American Journal of Business and Operations Research

♦ ISSN: 2692-2967 Vol: 12, No. 02, 2025 ♦ PP. 01–09

Article DOI: https://doi.org/10.54216/AJBOR.120201

FULL LENGTH ARTICLE

Optimized Composition of Business Process Web Services via QoS-Based Categorization Using Decision Tree Classifier and Knowledge-Based Decision Support

Larisa Ivascu[1,*]

[1] Faculty of Management in Production and Transportation, Politehnica University of Timisoara, Romania

Email: larisa.ivascu@upt.ro

Received: June 14, 2024 Revised: September 25, 2024 Accepted: December 13, 2024 ⋆ Corresponding author

ABSTRACT

Determining web services according to Quality of Service (QoS) restrictions is the topic of discussion in this section. Decision tree classifiers are used to accomplish this classification. Because of the ever-changing and expanding nature of online services, it is necessary to accurately categorize them in order to make choosing them more efficient for consumers. It makes use of decision tree techniques, more especially the C5.0 classifier, this is an advancement over older approaches such as the C4.5 classifiers. It incorporates characteristics like as noisy handling, incomplete information administration, and improved decision-making correctness. Web services are classified into four distinct groups: Outstanding, Good, Average, and Poor. These classifications are determined by QoS metrics that include time to response, accessibility, performance, dependability, and success rate. The choice of features is accomplished utilizing an evolutionary algorithm with a wrapper technique with the goal to maximize the effectiveness of this category. This method minimizes the number of repetitive features and improves the method of classification for the purpose of optimization. The resilience and predicted reliability of the algorithm are ensured by additional approaches like as cross-validation and error reduction. These approaches also address difficulties such as overfitting and redundant characteristics. The construction of integrated web services for complicated corporate operations is a particularly valuable use of this technology, which also considerably improves the procedure for making choices for identifying services and consumption. Service 7 stands out with an impressive 98% performance, while Service 6 and Service 3 are also among the top-performing services. Compared to the others, Service 1, Service 2, Service 5, and Service 4 all exhibit comparatively poor results.

Keywords: VoIP ▪ QoS ▪ C5.0 classifier ▪ Decision tree ▪ SWAM

1. INTRODUCTION

It is now possible for varied software programs to communicate and interact with one another in a seamless manner across a variety of locations thanks to the proliferation of web-based services, which have become a vital part of the digital community. A considerable transformation has taken place in the surroundings of distributed computation as a result of their capacity to let diverse systems interact with one another [1]. These systems are often designed using distinct programming languages and operate on various systems. The areas of e-commerce, cloud computing, and numerous other internet-driven fields have been significantly transformed as a result of this change. A significant factor that has contributed to the fast growth of web services is their adaptability, which allows them to provide solutions that are platform-independent and

© 2025 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

reusable for diverse operational needs.

Due to the constantly growing number of services that are now accessible, both businesses and consumers are confronted with the issue of determining which service is the most appropriate for meeting their particular requirements. It is not an easy job to choose the appropriate online service since there is such a wide variety of possibilities available. These options often share capabilities that are similar to one another, but they exhibit significant differences with regard to quality and reliability. When deciding whether or not a web service is suitable for a given purpose, QoS characteristics, which include response time, availability, throughput, success rate, and dependability, play an essential role [2, 3]. The non-functional characteristics of a service that are essential for satisfying both the demands of users and the needs of operations are included by these specifications. As a consequence, there is an urgent need for computerized programs that can classify, rank, and propose online services based on QoS characteristics. As a viable approach to addressing this difficulty, the topic of web service categorization has attracted a lot of interest in recent years. Within the scope of this discussion, classification refers to the process of categorizing online services into various groups according to predetermined criteria, most often their QoS characteristics.

Furthermore, this classification method increases the speed of service discovery and enhances the efficacy of discovering services that are in alignment with user needs. Decision-tree classifiers, which are an instance of artificial intelligence approaches, have come to prominence as a potentially useful tool for resolving the complexities connected with online service categorization [4, 5]. The categorization of items into distinct types is made possible through decision trees, which are complex structures that split data depending on attribute values at each level. Because they provide answers for inductive deductions that are both readily apparent and computationally fast, they are especially useful in situations that include large datasets. As a result of its higher effectiveness in terms of correctness and flexibility, the C5.0 decision tree algorithm is an excellent option for web service categorization jobs. This method is an enhanced iteration of previous approaches such as C4.5 and ID3, and it has exhibited outstanding efficiency. Noise tolerance, management of missing data, and error trimming are some of the elements included in the C5.0 method. These properties, taken together, contribute to the algorithm's increased resilience and flexibility.

Using the QoS characteristics of online services, the algorithm divides services into classifications such as Excellent, Good, Average, and Poor [6]. The first step in the categorization process is the development of a decision tree using training data that reflect the QoS characteristics of a number of different web services. A number of choice criteria are represented by nodes in the tree, and the branches indicate all of the results that may occur depending on those criteria. Consequently, the leaf nodes correspond to the final categorization categories that have been determined [7]. Because of this organized method, the methodology is able to categorize online services that have not yet been viewed with a high degree of accuracy, thereby making it easier to identify effective services.

In order to achieve correct categorization, choosing features is an essential component of the classification process. This process entails determining which quality-of-service characteristics are the most relevant [8]. Feature selection not only lowers the overall size of the dataset, but also removes redundant or useless characteristics that might impair the correctness of the model. Genetic algorithms, which are based on the fundamentals of natural selection, have been used as an efficient way to select features in the suggested approach. These techniques refine an ensemble of potential options in an iterative manner, finally converging on an optimum subset of characteristics. The classifier method's overall effectiveness is improved as a result of using evolutionary algorithms and decision tree classifiers.

Cross-validation is an empirical method used to verify the dependability and generalization of the classification system [9, 10]. Through cross-validation, the dataset is partitioned into various subsets for training and testing. This guarantees that the efficiency of the model is not influenced by particular data partitions. In addition, error-pruning methods simplify the decision tree by deleting branches that contribute only a minor amount to categorization reliability. The complexity of the model and the danger of overfitting are reduced. Overfitting occurs when models operate well on the data on which they have been trained but badly on unseen data.

The developed approach tackles a number of problems associated with traditional categorizing methods [11]. These limitations include managing an abundance of noisy data, calculating the appropriate tree depth, and treating missing attribute values. A substantial classification accuracy of 97% is achieved through the capabilities of the C5.0 algorithm, genetic codes, and error trimming. With this degree of accuracy, the dependability of online provider selection is considerably improved, and consumers are given the chance to make well-informed judgments based on the QoS characteristics that are most important to them. The organization and categorization of web-based services according to QoS characteristics have far-reaching ramifications for a variety of applications. Enterprises may use the categorization model to pick online services that fit their operational requirements, such as lowering response times or optimizing dependability, particularly in relation to e-business [12].

Additionally, in the realm of cloud computing, service vendors can use the categorization structure to give consumers suggestions specifically suited to their speed requirements. The technique also allows combined web services to be built by combining different services to satisfy the needs of complicated individual users [13]. By correctly classifying specific services, the system guarantees that the composite service will have only the elements that are particularly appropriate for it. This results in an increase in both general effectiveness and subscriber satisfaction. To illustrate the proposed approach, a block diagram outlines the primary steps of the categorization process. Obtaining a dataset that contains QoS characteristics for a number of different online services is the first step. The dataset is pre-processed to deal with missing values and remove noise, ensuring that the input data are appropriate for categorization.

The subsequent step entails developing the decision tree through the C5.0 algorithm, while feature identification is carried out simultaneously through genetic approaches [14, 15].

To guarantee the dependability and precision of the produced decision tree, it is verified through cross-validation. The tree then undergoes pruning to lessen detail and improve generalization, ultimately categorizing web-based services into predefined subcategories. A robust and effective structure for website service categorization is produced as a consequence of integrating these elements. This structure addresses the issues presented by the dynamic and varied nature of online services. The technique not only makes service discovery more efficient, but also gives consumers the ability to make fact-based judgments when picking the online services most suitable for their individual requirements.

By automating the categorization process, the structure cuts down on the time and effort necessary for service selection [16]. This enables users to concentrate on using the selected services to accomplish their goals. Categorizing online services according to QoS characteristics is an essential factor that enables efficient and successful service consumption in the digital era. A substantial step forward in this field is represented by the suggested technique, which focuses on decision-tree classifiers, genetic algorithms, and error pruning. Because of its capacity to properly classify services according to their functionality, it guarantees that users can make well-informed choices, maximizing the value generated from web services [17]. The need for robust and extensible categorization methods will only increase as the online ecosystem continues to develop.

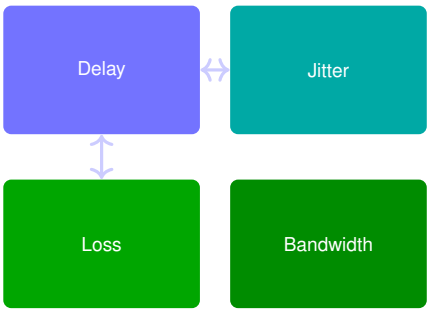

Figure 1. Features of innovation.

Figure 1 illustrates the four basic forms of traffic measured by QoS in networks: bandwidth, delay, loss, and jitter. Bandwidth is defined as the quantity of information that can be transferred across a system during an interval of time [18]. Delay-sensitive applications, such as video conferences or Voice over Internet Protocol (VoIP), need a delay measure that monitors the amount of time required for information to travel from the sender to the receiver. As a crucial indicator for measuring system dependability, network loss is the number of data packets unable to arrive at their intended location. Jitter monitors the variance in packet delivery times and is responsible for interruptions in real-time services such as online gaming or video streaming. These interruptions can negatively affect service performance and user experience. For applications that need high data integrity and low delay, these criteria are extremely important when analysing and enhancing network efficiency.

2. EXISTING WORK

Web-service categorization using quality-of-service metrics was investigated using the J48 method. This research shed light on ranking online services according to performance features using confusion-matrix evaluation to assess the precision of service selection [19, 20]. The technique provides an easy way to categorize services according to response time and dependability and stands out for its efficacy on large datasets. Table 1 summarizes the related work.

The approach may encounter scaling issues in dynamic settings where QoS measures are subject to frequent changes, and the study has shown that it has trouble dealing with datasets that include noise. The C5 classifier is an upgrade to C4.5 that enhanced classification accuracy and handled missing data better. C5.0 is well suited to larger datasets because of improvements in memory economy and overfitting avoidance. However, these models are not very flexible and therefore cannot keep up with the dynamic QoS requirements of online services.

3. OBJECTIVE OF THE RESEARCH WORK

The goal of the research is to develop a reliable system for categorizing web services according to their QoS characteristics. Classifying services as Excellent, Good, Average, or Poor according to critical QoS metrics, including availability, performance, response time, and dependability, is intended to enhance the website-service selection process. An effective model for web-service classification can be constructed using decision-tree classifiers, especially the C5.0 algorithm. Feature selection using neural networks is another area of study; this technique aids in optimizing the categorization procedure by removing superfluous or unnecessary features, thereby improving model reliability. The study further improves the categorization model's generalizability and durability by using cross-validation and error reduction. The end goal is to create a trustworthy, accurate, and scalable system for finding and choosing online services so that organizations and individuals can make better decisions in complex, ever-changing settings.

4. MOTIVATION FOR THE RESEARCH WORK

An ever-increasing variety and number of online services drives the research discussed in this paper. As organizations and consumers increasingly rely on online services, the difficulty of choosing the best service according to non-functional criteria such as availability, dependability, and response time becomes paramount. Manual selection is time-consuming and prone to mistakes because many providers offer comparable features but different performance. Therefore, a scalable, effective, and automatic method is needed to categorize and sort web services according to their QoS qualities.

5. THE PROJECTED METHOD

One of the methods used for classification is the C5.0 algorithm, an improvement on C4.5. C5.0 is applied to large data collections, and its performance and memory use are superior to those of C4.5. The C5.0 model partitions a sample according to the field that offers the greatest information gain.

Table 1. Overview of related work.

Method	Advantage	Strength of the Work	Research Gap
J48 Algorithm [21]	Sorts services according to their quality of service using a decision tree and a confusion matrix.	Effective categorization for massive datasets; simple to understand.	Problems with scalability, controlling noise, and maintaining data integrity.
C5 Classifier [22]	Revised and enhanced version of C4.5; now handles missing data and overfitting more effectively.	Resolves missing values and huge datasets with ease; dependable.	Deals poorly with situations when service quality is changing quickly.
Ranking Model with PCA [23, 24]	Creates smaller subsets of the dataset to facilitate categorization.	Accelerates the process of finding and classifying services.	Potentially needs sophisticated trimming methods due to sensitivity to superfluous characteristics.
Naive Bayesian Network (SWAM) [25]	Sorts features according to their categorization weights using Principal Component Analysis (PCA).	Service rankings using quality-of-service metrics are quite accurate.	Too computationally heavy for real-time applications; performs poorly in fluid settings.
Genetic Algorithm	Quick and easy categorization for simpler datasets.	Simple to execute and uses little computing power.	A decline in performance while dealing with massive amounts of multidimensional QoS data.
Cross-Validation	Reducing unnecessary features is the goal of feature selection using the wrapper technique.	By picking the most relevant QoS criteria, model accuracy is improved.	Relies on static datasets alone; more study is needed to include dynamic QoS changes.
Error Pruning	Contributes to model validation by allowing subsets of data to be tested.	Strengthens the categorization model and makes it more applicable to different scenarios.	Optimization is necessary for complicated multidimensional elements and large-scale datasets.
Artificial Neural Networks (ANN)	Trims decision-tree branches that do not add accuracy.	Makes the model more efficient and aids in avoiding overfitting.	Very severe trimming might cause underfitting.
Support Vector Machines (SVM)	Adapts weights to ensure precise categorization based on learned data.	Extremely flexible, particularly with complicated nonlinear data.	Big datasets need much more processing power and training time.
K-Nearest Neighbors (KNN)	Uses hyperplanes for data classification with the goal of maximizing the margin among classes.	Impressive generalizability; works well with high-dimensional feature spaces.	Complex; requires improved management of nonlinear correlations in QoS data and is computationally demanding.
Random Forest Classifier	Closest-point-based classification is a non-parametric technique.	Easy to understand and use; suitable for datasets of modest to medium size.	Struggles with high-dimensional data and is slower with larger datasets.
Fuzzy Logic Systems	Uses an ensemble of decision trees to improve accuracy.	Constructive, resistant to overfitting, and adept at dealing with missing data.	Potentially inefficient when dealing with high-dimensional datasets on a large scale.
Clustering Methods (K-Means, DBSCAN)	Deals with lack of clarity and precision while classifying data.	Effective when QoS criteria are unclear.	Requires fine-tuning to account for QoS variations; limited by the number of chosen clusters.
Markov Decision Processes (MDP)	Organizes services into groups according to shared QoS characteristics.	Assists in categorizing services based on shared performance measures.	Mathematically difficult and difficult to execute in large-scale real-time scenarios.
Principal Component Regression (PCR)	Models the process of classifying services as a series of decisions.	Beneficial for ever-changing situations with changing QoS circumstances.	Dimension reduction may lead to data loss, which can affect the processing of complicated datasets.
Linear Discriminant Analysis (LDA)	Directs attention to primary components, decreasing dimensionality and boosting classification accuracy.	Facilitates preservation of critical information while minimizing feature space.	Unsuitable for QoS data with complicated nonlinear connections.
Evolutionary Algorithms (EA)	Determines the most effective linear combination of attributes for classifying data.	Useful when classes can be easily separated along a linear path.	Needs substantial processing power and tuning under changing QoS circumstances.

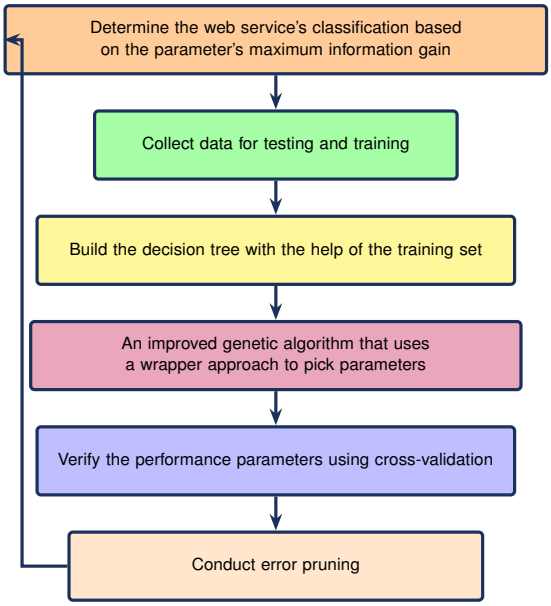

Figure 2. The suggested procedure for web-service categorization.

Through this process, samples can be divided according to the most significant information-gain field.

Among the algorithms that may be used to form a decision tree, some of the most prominent are ID3, C4.5, and C5. Quinlan first presented the ID3 categorization technique, which generates decision trees based on provided data and constraints. Three different kinds of datasets are used. The layout of the file is determined by non-categorical properties, which are the first kind. The initial training set and the test dataset are included in the two subsequent sets. When categorical data are considered, values such as "good," "average," and "satisfactory" can be identified. The ID3 method has difficulty dividing a continuous spectrum of data. This issue was addressed by C4.5, which adheres to the same basic criteria as ID3.

The C5 classifier is a better version of the C4.5 algorithm and adheres to the principles used by C4.5. It also includes several additional features:

1. The extensive decision tree may be viewed as a collection of guidelines.
2. It acknowledges noise and missing data.
3. It provides a solution for overfitting and error pruning.
4. It makes it feasible to anticipate relevant aspects of the situation.

As a result of these additional attributes, the technique has been improved and applied to the categorization of web-based services to obtain an effective compositional structure. Decision trees can have a number of shortcomings, including irrelevant attributes, decision-making borders, duplication of subtrees, continuous categorical attributes, emphasis on important characteristics, and missing attribute values. These issues are addressed by the presented technique through feature selection, error pruning, cross-feature validation, and validation of model functions. In addition, it establishes the level of complexity of the decision tree.

$$J(R) = -\sum_{j=1}^{N} q_j \log_2(q_j) \tag{1}$$

The variability or impurity of a dataset is measured by its entropy. The percentage of samples in class j is denoted by q_j, and there are m classes.

$$F(B) = -\sum_{i=1}^{m} q_{m,n} \log_2(q_{m,n}) \tag{2}$$

All n potential splits (such as Excellent, Good, Average, and Poor) for an attribute A are accounted for by entropy. The likelihood of a sample being a member of class i in subset j is denoted by $q_{m,n}$.

$$G(B) = J(R) - F(B) \tag{3}$$

$G(B)$ is the decrease in entropy when a dataset is partitioned according to attribute B. The entropy of the whole dataset prior to splitting is denoted by $J(R)$, and the weighted entropy after splitting by B is denoted by $F(B)$.

$$GR(B) = \frac{G(B)}{SI(B)} \tag{4}$$

$$SI(B) = -\sum_{j=1}^{N} q_j \log_2(q_j) \tag{5}$$

The intrinsic knowledge acquired by dividing data by B is measured by $SI(B)$. The post-splitting percentage of samples in subset j is denoted by q_j.

$$RT = v_E - v_S \tag{6}$$

When the service finishes responding, the time is represented by v_E. When the service receiving the request is started, the time is represented by v_S.

$$ND = E + M + C \tag{7}$$

Decision-tree depth (E), node count (M), and branch count (C) are added together.

$$E(y) = \frac{1}{1 + F(y)} \tag{8}$$

The fitness coefficient $E(y)$ determines how good a solution y is, while the misclassification error of the selected parameters is denoted by $F(y)$.

$$D = \alpha Q_1 + (1 - \alpha) Q_2 \tag{9}$$

This creates a new candidate D by merging parental solutions Q_1 and Q_2, with the blend factor α controlling the process.

$$N_j = Q_j + \beta\Delta \tag{10}$$

Each unique Q_j is modified by a small amount $\beta\Delta$, which introduces unpredictability.

$$F_q = \sum_{j=1}^{m} z_j f_j \tag{11}$$

Here, f_j is an error at each node in the subtree, with weight z_j.

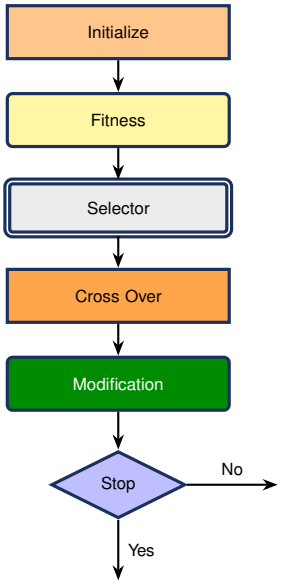

Figure 3. A genetic algorithm using the wrapped approach.

Through feature selection, a subset of traits is chosen from the initial feature set without altering the material values of the initial features. In artificial intelligence and data mining, dimensionality reduction is used for feature selection. Selecting characteristics creates a faster model by eliminating unnecessary, redundant, and noisy features. Genetic methods are applied to populations to obtain more accurate estimates. Natural-selection search heuristics employ a group of individuals to provide a solution to the problem. Genes provide the basis for an individual's characteristics and are linked together to form a chromosome. In this context, genes are factors and chromosomes are solutions. Individuals are represented by strings of ones and zeros.

5.1 Fitness

A fitness score is assigned to every individual, and the system then determines each individual's status for selection and reproduction.

5.2 Selector

Individuals are selected according to their fitness assessments.

5.3 Cross Over

Genetic material is selected from randomly mated parents.

5.4 Modification

Modification preserves population diversity to forestall early convergence.

The process of analysing and contrasting different learning methods is known as cross-validation. This technique separates data into two phases: the first phase is used to learn or train a model, and the second is used to verify the model's accuracy. To achieve higher classification accuracy, model detail may be increased by changing model parameters, which also increases complexity. In artificial intelligence, pruning is a strategy that minimizes the total number of decision-tree branches by deleting areas of the tree that provide little ability to categorize cases. By decreasing overfitting and eliminating portions of a classification that might be generated from noisy or incorrect data, pruning both reduces the difficulty of the final algorithm and improves the assumed precision of the classification method.

The C5.0 model divides samples according to the information-gain field that is most significant. Nevertheless, several challenges are involved in learning decision trees. Choosing how deeply to build the tree is difficult because of the complexity involved. Selecting an effective attribute-choice method and managing training data that contain missing attribute values are also challenging. These problems and the complexity of the design are addressed in the presented method.

Service categorization characterizes many degrees of service-provisioning attributes. There are four service categories: Outstanding, Very Good, Average, and Poor. Classification is distinguished according to the overall quality assessment and average values of the selected QoS parameters. The suggested algorithm classifies and selects the most important amenities within the tree structure. First, a classifier is trained and tested. The resulting decision tree is then used to categorize previously unseen data. During attribute selection, the algorithm concentrates only on features relevant to the application.

$$B_{DU} = \frac{1}{l}\sum_{j=1}^{l} B_j \tag{12}$$

B_{DU} is the cross-validation accuracy, l is the number of cross-validation folds, and B_j is the accuracy for the jth fold.

$$P_{\text{normalize}} = \frac{P_j - P_n}{P_m - P_n} \tag{13}$$

Here, P_j is the normalized QoS value, while P_n and P_m are the minimum and greatest perceived QoS values in the dataset.

$$H(B) = 1 - \sum_{j=1}^{n} q_j^2 \tag{14}$$

B is the quality being assessed, n is the total number of classes, and q_j^2 is the proportion of samples in class j. Splits are better when the $H(B)$ value is lower.

$$Q = \alpha E + \beta P \tag{15}$$

Using the overfitting measure (P) and tree depth (E), this expression penalizes inappropriate training. The parameters α and β are criterion weights.

$$G_o = \arg\min_{G} L(G) \tag{16}$$

The loss function $L(G)$ depends on the hyperparameter G.

$$P_a = \sum_{j=1}^{n} z_j P_j \tag{17}$$

$$\sum_{j=1}^{m} z_j y_j + c = 0 \tag{18}$$

The feature y_j and its allocated weight are denoted by z_j, while the bias term is denoted by c. The term P_j is QoS measure j (for instance, response time or availability). There are m QoS measures, and each metric has weight z_j.

6. RESULTS

Evaluation of the suggested approach is carried out using the given metrics and parameters. This section discusses the QoS characteristics and measures used in the proposed system.

6.1 Response Time

The amount of time taken by the web service to reply to a request is used to evaluate the service. Web-service users demand lower response latency in milliseconds.

6.2 Reliability

Reliability is a web service's capacity to carry out its specified operations within a certain time frame and set of parameters.

6.3 Performance

Performance is a multidimensional statistic that measures how well the system classifies online services, adapts to new situations, and efficiently uses its resources.

6.4 Availability

Availability indicates that the network is ready when the service is called. To keep customers satisfied, service providers should deliver their online services with a high availability ratio.

Table 2. Investigative DL models in connection to suggested approaches.

Services	Response Time (ms)
Service 1	106.00
Service 2	321.50
Service 3	783.81
Service 4	524.11
Service 5	538.50
Service 6	249.00
Service 7	78.00

The seven online services' response times show how different they are in terms of efficiency. With a response time of only 78.00 ms, Service 7 tops the chart for quickness and effectiveness. In contrast, Service 3 has the longest response time, 783.81 ms, making it the least ideal for uses requiring instantaneous replies.

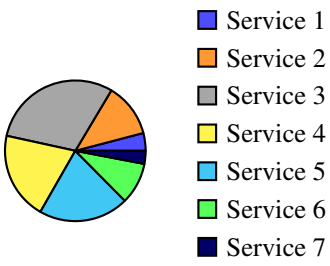

Figure 4. Assessment of ML models in relation to conventional methods.

Services 1 and 6 have comparatively low response times of 106.00 and 249.00 ms, indicating that they are quicker than most other services.

Services 2, 4, and 5 are in the middle, with response times of 321.50, 524.11, and 538.50 ms, re-spectively. Service 7 is therefore the best option for jobs thatneed a quick response, while Services 3, 4, and 5 may be adequate when a fast response is not strictly necessary.

Table 3. Comparison of the proposed technique with current methods.

Services	Reliability (%)
Service 1	63
Service 2	61
Service 3	89
Service 4	76
Service 5	67
Service 6	73
Service 7	92

Service 7's dependability of 92% makes it the most trustworthy alternative for crucial jobs, although the reliability of the seven online services displays a broad range. Service 3 follows with 89%, demonstrating proficiency in consistently delivering service. The moderate reliability values of 76% for Service 4 and 73% for Service 6 indicate that they are adequate for many applications. Services 5 and 1, with 67% and 63% reliability, remain usable for non-critical activities. Service 2 has the lowest dependability at 61%, suggesting a greater probability of breakdowns or uneven performance. Overall, Service 7 is clearly the more reliable option.

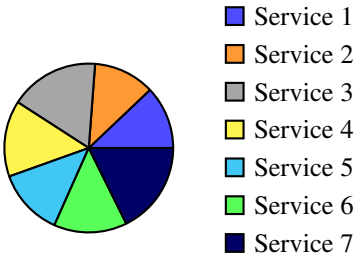

Figure 5. Comparison of ML models with conventional methods for evaluation.

Table 4. Parameters of algorithms sourced from nature.

Services	Availability (%)
Service 1	82
Service 2	96
Service 3	94
Service 4	88
Service 5	74
Service 6	71
Service 7	99

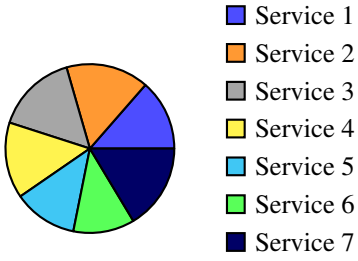

Figure 6. Effectiveness of different systems.

With an availability rate of 99%, Service 7 is the most trustworthy of the seven online services and is frequently available. Services 2 and 3 follow with availability ratings of 96% and 94%, respectively. Applications that require constant service access may also use Service 4, which has 88% availability. Service 1 has moderate availability at 82%, while Services 5 and 6 have the lowest values at 74% and 71%, respectively. Given these lower results, they are less reliable for mission-critical applications requiring high uptime. Services 5 and 6 may require substantial improvement to meet higher availability requirements, while Service 7 is clearly the most user-friendly.

Table 5. Exploratory DL models in relation to proposed methods.

Services	Performance (%)
Service 1	57
Service 2	79
Service 3	81
Service 4	69
Service 5	76
Service 6	85
Service 7	98

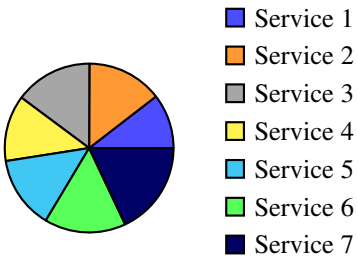

Figure 7. Effectiveness of different models.

After comparing the seven online services, Service 7 was the most efficient and reliable, earning a performance score of 98%. Services 3 and 6 also demonstrate significant capabilities well suited to complex applications, with scores of 81% and 85%, respectively. With ratings of 79% and 76%, Services 2 and 5 are dependable general-purpose options. Service 4 performs around average at 69%, while Service 1 comes last at 57%, suggesting that it may not be the best choice for jobs requiring substantial power. Services 6 and 3 achieved outstanding performance, but Service 7 was the highest-performing service overall.

7. CONCLUSION

Using decision-tree classifiers, in particular the C5.0 algorithm, this paper presents a complete method for categorizing online services according to their QoS characteristics. The proposed technique improves the effectiveness of service search and selection by classifying internet services into categories such as Excellent, Good, Average, and Poor. The reliability of the decision-tree structure can be increased while reducing its level of detail by including error-pruning methods and genetic algorithms for feature selection. Furthermore, cross-validation guarantees the dependability and generality of the framework across a wide variety of datasets.

Despite these developments, a number of issues and limitations remain, especially with regard to managing dynamic and adaptable QoS indicators. For real-time service selection in massive networks, the computational complexity of ranking algorithms and classifiers such as C5.0 can become a bottleneck. The scalability of the model, particularly for managing large amounts of data in real-time settings, should be the primary emphasis of future study. To ensure that the system remains resilient and adaptive, more research is required to modify categorization models so that they account for rapidly changing QoS features. Incorporating machine-learning methodologies, including deep learning or reinforcement learning, may result in more efficient and adaptable categorization systems for software applications. For optimum judgments, decision makers should also consider unique contextual factors; future research may apply multi-criteria decision-making procedures to assess new product-marketing strategies.

REFERENCES

[1] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 355–370, Feb. 2017.

[2] E. Mohamed, "The relationship between artificial intelligence and internet of things: A quick review," *Journal of Cybersecurity and Information Management*, vol. 1, no. 1, pp. 30–34, 2020.

[3] A. B. Gadicha and V. B. Gadicha, "Implicit authentication approach by generating strong password through visual key cryptography," *Journal of Cybersecurity and Information Management*, vol. 1, no. 1, pp. 5–16, 2020.

[4] B. S. Balaji, S. Balakrishnan, K. Venkatachalam, and V. Jeyakrishnan, "Retracted article: Automated query classification based web service similarity technique using machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 6169–6180, Jun. 2021.

[5] M. S. Das, A. Govardhan, and D. V. Lakshmi, "Classification of web services using data mining algorithms and improved learning model," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 6, pp. 3191–3198, Dec. 2019.

[6] M.-T. Wu, "Confusion matrix and minimum cross-entropy metrics based motion recognition system in

the classroom," *Scientific Reports*, vol. 12, no. 1, pp. 1–10, Feb. 2022.

[7] C. Vivek, M. Indu, and N. Nandhini, "Speech recognition using artificial neural network," *Journal of Cognitive Human-Computer Interaction*, vol. 5, no. 2, pp. 8–14, 2023.

[8] V. Narayanan, P. Nithya, and M. Sathya, "Effective lung cancer detection using deep learning network," *Journal of Cognitive Human-Computer Interaction*, vol. 5, no. 2, pp. 15–23, 2023.

[9] S. Ruuska, W. Hämäläinen, S. Kajava, M. Mughal, P. Matilainen, and J. Mononen, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behavioural Processes*, vol. 148, pp. 56–62, Mar. 2018.

[10] N. Agarwal, G. Sikka, and L. K. Awasthi, "Enhancing web service clustering using length feature weight method for service description document vector space representation," *Expert Systems with Applications*, vol. 161, p. 113682, Dec. 2020.

[11] M. B. Blake, W. Cheung, M. C. Jaeger, and A. Wombacher, "WSC-06: The web service challenge," in *Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, Jun. 2006, p. 62.

[12] D. A. Pustokhin and I. V. Pustokhina, "FLC-NET: Federated lightweight network for early discovery of malware in resource-constrained IoT," *International Journal of Wireless and Ad Hoc Communication*, vol. 6, no. 2, pp. 43–55, 2023.

[13] M. Saber, E.-S. M. El-Kenawy, A. Ibrahim, M. M. Eid, and A. A. Abdelhamid, "Metaheuristic optimized ensemble model for classification of SMS spam in computer networks," *International Journal of Wireless and Ad Hoc Communication*, vol. 6, no. 2, pp. 56–64, 2023.

[14] A. V. Dastjerdi and R. Buyya, "Compatibility-aware cloud service composition under fuzzy preferences of users," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 1–13, Jan. 2014.

[15] M. Razian, M. Fathian, H. Wu, A. Akbari, and R. Buyya, "SAIoT: Scalable anomaly-aware services composition in CloudIoT environments," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3665–3677, Mar. 2020.

[16] M. Gao, M. Chen, A. Liu, W. H. Ip, and K. L. Yung, "Optimization of microservice composition based on artificial immune algorithm considering fuzziness and user preference," *IEEE Access*, vol. 8, pp. 26 385–26 404, 2020.

[17] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, and M. A. Netto, "A manifesto for future generation cloud computing: Research directions for the next decade," *ACM Computing Surveys*, vol. 51, no. 5, pp. 1–38, 2018.

[18] H. S. Salih and F. A. Mohamed, "Fusion-based diversified model for internet of vehicles: Leveraging artificial intelligence in cloud computing," *Fusion: Practice and Applications*, vol. 12, no. 2, pp. 54–69, 2023.

[19] Z. N. Al-kateeb and D. B. Abdullah, "A smart architecture leveraging fog computing fusion and ensemble learning for prediction of gestational diabetes," *Fusion: Practice and Applications*, vol. 12, no. 2, pp. 70–87, 2023.

[20] G. Chen, T. Jiang, M. Wang, X. Tang, and W. Ji, "Modeling and reasoning of IoT architecture in semantic ontology dimension," *Computer Communications*, vol. 153, pp. 580–594, Mar. 2020.

[21] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Network*, vol. 33, no. 2, pp. 36–43, mar/apr 2019.

[22] S. K. Gavvala, C. Jatoth, G. R. Gangadharan, and R. Buyya, "QoS-aware cloud service composition using eagle strategy," *Future Generation Computer Systems*, vol. 90, pp. 273–290, Jan. 2019.

[23] M. Vucetic, M. Hudec, and B. Bozilovic, "Fuzzy functional dependencies and linguistic interpretations employed in knowledge discovery tasks from relational databases," *Engineering Applications of Artificial Intelligence*, vol. 88, p. 103395, Feb. 2020.

[24] M. Sozat and A. Yazici, "A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations," *Fuzzy Sets and Systems*, vol. 117, no. 2, pp. 161–181, Jan. 2001.

[25] B. Sheng, O. M. Moosman, J. Del Pozo-Cruz, B. 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, p. 107480, Mar. 2020.