known as Dirichlet and generalized Dirichlet, which represented the multiple correlated metrics. For instance, the throughput QoS metric is correlated with response time and availability of QoS metrics. It has been stated that the increase in throughput value results in increasing the availability score of web services and decreasing the response time value of users' requests. Reliability as a QoS metric has a strong correlation with response time and throughput QoS metrics.

Moreover, the lateral mentioned study proposed the aggregate reputation feedback algorithm to deal with the malicious feedback, which propagates between interacting web services. Results endorsed that the proposed approach, along with the algorithm is capable of showing the better determination of trustworthiness in comparison with the state of the art approaches and algorithms. Before the latter mentioned research work, Deng *et al.* [44] proposed a CTrust framework to evaluate the trustworthiness of cloud web services by combining the customers' satisfaction estimation and QoS prediction. Both of these research studies were aimed at addressing the trustworthiness issue of web services. However, the previous research explores QoS metrics for the estimation of trustworthiness web services, while the

latter study is investigating both trustworthiness and QoS prediction.

In a recently published work, Tibermacine *et al.* [47] proposed a method to determine the reputation of similar web services. Researchers have employed the application of support vector regression algorithm to estimate the unknown QoS values of web services from their known values. The proposed reputation estimation method has been evaluated on two web services QoS datasets. The proposed reputation estimation method is mainly focused on determining the reputation of newcomer web services because reputation is a similar issue to trust, and security issues of web services deployment. Therefore, the trust and security of web services can be undertaken in future works to ensure the quality of web services because users show their high confidence in the high-quality web services as compared to low-quality web services.

Mao *et al.* [52] pointed out that trustworthiness was a significant indicator for the selection and recommendation of services. Trust prediction based on the QoS value is a challenging task due to a non-linear association between QoS values and the trust rate of services. Although neural networks (NNs) have the capability of trust prediction, but their parameters' setting further requires research work to improve their performance. Therefore, researchers in the lateral-mentioned study introduced particle swarm optimization (PSO) to enhance the environment of NNs to trust prediction of cloud web services accurately. For the evaluation of the PSO supported trust prediction, experiments were performed on the public QoS dataset. The results showed that NNs with PSO outperformed the basic NNs in the trust-based classification of web services.

Somu *et al.* [53] called that trustworthiness was itself a quality metric used for the assessment of the quality of web services. It has been earlier mentioned in [52] that the trust prediction from QoS attributes is a challenging task. To overcome this problem, a multi-level 'Hypergraph Coarsening based Robust Heteroscedastic Probabilistic Neural Network' (HC-RHRPNN) was proposed in [53] for trust prediction of cloud web services. Informative samples were identified by employing the hypergraph coarsening of HC-RHRPNN. Afterward, the training of the proposed model was done by using the identified informative samples. Moreover, informative samples improved the prediction accuracy and also minimized the execution time. The proposed HC-RHRPNN outperformed the earlier proposed neural networks with regards to performance. We have observed an extension in the latter work in [54] in which researchers used artificial neural networks (ANNs). The PSO technique was applied to train the ANN. Moreover, 'binary particle swarm optimization' (BPSO) has been used for the selection of quality attributes. The evaluation of the proposed approach was performed by using a public QoS dataset. The results showed that the prediction accuracy of the proposed model remained better than the existing models. Further works require to improve the trust prediction accuracy of the chosen models.

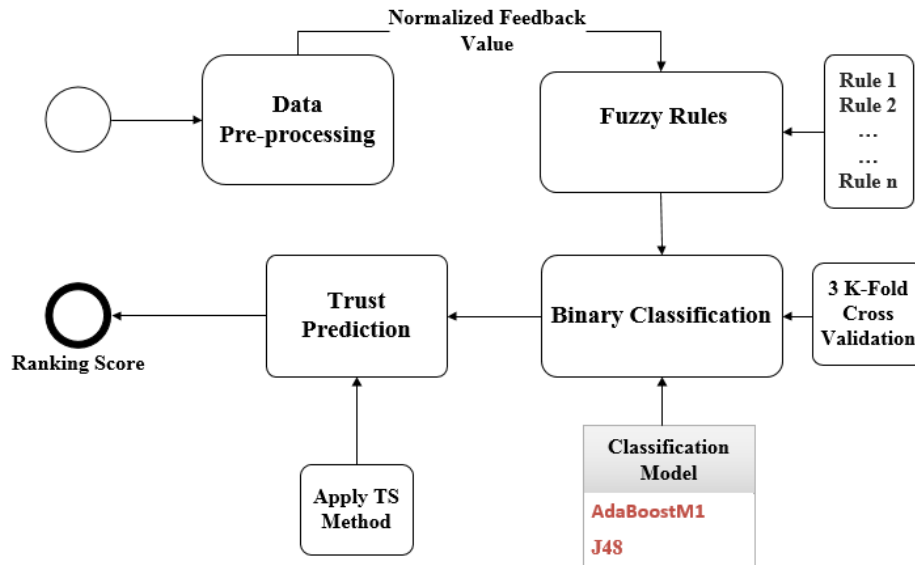


FIGURE 1. Phases of the proposed approach.

→ Researchers in studies [53], [54] studied the trust prediction of web services regarding the evaluation of users’ feedback in terms of quality attributes. Contrary to these studies, Nivethitha *et al.* [55] highlighted the issue of the selection of trustworthy web cloud services providers (CSPs). This problem arises due to the varying functional and non-functional requirements of web services users. Also, the complexity in the selection of cloud web services is increased due to the addition of new web services. To evaluate the quality of CSPs, proposed ‘rough set theory-based hypergraph-binary fruit fly optimization’ (RST-HGBFFO), a bio-inspired approach was used to select the most optimal trust measure parameters (TMPs).

III. PROPOSED TRUST PREDICTION APPROACH

This section presents the strategy alongside four phases involved in proposing the trust prediction of web service users. We present the structure of the proposed approach and then discuss the main phases used for our proposed approach. The proposed phases of the trust prediction approach are shown in Fig. 1. The phases of the proposed approach are discussed in the following subsections.

A. DATA PREPROCESSING

To improve the accuracy of binary classifiers on the numerical dataset, we preprocessed the data of chosen web services. This phase of the proposed approach involved the pre-processing of web services data obtained from the GitHub WS-Dream data repository. To normalize the data, we used the min maxim normalization approach shown in the following Eq. (1).

$$Z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where x_i denotes the value of a quality attribute, and $\max(x)$, and $\min(x)$ denote the maximum and minimum values on all value of the given quality attributes. Normalized data were stored as.csv in Excel files, which were subsequently used for binary classification of web service instances.

Several normalization methods have been proposed in the literature. The most popular methods include min-max and z-score normalization, as discussed in [75]. The first method as min-max is used to normalize the features in the range [0, 1], as shown in Eq. (1). The min-max normalization method helps to preserve the association among the ordinal input data [76]. Normalization methods based on mean and standard deviations of the data do not show consistent performance because values of these measures vary over time. Since the values of both attributes (throughput and response time) are based on historical information and do not change with time, so the use of min-max normalization is more appropriate in this study.

B. FUZZY RULES

In the second phase, the feedback input is given, and every input (throughput + response time) is matched to every fuzzy rule given in the following. Every combined input data from TP and RT is processed according to the membership function. Six fuzzy rules are constructed to handle the binary classification of web services instances. A change in the number of quality metrics can be enforced manually by updating the fuzzy rules. The association between quality metrics and fuzzy rules can be adjusted by adding new fuzzy rules.

To convert the crisp input values of response time and throughput metrics, we have proposed a fuzzy system that is based on three main steps, namely, fuzzification, inference, and defuzzification. The first step as fuzzification decomposes the input and output into one or more than one fuzzy set.

In the inference step, our proposed IF-THEN rules are used to compute the fuzzy output from fuzzy input, as given in the following. In the defuzzification step, crisp value is obtained from the conversion of fuzzy values by using the membership function. We propose to use the Sugeno Fuzzy Model [18] for the identification of non-linear relationships; between two variables (response time and throughput).

Due to a complete set of fuzzy rules, inconsistencies among fuzzy rules decrease. With the increased number of the exponential rules and linguistic variables and labels, domain experts need to aware of the differences between rules and variation demonstrated in output. Therefore, the proposal of rules relevant to the problem at hand and which are more intuitive to a domain expert are generated [77]. Thus, high-level rules based on the 'IF-THEN' expression are preferred over the complex statements. The fuzzy If-Then rules are proposed to represent the relationship between variables. These rules occupy a form such as "If antecedent proposition Then the consequent proposition." A linguistic model is capable of capturing the qualitative as well as high uncertain knowledge by using the If-Then rules as follows:

R: If x is A_i then y is B_i

In terms of classification, response time and throughput instances can be naturally considered fuzzy. Thus their behavior is not clear cut, especially when different users report the varying values of instances of both metrics. Before our study, Liu *et al.* [48] proposed fuzzy rules to train the classifiers on the text data. Ambiguous and unclear speeches cannot be easily classified, and hence, fuzzy rules can solve the classification of complex instances into one or more than one class. The latter-mentioned fuzzy-based study inspired us to propose a method to make more transparent the instances of both metrics to which category they belong and then train and evaluate the classifiers on those instances of web services. Additionally, a set of fuzzy rules can be designed to decide the complicated classification of instances of web services. A simple heuristic rule helps reduce time consumption on the training of models and computation complexity [49]. However, such types of proposed methods rely on manual observation with regards to the construction of fuzzy rules.

We have used a limited number of linguistic terms translation for supporting the binary classification of web service instances and handle the trust-based ranking of web services. These linguistic terms have been extracted from the prior knowledge as well as expert experience [68]. We keep linguistic terms translation small due to limited existing knowledge and experts' expertise.

A natural way to express numerical values is through the use of linguistic phrases. It is easier to say, very high, high, medium, low, and very low, rather than providing the numerical values. As in our case, web services instances have numerical quantities. The concept of fuzzy sets introduced by Zadeh [69] provides a suitable way to express the imprecise statements. A quintuple proposed in [70] has been used to

characterize a linguistic variable as follows:

$$(n, T(n), X, G, M)$$

where n expresses the name of a variable, $T(n)$ represents the term set of n , and it is the set of names of linguistic values of n , and each value is defined as a fuzzy variable on X . Moreover, G is called as a syntactic rule to generate the name of values with regards to n ; and M represents the semantic rule used to associate each value with its meaning. Also, n being a particular, which is produced by G , is known as a term.

Definition: If the trust of a user in web services is represented by a linguistic variable, then a term set T_a can be of the following form:

$$T_a = \text{Very high, high, medium, low, very low}$$

Each linguistic term above given is associated with the fuzzy set defined on the domain $[1, 0]$. Very high can be associated with near to 1, and very low can be linked near to 0; high can be linked to 0.8; medium can be linked to 0.6, and low can be linked to 0.4.

A fuzzy rule is the combination of linguistic statements, which are used for decision making in assigning inputs or outputs with regards to classification. Hence, this decision-making through linguistic statements is known as knowledge engineering. A fuzzy rule follows the structure as providing input for classification and then making decisions for an output. The fuzzy rule is constructed from various sources, such as the opinion of domain experts, knowledge engineering, and historical data analysis [19]. We proposed the use of combined information from existing literature and knowledge engineering for the construction of fuzzy rules [20]. Hence, we used the fuzzy information in the existing studies [21]. We used 'AND' and 'OR' logical operators to express the rules for the classification of web service instances. For rule construction, we maintained the values between 0 and 1. We used data discretization to maintain TP and RT values at equal intervals. We proposed to construct six rules and maintain values in five intervals. We constructed fuzzy rules with the help of logical operators, which have been used in the reference [22] to address the binary classification problem. We presented the construction of six fuzzy rules, as follows.

1) RULE 1

If the throughput value is very high OR the response time is very low, then a user is trusted on certain web service. OR

"If $TP \leq 1.0$ and > 0.8 OR $RT > 0$ and ≤ 0.20 then a user is trusted."

→ We assign a member function value to each part of the statement above. The statement above indicates that inputs 1 (throughput) and 2 (response time) as the feedback from a user. Output 1 (user's trust) results from two inputs such as throughput and response time. The use of the membership function to determine the 'very high' and 'very low' values is known as fuzzification.

2) RULE 2

If the throughput value is high OR the response time value is low, then a user is relatively trusted on certain web services. OR

If $TP > 0.6$ and ≤ 0.8 OR $RT > 0.20$ and ≤ 0.40 , then a user is trusted.

→ For rule 2, we used OR operator to mention that either the TP value was high or RT value was low; then, a user is trusted in a web service.

3) RULE 3

If the throughput value AND the response time value are medium, then a user is untrusted. OR "If $TP > 0.4$ and ≤ 0.6 AND $RT > 0.40$ and ≤ 0.60 , then a user is untrusted"

4) RULE 4

If the throughput value is low AND the response time value is high, then a user is untrusted. OR

"If $TP > 0.2$ and ≤ 0.4 AND $RT > 0.6$ and ≤ 0.80 , then a user is untrusted"

5) RULE 5

If the throughput value is very low AND the response time value is very high, then a user is untrusted. OR

"If $TP > 0.0$ and ≤ 0.2 AND $RT > 0.8$ and ≤ 1.00 , then a user is untrusted"

6) RULE 6

If the throughput value is medium AND the response time value is high, then a user is untrusted. OR "If $TP > 0.4$ and ≤ 0.6 AND $RT > 0.6$ and ≤ 0.80 , then a user is untrusted"

Prior to binary classification phase, we need to translate the linguistic terms into a decision group to align the setup for binary classification. As shown in Table 1, we translated the linguistic terms, such as very high, high, medium, low, and very low, into two groups. Both very high and high linguistic terms are maintained in the same group called c1, and the remaining three linguistic terms, namely, as a medium, low, and very low, were kept in the second group named c0.

TABLE 1. Linguistic terms translation.

Rule No.	Translation	Decision
1	Very High OR Very Low	Trusted
2	High OR low	Trusted
3	Medium AND Medium	Untrusted
4	Low AND High	Untrusted
5	Very Low AND Very High	Untrusted
6	Medium AND High	Untrusted

Fuzzy intervals were fixed at discrete values. In rule 1, the fuzzy value for TP input is set between 0.8 and 1.0, so the lower bound is 0.8, and the upper bound is 1.0. Similarly, terms in other rules obtain weights by decreasing linear function. On the other hand, RT fuzzy value for the linguistic term in rule 1 is fixed between 0.00 and 0.20, so the lower bound is

0.00, and the upper bound is 0.20. Similarly, linguistic terms in other rules obtain weights by the increasing linear function. Fuzzy values approaching the upper bounds or lower bounds have more uncertainties than the centroid.

Conformance checking is aimed at establishing if a system externally observed presents the satisfaction and fulfills some expectations. Therefore, the conformance notion directly relates to the notion of expectations. Conformance measure is widely applied to different challenges, i.e., instance marching. The formal definition of conformance outlines the proximity between linguistic terms. We adopt the proposed fuzzy functional dependencies in [71] and highlight the possibility to determine the conformance of attribute domain, such as (very high, high, medium, low, and very low). More precisely, the conformance checking of rules provides an effective manipulation of linguistic terms to define data dependencies, which are not adequately measured.

We define the attribute of distance (S-distance) to illustrate the proximity relation. This attribute can express the distance between two points. Furthermore, it can be fuzzified into a number of fuzzy sets. For instance, in our case, we define five sets of linguistic terms, as shown in Table 2. The proximity depends upon the expert or a user opinion. As shown in Table 2, that 'very high' and 'high' values of throughput attributes are close to each other as compared to the rest of the sets. For response time attribute 'very low' and 'low' values of sets are close to each other in comparison with the rest of the sets. Based on the defined conformance principle by Sözat and Yazici [72], we present the proximity relation in Table 2. Conformance is also aimed to preserve the interpretability when using the granules with the variable granularity.

TABLE 2. Proximity Relation.

S-distance	Very Low	Low	Medium	High	Very High
Very Low	1	0.8	0.60	0.4	0.2
Low	-	1	0.8	0.60	0.4
Medium	-	-	1	0.8	0.60
High	-	-	-	1	0.8
Very High	-	-	-	-	1

C. BINARY CLASSIFICATION

In the third phase of the proposed approach, the binary classification of web services is performed to classify the web services instances. To create a classification model of web services instances, two top techniques AdaBoostM1 and J48, are implemented. Both classifiers are trained on web services datasets and processed for binary classification. AdaBoostM1 classifier as a boosting algorithm is chosen due to its high accuracy in results. The boosting technique constructs the robust classification model by focusing on the misclassified records of past models [23]. AdaBoostM1 technique gives value to every record or instance. Subsequently, the weight at first is set to 1/n and refreshed on each cycle of technique. The mix of two distinct sorts (boosting, and

decision tree) of techniques is aimed at decreasing the changes in a robust model [24]. Therefore, the selection of boosting and decision tree techniques enhance the robustness and prediction power aspects.

1) AdaBoostM1

AdaBoostM1 is one of the most well-known classifiers of the boosting family implemented in WEKA. This classifier works on the sequential training of models, and each round has a trained model. The algorithm of AdaBoostM1 classifier is shown in the following. Misclassified instances are identified at the end of each round, and they are considered in a new training set that is processed in training a new model [25]. We are dealing with the binary classification in this paper. Therefore, in our experiments, we considered binary features, that is, c1 versus c0 classification. Cortes *et al.* [26], also emphasized using AdaBoostM1 for binary classification. In our experiments, we used J48 to evaluate the results from AdaBoostM1 on the web service datasets. The original algorithm of AdaBoostM1 is given by Chen and Pan [27], as follows:

Algorithm 1 AdaBoost.M1

- 1) Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \dots, N$
 - 2) For $m = 1$ to M :
 - a) Fit a classifier $G_m(x)$ to the training data using weights w_i .
 - b) Compute

$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
 - c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
 - d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \dots, N$.
 - 3) Output $G(x) = \text{sgn}[\sum_{m=1}^M \alpha_m G_m(x)]$.
-

AdaBoostM1 classifier generates a strong classifier from the set of weak classifiers. Because of iterations, and each sample which is not correctly weighted is considered for the next iteration. Both J48 and AdaBoostM1, as supervised binary classifiers, show a better classification performance on the different and multidimensional datasets in comparison with the other conventional classifiers. In a recently published work, Rhmann *et al.* [61] stated that the J48 classifier outperformed the other classifiers in fault prediction. It is proven that both AdaBoostM1 and J48 classifiers have a higher prediction accuracy on datasets in comparison with the rest of the techniques. The selection of two classifiers is due to their advantages: AdaBoostM1 is the productive classification technique with its boosting features and its enhanced characterization rate [62]. On the other hand, J48 construction is based on the simple graphical representation structure for the classification and higher prediction [63].

2) CROSS VALIDATION METHOD

We used the k-fold CV method to evaluate the proposed approach. The CV was used to select a model in practicing the learning problem in n iterations [28]. Three k folds, such as 5, 10, and 15 folds were focused in our chosen CV methods. One of the advantages of using many k-fold methods in our experiments is to avoid the biases and overfitting issues. CV minimizes the generalization of errors. For the former issue, the CV method fits and evaluates the model on separate datasets to ensure that performance evaluation is unbiased [29]. For the five-fold CV, data are randomly split into the k number of subsets. K-1 is used for training, and the remaining subset is used for testing [30]. This process continued until all samples were tested. Similarly, 10-fold and 15-fold CVs were used to train and test the subsets.

D. TRUST PREDICTION

To calculate the classification accuracy of classifiers, the accuracy metric has been primarily used in studies [73], [74]. The accuracy metric involves the confusion matrix measures, as shown in the following Eq. (2) and Eq. (3).

$$\text{Accuracy} = \frac{\text{Total no. of corrected predictions}}{\text{Total no. of predictions}} \quad (2)$$

OR

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

→ Based on the Eq. (3), we have proposed to use the confusion matrix measures in a similar fashion to calculate the trust score (TS) in this study.

Using the confusion matrix, we proposed to determine the "trust score" (TS), which is measured in a percentage score, as shown in Eq. (4). The TS prediction denoted the accurate classification of trusted instances resulting from web service invocations, and then we determine the rank of individual web service from classification results:

$$\text{TS\%} = \frac{TP}{TP + FP + TN + FN} * 100 \quad (4)$$

Eq. (4) shows the TS percentage of instances from invoked web services.

Similar to that in the study of Silva-Palacios *et al.* [31], we derived a relationship between classes from the confusion matrix. The simple interpretation of a confusion matrix was that how a classifier finds it hard to distinguish between the classes. Instead of using directly the four measures of a confusion matrix, we used correctly predicted instances of confusion matrix measures to obtain the maximum information on the trusted instances of web services. As shown in Eq. (4), TS percentage was analogous to the accurate prediction of trusted instances.

IV. RESULTS AND DISCUSSION

This section presents the evaluation of the proposed trust-based ranking approach. We performed experiments on

TABLE 3. WSDL URL of Web Services.

Web Services Id	WSDL Addresses
WS1	http://ewave.no-ip.com/ECallws/CinemaData.asmx?WSDL
WS2	http://ewave.no-ip.com/ECallws/StadiumSynchronization.asmx?WSDL
WS3	http://ewave.no-ip.com/EcallWS/CinemaSynchronization.asmx?WSDL
WS4	http://ewave.no-ip.com/ECallws/StadiumData.asmx?WSDL
WS5	http://ewave.no-ip.com/ECallws/BuyerData.asmx?WSDL

a real-world dataset. Moreover, we report the results and findings of the confusion matrix' evaluation and the proposed trust score (TS) method.

A. DATASET

We used the quality WS-Dream dataset to evaluate the performance of our proposed approach. This dataset was published by Zheng *et al.* [34] with the help of the Planet Lab platform, consisted of the invocation records of 339 users and 5825 web services and is accessible from GitHub and The Chinese University of Hong Kong websites [32], [33]. We have chosen five web services randomly because we plan to include more web services in our future work. Every web service has metadata information along with response time (RT) matrix, and throughput (TP) matrix, which are denoted as *rtmatrix*, and *tpmatrix*, respectively. This dataset is the collection of real-world web services' QoS metrics values by users.

Our preliminary experiments were performed on web services datasets given in the following. Table 3 displays web services datasets with their respective WSDL 'universal resource locator' (URL) addresses. The WS-Dream dataset has been widely used by many researchers in the selection and ranking of web services [35]. We used 20% density information from our web services datasets. In addition to metric values, each web service has web service Id, WSDL address, 'internet protocol' IP address, country, 'autonomous system' (AS), latitude, and longitude properties.

B. ACCURACY RESULTS

In this section, we compared the performance of AdaBoostM1 and J48 by using the information collected from the experiments. Experiments performed on web service datasets accumulated the results of various evaluation metrics.

We presented the accurate classification, Kappa, Precision, Recall, and F-Measure statistics for each classifier on the web service datasets. Table 4 shows the results of these accuracy metrics. Among these accuracy metrics, we used Kappa statistics to evaluate our proposed approach because Ben-David and Frank [36] reported that Kappa statistics show good prediction performance of classifiers in the binary classification problem. Kappa statistics does not ignore the classification that occurs due to mere chances. A high Kappa statistics value indicates that the assignment of instances to a group is not random; AdaBoostM1 and J48 are well-trained to classify web service instances. Therefore, Kappa statistics show the best classification ability of a classifier [37]. We obtained the average Kappa value for each classifier with

regard to the web service datasets. Kappa statistics were used to test the inter-rater reliability or agreement between the predicted and actual instances of web services. The Kappa statistics value varied between 0 and 1. Kappa statistics value of <0.4 showed an extremely low similarity; the value between 0.4 and 0.55 was acceptable; the value between 0.55 and 0.70 indicated a good similarity; the value between 0.70 and 0.85 indicated an extremely high similarity, and the value of >0.85 showed a perfect matching between predicted and actual web service instances.

We can see in Table 4 that AdaBoostM1 classifier outperformed the J48 in the case of the WS1 dataset. The values of Kappa statistics along with the Precision, Recall, and F-Measure accuracy metrics were better for AdaBoostM1 than the J48 classifier. For the WS2 dataset, the AdaBoostM1 classifier showed a higher accuracy at 10 k-fold compared to the accuracy values achieved by the J48 classifier. For WS3-WS5 datasets, both classifiers showed accuracy performance with negligible difference. As we expected, that AdaBoostM1 and J48 classifiers got better accuracies, because they are capable of capturing the web services instances classification in each web service dataset.

Fig. 2 shows the average Kappa statistics of the chosen web service dataset for the binary classification of the users' invoked instances. After ranking the web services, we need to evaluate the proposed approach. Therefore, we used the data of the web services to check the precision of the Kappa coefficient. Kappa coefficient was measured from each classifier.

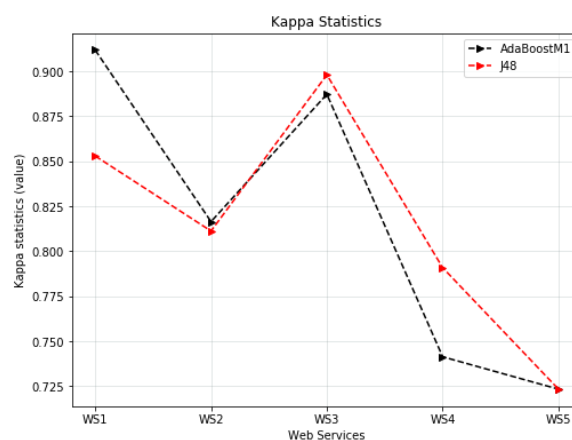
**FIGURE 2.** Average Kappa statistics of AdaBoostM1 and J48 classifier.

TABLE 4. Statistics of accuracy metrics for AdaBoostM1 and J48 classifier.

Datasets	K-Fold	Classifiers	Correctly classified%	Kappa	Precision	Recall	F-Measure
WS1	5-Fold	AdaBoostM1	94.1176	0.8824	0.943	0.941	0.941
	10-Fold		95.5882	0.9118	0.956	0.956	0.956
	15-Fold		97.0588	0.9412	0.971	0.971	0.971
	5-Fold	J48	91.1765	0.8235	0.912	0.912	0.912
	10-Fold		92.6471	0.8529	0.930	0.926	0.926
	15-Fold		94.1176	0.8824	0.943	0.941	0.941
WS2	5-Fold	AdaBoostM1	91.1765	0.7809	0.912	0.912	0.912
	10-Fold		94.1176	0.8539	0.941	0.941	0.941
	15-Fold		92.6471	0.8144	0.926	0.926	0.926
	5-Fold	J48	92.6471	0.8258	0.934	0.926	0.928
	10-Fold		91.1765	0.7877	0.916	0.912	0.913
	15-Fold		92.6471	0.8203	0.928	0.926	0.927
WS3	5-Fold	AdaBoostM1	95.5882	0.898	0.957	0.956	0.956
	10-Fold		94.1176	0.8657	0.944	0.941	0.942
	15-Fold		95.5882	0.898	0.957	0.956	0.956
	5-Fold	J48	95.5882	0.898	0.957	0.956	0.956
	10-Fold		95.5882	0.898	0.957	0.956	0.956
	15-Fold		95.5882	0.898	0.957	0.956	0.956
WS4	5-Fold	AdaBoostM1	88.2353	0.7622	0.890	0.882	0.883
	10-Fold		85.2921	0.6996	0.856	0.853	0.854
	15-Fold		88.2353	0.7622	0.890	0.882	0.883
	5-Fold	J48	88.2353	0.7597	0.885	0.882	0.883
	10-Fold		89.7059	0.7909	0.902	0.897	0.898
	15-Fold		91.1765	0.8217	0.919	0.912	0.912
WS5	5-Fold	AdaBoostM1	85.2941	0.6912	0.861	0.853	0.850
	10-Fold		88.2353	0.7552	0.886	0.882	0.881
	15-Fold		86.7647	0.7233	0.873	0.868	0.865
	5-Fold	J48	86.7647	0.7233	0.873	0.868	0.865
	10-Fold		86.7647	0.7233	0.873	0.868	0.865
	15-Fold		86.7647	0.7233	0.873	0.868	0.865

We obtained the Kappa coefficient average in all cases. The Kappa coefficient, as shown in Fig. 2, indicated good agreement between the predicted and actual web service instances for all web service datasets. The proposed approach, with the help of data mining, provided high precision, and accuracy for all (i.e., WS1 to WS5) datasets. The proposed approach was also evaluated using the J48 in the similar ways as AdaBoostM1. The ability of the proposed approach to determine the complex interaction between predictive web service instances and decrease to the biases was indicated by the Kappa coefficient and other accuracy metrics. For datasets WS1 and WS3, the average Kappa statistics values of AdabBoostM1 were 0.9118 and 0.8872, respectively, thereby showing a perfect agreement between predictive and actual web service instances. Meanwhile, the Kappa coefficient values of J48 for WS1 and WS3 datasets were 0.8529 and 0.8980, respectively. For the remaining datasets (i.e., WS2, WS4, and WS5), the Kappa coefficient values from AdaBoostM1 were between 0.70 and 0.85, which showed an extremely high similarity between predicted and actual web services instances.

C. CROSS VALIDATION RESULTS

→ We present our results from three k-fold CV on web services datasets, as follows. We performed experiments on

TABLE 5. Confusion matrix for WS1 at 5-Fold cross validation.

N=68	Predicted No	Predicted Yes	
Actual No	TN = 31	FP = 1	32
Actual Yes	FN = 3	TP = 33	36
	34	34	

TABLE 6. Confusion matrix for WS1 at 10-Fold cross validation.

N=68	Predicted No	Predicted Yes	
Actual No	TN = 32	FP = 1	33
Actual Yes	FN = 2	TP = 33	35
	34	34	

training several classifiers on our datasets and finally selected AdaBoostM1 and J48, which improved numerical prediction of instances.

→ We determined confusion matrix measures for each of the five web services datasets. The confusion matrix contains the information on the actual and predicted classification of web service instances. Prior to this work, Mehdi et al. [38] used the confusion matrix to present true and predicted classes. We used the confusion matrix with all its measures to compute the evaluation parameters. The percentage of accurately classified web service instances from 5-, 10-, and 15-fold CVs was used as the measure for the model. Tables (5-7) show the confusion matrix results for WS1 by using AdaBoostM1 for three different k-fold CV

TABLE 7. Confusion matrix for WS1 at 15-Fold cross validation.

N=68	Predicted No	Predicted Yes	
Actual No	TN = 33	FP = 1	34
Actual Yes	FN = 1	TP = 33	34
	34	34	

methods. Table 5 shows the obtained confusion matrix results of WS1 from AdaBoost M1. A total of 64 out of 68 instances were accurately classified. Table 6 shows the confusion matrix results of WS1 in the 10 k-fold CV method. A total of 65 out of 68 instances were accurately classified.

Table 7 shows that by adjusting the desired k-fold at a 15-fold CV method, the maximum number of instances, that is, 66 out of 68 web service instances, have been correctly classified. Confusion matrix results for WS1 to WS5 in 5-, 10-, and 15-fold CV methods are shown in Table 8. These results are presented using the AdaBoostM1 technique. Furthermore, we list the number of trusted and untrusted instances detected in each dataset. In Table 8, TP indicates the number of web services instances was correctly assigned from the trusted class of instances, and FP shows those instances which were wrongly assigned. Similarly, TN indicates the number of web services instances were correctly assigned from the untrusted class of instances, and FN shows those instances which were wrongly assigned.

TABLE 8. Number of trusted and untrusted instances detected in web services datasets.

Dataset	k-fold	TP	FP	TN	FN
WS1	5-fold	33	1	31	3
	10-fold	33	1	32	2
	15-fold	33	1	33	1
WS2	10-fold	16	3	46	3
	10-fold	17	2	47	2
	15-fold	16	3	47	2
WS3	5-fold	20	1	45	2
	10-fold	20	1	44	3
	15-fold	20	1	45	2
WS4	5-fold	26	2	34	6
	10-fold	24	4	34	6
	15-fold	26	2	34	6
WS5	5-fold	21	8	37	2
	10-fold	23	6	37	2
	15-fold	22	7	37	2

D. RANKING RESULTS

The main objective of using classifiers within three k-fold validation methods was to identify how the prediction of trusted and untrusted instances of users was performed and interpreted. To interpret the predicted instances accurately, we ranked the web services in terms of the accurate prediction of trusted and untrusted instances.

→ Table 9 shows the computed web service ranking by using Eq. (4) mentioned above. The results in Table 8 were used to determine the average TS percent and the web service ranking.

TABLE 9. Ranking score of web services.

Dataset.	TS%	Average%TS	Ranking
WS1	48.5294+48.5294+48.5294	48.5294	1
WS2	23.5294+25.0000+23.5294	24.0196	5
WS3	29.4117+29.4117+29.4117	29.4117	4
WS4	38.2352+35.2941+38.2352	37.2548	2
WS5	30.2823+33.8235+32.3529	32.1529	3

Table 9 also shows the final ranking of web services from the computed average TS percent values. A web service with the highest average TS percent value was predicted as the most trusted web service from the users. Table 8 shows the simple implementation of our proposed Eq. (4) by using the results in Table 9. The ranking method was mainly based on trust criteria of the average TS percent, showing that WS1 was the most trusted by users with 48.5294% score, and WS2 was the least trusted with a 24.0196% score. Similarly, we computed the ranking score of the remaining web services, namely, WS3, WS4, and WS5, by using our proposed TS percent ranking criteria. We can interpret the results shown in Fig. 3 by considering the trust score calculated from the binary classification of web services instances. Our trust-based web service ranking was based on the accurate prediction of true instances of a given dataset. TP, FP, TN, and FN were four measures of a confusion matrix for binary classification.

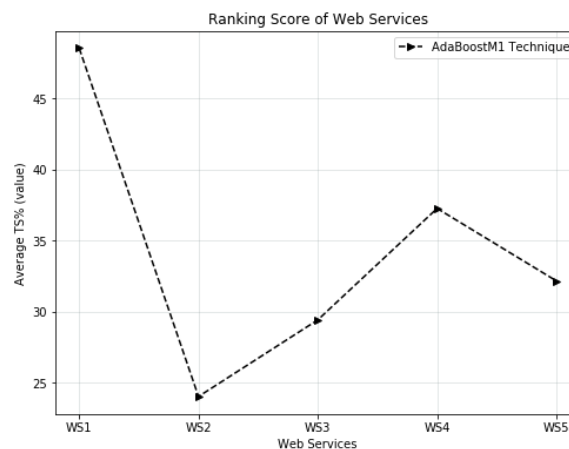


FIGURE 3. Presentation of web services ranking.

E. IMPACT OF QoS ATTRIBUTES VALUES CHANGES ON WEB SERVICES RANKING

Hasnain et al. [56], in their recently published paper, highlighted the effects of several quality attributes. They found the dominating metrics which have a higher impact on the decision making for the selection of web services datasets. For instance, throughput and response time as quality metrics were among top quality metrics with their effects. Since in this study, we are dealing with the latter mentioned two quality attributes; the impact of changes in throughput and

response time metrics can be easily determined. The higher value of throughput instances of web services may change the ranking of web services.

As observed in [57], the lesser value of the QoS criterion has a higher impact on the proposed ranking results. Therefore, low values of quality metrics have effects on the ranking results of our proposed approach. As can be seen from Table 9, the increase in the value of the TS percent method may change the ranking results of the proposed approach. For instance, WS2 web service may get a new ranking if TS percent values are increased. As a result, it can be ranked at position four before WS3.

V. THE IMPACT OF THE DATA-SET SIZE ON THE TRUST PREDICTION PRECISION

It is known that dataset size profoundly influences the performance of a machine learning algorithm. A basic algorithm with lots of data shows the performance edge over the modern algorithms. Liu *et al.* [58] mentioned the datasets which are in tens of thousands in records i.e., Bitcoin and Ciao datasets. In addition to these datasets, the Epinions dataset with regards to review rating has been also used. Similar to earlier mentioned datasets, our proposed approach provides trust and distrust scores. The main difference between the previously used datasets and our dataset is the variance in trust and distrust scores. The highest average TS percent score of WS1 is 48.5294. The trust score may vary due to class imbalance issues. The class imbalance issue may be due to variance in the instances of a dataset.

The second point is the impact of dataset size on the trust prediction accuracy. To improve trust prediction accuracy, correct labeling of classes is significant. To do so, we have chosen the weak classifier, such as AdaBoostM1 and J48, which improve their learning capability. The smaller dataset size requirements to train classifiers may improve their prediction accuracy. In this regard, Wang *et al.* [59] stated that the short training dataset resulted in improving the prediction of the random forest algorithm. This explanation appears to be convincing in view of the results of this study because the training dataset size, in our case, is in tens of reviews of web services users. In addition to it, Heydari and Mountrakis [60] validated that 2% and 5% training dataset size did not show a massive difference in the prediction accuracy of the classifiers. Referring to Table 4 where prediction accuracy results with different K-folds are reported, we observe that the Kappa value alongside (Precision, Recall and F-Measure) are almost higher for both classifiers regarding trust prediction of web services. Table 4 shows us that Precision, Recall, and F-measure accuracy metrics values for both AdaBoostM1 and J48 classifiers are above than 8 value, which indicates a high accuracy from both classifiers.

VI. THREATS TO VALIDITY

This section of the paper presents validity threats to our trust-based ranking approach from the evaluation of confusion matrix measures of web services data.

The first internal threat to the proposed approach is the choice of selection of trust subject of users. There are some other choices to rank the web services. For instance, the security of web service is not directly measured in this paper. The security of web services is more relevant to web services standards and can be handled during the development of web services. Our proposed trust-based ranking approach of web services is evaluated on the web services data, which indirectly measures the confidentiality and reliability of web services.

The external validity of the proposed approach is the selection of web services datasets. Because performed experiments for the evaluation of our trust-based approach are undertaken on the five web services datasets, however, experiments can be performed on using more web services datasets from the same datasets and other published datasets of web services. We plan to include more web services datasets by accessible information from accessible data repositories.

VII. CONCLUSION AND FUTURE WORKS

We developed the web service ranking approach that uses feedback by users in terms of throughput and response time. We proposed fuzzy rules to make binary classification improve the effect by structuring the various conditions of users' feedback. Next, we established the trust prediction formula from confusion matrix measures. We used AdaBoostM1 to predict the trusted and untrusted web service instances and compared accuracy with J48 classification technique. From binary classification of web service instances, we used three k-fold CV methods and determined the trust score of web services. Kappa statistics were applied to evaluate the proposed approach.

This paper has implications for software architects and managers. The first implication of the proposed approach is that architects can build better web services by using the trust features of consumers. The second implication is that web services managers can use the ranking of web services based on users' trust to improve the quality of web services.

ACKNOWLEDGMENT

It is clearly stated that no funding was available for this research. This research article is relevant to the ongoing research in the School of Information Technology, Monash University Malaysia.

REFERENCES

- [1] A. Bawazir, W. Alhalabi, M. Mohamed, A. Sarirete, and A. Alsaig, "A formal approach for matching and ranking trustworthy context-dependent services," *Appl. Soft Comput.*, vol. 73, pp. 306–315, Dec. 2018.
- [2] M. Almulla, H. Yahyaoui, and K. Al-Matori, "A new fuzzy hybrid technique for ranking real world Web services," *Knowl.-Based Syst.*, vol. 77, pp. 1–15, Mar. 2015.
- [3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Cham, Switzerland: Springer, 2013.
- [4] O. Caelen, "A Bayesian interpretation of the confusion matrix," *Ann. Math. Artif. Intell.*, vol. 81, nos. 3–4, pp. 429–450, Dec. 2017.
- [5] R. Rajalakshmi and C. Aravindan, "A Naive Bayes approach for URL classification with supervised feature selection and rejection framework," *Comput. Intell.*, vol. 34, no. 1, pp. 363–396, Feb. 2018.

- [6] J. Font, L. Arcega, O. Haugen, and C. Cetina, "Achieving feature location in families of models through the use of search-based software engineering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 363–377, Jun. 2018.
- [7] K. Su, B. Xiao, B. Liu, H. Zhang, and Z. Zhang, "TAP: A personalized trust-aware QoS prediction approach for Web service recommendation," *Knowl.-Based Syst.*, vol. 115, pp. 55–65, Jan. 2017.
- [8] S. Hamdi, A. L. Gancarski, A. Bouzeghoub, and S. B. Yahia, "TISO-N: Trust inference in trust-oriented social networks," *ACM Trans. Inf. Syst.*, vol. 34, no. 3, pp. 1–32, May 2016.
- [9] K. Polat, S. Güneş, and A. Arslan, "A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 482–487, Jan. 2008.
- [10] S. Choudhury and A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," in *Proc. Int. Conf. Smart Technol. Manage. Comput., Commun., Controls, Energy Mater. (ICSTM)*, May 2015, pp. 89–95.
- [11] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, Mar. 2018.
- [12] F. Al-Obeidat and E.-S.-M. El-Alfy, "Hybrid multicriteria fuzzy classification of network traffic patterns, anomalies, and protocols," *Pers. Ubiquitous Comput.*, vol. 23, nos. 5–6, pp. 777–791, Nov. 2019.
- [13] S. Ding, S. Yang, Y. Zhang, C. Liang, and C. Xia, "Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems," *Knowl.-Based Syst.*, vol. 56, pp. 216–225, Jan. 2014.
- [14] W. Hussain, F. K. Hussain, M. Saberi, O. K. Hussain, and E. Chang, "Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs," *Future Gener. Comput. Syst.*, vol. 89, pp. 464–477, Dec. 2018.
- [15] N. Somu, G. R. M. R., K. Kirthivasan, and S. S. V. S., "A trust centric optimal service ranking approach for cloud service selection," *Future Gener. Comput. Syst.*, vol. 86, pp. 234–252, Sep. 2018.
- [16] 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.
- [17] Y. Ma, S. Wang, P. C. K. Hung, C.-H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown Web service QoS values," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 511–523, Jul. 2016.
- [18] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, Oct. 1988.
- [19] H. B. Yadav and D. K. Yadav, "A fuzzy logic based approach for phase-wise software defects prediction using software metrics," *Inf. Softw. Technol.*, vol. 63, pp. 44–57, Jul. 2015.
- [20] H. Wang, Z. Xu, and X.-J. Zeng, "Hesitant fuzzy linguistic term sets for linguistic decision making: Current developments, issues and challenges," *Inf. Fusion*, vol. 43, pp. 1–12, Sep. 2018.
- [21] M. Bouhentala, M. Ghanai, and K. Chafaa, "Interval-valued membership function estimation for fuzzy modeling," *Fuzzy Sets Syst.*, vol. 361, pp. 101–113, Apr. 2019.
- [22] C. J. Mantas, "A generic fuzzy aggregation operator: Rules extraction from and insertion into artificial neural networks," *Soft Comput.*, vol. 12, no. 5, pp. 493–514, Mar. 2008.
- [23] E. M. Bahgat, S. Rady, W. Gad, and I. F. Moawad, "Efficient email classification approach based on semantic methods," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 3259–3269, Dec. 2018.
- [24] M. A. Mohammed, B. Al-Khateeb, A. N. Rashid, D. A. Ibrahim, M. K. A. Ghani, and S. A. Mostafa, "Neural network and multi-fractal dimension features for breast cancer classification from ultrasound images," *Comput. Elect. Eng.*, vol. 70, pp. 871–882, Aug. 2018.
- [25] A. K. Tripathy and P. K. Tripathy, "Fuzzy QoS requirement-aware dynamic service discovery and adaptation," *Appl. Soft Comput.*, vol. 68, pp. 136–146, Jul. 2018.
- [26] E. A. Cortés, M. G. Martínez, and N. G. Rubio, "Multiclass corporate failure prediction by Adaboost. M1," *Int. Adv. Econ. Res.*, vol. 13, no. 3, pp. 301–312, 2007.
- [27] P. Chen and C. Pan, "Diabetes classification model based on boosting algorithms," *BMC Bioinf.*, vol. 19, no. 1, p. 109, Dec. 2018.
- [28] X. Zhang and Q. Song, "A multi-label learning based kernel automatic recommendation method for support vector machine," *PLoS ONE*, vol. 10, no. 4, 2015, Art. no. e0120455.
- [29] J. Lei, "Cross-validation with confidence," 2017, *arXiv:1703.07904*. [Online]. Available: <http://arxiv.org/abs/1703.07904>
- [30] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.
- [31] D. Silva-Palacios, C. Ferri, and M. J. Ramírez-Quintana, "Probabilistic class hierarchies for multiclass classification," *J. Comput. Sci.*, vol. 26, pp. 254–263, May 2018.
- [32] *WS-Dream*. Accessed: Feb. 5, 2020. [Online]. Available: <https://github.com/wsdream>
- [33] *WS-Dream. Towards Open Datasets and Source Code for Web Services Research*. Accessed: Feb. 7, 2020. [Online]. Available: <http://wsdream.github.io/>
- [34] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Jul. 2013.
- [35] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world Web services," *IEEE Trans. Services Comput.*, vol. 7, no. 1, pp. 32–39, Jan./Mar. 2014.
- [36] A. Ben-David and E. Frank, "Accuracy of machine learning models versus 'hand crafted' expert systems—A credit scoring case study," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5264–5271, 2009.
- [37] P. Shrivastava, K. K. Bhojar, and A. S. Zadgaonkar, "Image classification using fusion of holistic visual descriptions," *Int. J. Image. Graph. Signal Process.*, vol. 8, no. 8, pp. 47–57, Aug. 2016.
- [38] M. Mehdi, N. Bouguila, and J. Bentahar, "Probabilistic approach for QoS-aware recommender system for trustworthy Web service selection," *Int. J. Speech Technol.*, vol. 41, no. 2, pp. 503–524, Sep. 2014.
- [39] M. Tang, X. Dai, J. Liu, and J. Chen, "Towards a trust evaluation middleware for cloud service selection," *Future Gener. Comput. Syst.*, vol. 74, pp. 302–312, Sep. 2017.
- [40] B. Zhou, Q. Zhang, Q. Shi, Q. Yang, P. Yang, and Y. Yu, "Measuring Web service security in the era of Internet of Things," *Comput. Electr. Eng.*, vol. 66, pp. 305–315, Feb. 2018.
- [41] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014.
- [42] Z. M. Aljazzaf, M. A. M. Capretz, and M. Perry, "Trust bootstrapping services and service providers," in *Proc. 9th Annu. Int. Conf. Privacy, Secur. Trust*, Montreal, QC, Canada, Jul. 2011, pp. 195–200.
- [43] H. T. Nguyen, W. Zhao, and J. Yang, "A trust and reputation model based on Bayesian network for Web services," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2010, pp. 251–258.
- [44] S.-G. Deng, L.-T. Huang, J. Wu, and Z.-H. Wu, "Trust-based personalized service recommendation: A network perspective," *J. Comput. Sci. Technol.*, vol. 29, no. 1, pp. 69–80, Jan. 2014.
- [45] S. Wang, L. Huang, C.-H. Hsu, and F. Yang, "Collaboration reputation for trustworthy Web service selection in social networks," *J. Comput. Syst. Sci.*, vol. 82, no. 1, pp. 130–143, Feb. 2016.
- [46] M. Mehdi, N. Bouguila, and J. Bentahar, "Trust and reputation of Web services through QoS correlation lens," *IEEE Trans. Services Comput.*, vol. 9, no. 6, pp. 968–981, Nov. 2016.
- [47] 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.
- [48] H. Liu, P. Burnap, W. Alorainy, and M. L. Williams, "A fuzzy approach to text classification with two-stage training for ambiguous instances," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 2, pp. 227–240, Apr. 2019.
- [49] M.-S. Hosseini and A.-M. Eftekhari-Moghadam, "Fuzzy rule-based reasoning approach for event detection and annotation of broadcast soccer video," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 846–866, Feb. 2013.
- [50] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms," *Expert Syst. Appl.*, vol. 80, pp. 323–339, Sep. 2017.
- [51] D. Wang, X. Tong, and Y. Wang, "An early risk warning system for outward foreign direct investment in mineral resource-based enterprises using multi-classifiers fusion," *Resour. Policy*, vol. 66, Jun. 2020, Art. no. 101593.
- [52] C. Mao, R. Lin, C. Xu, and Q. He, "Towards a trust prediction framework for cloud services based on PSO-driven neural network," *IEEE Access*, vol. 5, pp. 2187–2199, 2017.
- [53] N. Somu, G. R. M. R., K. V., K. Kirthivasan, and S. S. V. S., "An improved robust heteroscedastic probabilistic neural network based trust prediction approach for cloud service selection," *Neural Netw.*, vol. 108, pp. 339–354, Dec. 2018.
- [54] M. Bisi and S. Patel, "A BPSO-ANN model for trust prediction of cloud services," in *Proc. Global Conf. Advancement Technol. (GCAT)*, Oct. 2019, pp. 1–5.

- [55] S. Nivethitha, M. R. G. Raman, O. Gireesha, K. Kannan, and V. S. S. Sriram, "An improved rough set approach for optimal trust measure parameter selection in cloud environments," *Soft Comput.*, vol. 23, no. 22, pp. 11979–11999, Nov. 2019.
- [56] M. Hasnain, M. F. Pasha, I. Ghani, B. Mehboob, M. Imran, and A. Ali, "Benchmark dataset selection of Web services technologies: A factor analysis," *IEEE Access*, vol. 8, pp. 53649–53665, 2020.
- [57] A. Ouadah, A. Hadjali, F. Nader, and K. Benouaret, "SEFAP: An efficient approach for ranking skyline Web services," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 2, pp. 709–725, Feb. 2019.
- [58] S. Liu, L. Zhang, and Z. Yan, "Predict pairwise trust based on machine learning in online social networks: A survey," *IEEE Access*, vol. 6, pp. 51297–51318, 2018.
- [59] H. Wang, R. Magagi, K. Goita, M. Trudel, H. McNairn, and J. Powers, "Crop phenology retrieval via polarimetric SAR decomposition and random forest algorithm," *Remote Sens. Environ.*, vol. 231, Sep. 2019, Art. no. 111234.
- [60] S. S. Heydari and G. Mountrakis, "Meta-analysis of deep neural networks in remote sensing: A comparative study of mono-temporal classification to support vector machines," *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 192–210, Jun. 2019.
- [61] W. Rhmann, B. Pandey, G. Ansari, and D. K. Pandey, "Software fault prediction based on change metrics using hybrid algorithms: An empirical study," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 4, pp. 419–424, May 2020.
- [62] V. Sharma and K. C. Juglan, "Automated classification of fatty and normal liver ultrasound images based on mutual information feature selection," *IRBM*, vol. 39, no. 5, pp. 313–323, Nov. 2018.
- [63] H. Hong, J. Liu, D. T. Bui, B. Pradhan, T. D. Acharya, B. T. Pham, A.-X. Zhu, W. Chen, and B. B. Ahmad, "Landslide susceptibility mapping using J48 decision tree with AdaBoost, bagging and rotation forest ensembles in the Guangchang area (China)," *CATENA*, vol. 163, pp. 399–413, Apr. 2018.
- [64] N. Somu, G. R. MR, A. Kaveri, A. Rahul, K. Krithivasan, and S. Sriram, "BGSS: An improved binary gravitational search algorithm based search strategy for QoS and ranking prediction in cloud environments," *Appl. Soft Comput.*, vol. 88, pp. 1–20, 2020.
- [65] C. Mao, J. Chen, D. Towey, J. Chen, and X. Xie, "Search-based QoS ranking prediction for Web services in cloud environments," *Future Gener. Comput. Syst.*, vol. 50, pp. 111–126, Sep. 2015.
- [66] H. Ma, H. Zhu, Z. Hu, K. Li, and W. Tang, "Time-aware trustworthiness ranking prediction for cloud services using interval neutrosophic set and ELECTRE," *Knowl.-Based Syst.*, vol. 138, pp. 27–45, Dec. 2017.
- [67] F. Qu, J. Liu, H. Zhu, and B. Zhou, "Wind turbine fault detection based on expanded linguistic terms and rules using non-singleton fuzzy logic," *Appl. Energy*, vol. 262, Mar. 2020, Art. no. 114469.
- [68] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," in *Computational Linguistics*. New York, NY, USA: Pergamon, 1983, pp. 149–184.
- [69] R. R. Yager, M. Z. Reformat, and N. D. To, "Drawing on the iPad to input fuzzy sets with an application to linguistic data science," *Inf. Sci.*, vol. 479, pp. 277–291, Apr. 2019.
- [70] M. Vučetić, M. Hudec, and B. Božilović, "Fuzzy functional dependencies and linguistic interpretations employed in knowledge discovery tasks from relational databases," *Eng. Appl. Artif. Intell.*, vol. 88, Feb. 2020, Art. no. 103395.
- [71] M. Sözat and A. Yazici, "A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations," *Fuzzy Sets Syst.*, vol. 117, no. 2, pp. 161–181, Jan. 2001.
- [72] B. Sheng, O. M. Moosman, B. Del Pozo-Cruz, J. Del Pozo-Cruz, R. M. Alfonso-Rosa, and Y. Zhang, "A comparison of different machine learning algorithms, types and placements of activity monitors for physical activity classification," *Measurement*, vol. 154, Mar. 2020, Art. no. 107480.
- [73] F. Lopes, J. Agnelo, C. A. Teixeira, N. Laranjeiro, and J. Bernardino, "Automating orthogonal defect classification using machine learning algorithms," *Future Gener. Comput. Syst.*, vol. 102, pp. 932–947, Jan. 2020.
- [74] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, May 2019, Art. no. 105524.
- [75] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, "Mixture of activation functions with extended min-max normalization for forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.
- [76] P. Hilletoft, M. Sequeira, and A. Adlemo, "Three novel fuzzy logic concepts applied to reshoring decision-making," *Expert Syst. Appl.*, vol. 126, pp. 133–143, Jul. 2019.



MUHAMMAD HASNAIN was born in Bhakkar, Punjab, Pakistan, in 1977. He received the M.Sc. degree in computer science from Abasyn University Islamabad, Pakistan, in 2016. He is currently pursuing the master's degree with the School of Information Technology, Monash University Malaysia. From 2016 to 2017, he worked as a Lecturer with the Army Public College of Management Sciences, Rawalpindi, Pakistan. His research interest is focused on web services quality enhancement.



MUHAMMAD FERMI PASHA (Member, IEEE) was born in Indonesia. He received the Ph.D. degree in computer science from Universiti Sains Malaysia, in 2010.

After his Ph.D. degree, he worked as a Research Fellow at Universiti Sains Malaysia. He is currently working as a Lecturer at the School of Information Technology, Monash University Malaysia. His research interests are focused on computational neuroimaging, intelligent network security traffic analysis, and healthcare and radiology IT with emphasis on big data. He is also supervising the Ph.D. students in the latter mentioned research areas.



IMRAN GHANI was born in Pakistan. He received the Ph.D. degree from Kookmin University, South Korea, in 2010, and the M.Sc. degree in computer science from UTM, Malaysia, in 2007. He worked as a Senior Lecturer with Monash University Malaysia. He is currently working as an Associate Professor of computer science with the Mathematical and Computer Science Department, Indiana University of Pennsylvania. He has published more than 80 research articles in reputed journals and also edited two books. His research interests are focused on software engineering, web services, web mining, and cloud computing. He is currently supervising the Ph.D. students in the latter mentioned research areas.



MUHAMMAD IMRAN was born in Lahore, Pakistan. He received the master's degree in computer science from COMSATS University, Lahore, Pakistan. He is currently working as a Senior Software Engineer in a software industry in Pakistan. His research interests include data mining, machine learning, and software engineering.



MOHAMMED Y. ALZHRANI received the master's and Ph.D. degrees in computer science from Heriot-Watt University, U.K., in 2010 and 2015, respectively. He is currently the Dean of the College of Computer Science and Information Technology, Albaha University, Saudi Arabia. His research interests include model checking and verification, intelligent healthcare systems, and information security.



RAHMAT BUDIARTO received the B.Sc. degree from the Bandung Institute of Technology, in 1986, and the M.Eng. and Dr.Eng. degrees in computer science from the Nagoya Institute of Technology, in 1995 and 1998, respectively. He is currently a Full Professor at the College of Computer Science and IT, Albaha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs.

• • •