A Probabilistic Trust Model for Cloud Services Using Bayesian Networks

Mihan Hosseinnezhad
Islamic Azad University of Miyaneh: Islamic Azad University Miyaneh Branch
Mohammad Abdollahi Azgomi (azgomi@gmail.com)
Iran University of Science and Technology https://orcid.org/0000-0002-9605-8412
Mohammad Reza Ebrahimi Dishabi
Islamic Azad University Miyaneh Branch

Research Article

Keywords: Cloud Computing, Cloud Services, Trust Models, Bayesian Networks.

Posted Date: December 21st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-281906/v1

Version of Record: A version of this preprint was published at Soft Computing on May 3rd, 2023. See the published version at https://doi.org/10.1007/s00500-023-08264-z.

Abstract

With the rapid adoption of cloud computing in the industry, there has been a significant challenge in managing trust between cloud service providers and service consumers. In fact, trust management in cloud computing has become very challenging given the urgent need for cloud service requesters to choose efficient, trustworthy and non-risky services. One of the most important factors that can be considered in the trust or distrust of a service by the applicant is the different quality of services related to the service. Therefore, approaches are needed to assess the trustworthiness of cloud services with respect to the values of their Quality of Service (QoS). Given the uncertainty that exists for cloud services, it is more realistic to model their QoS parameters as random variables and also consider different dependencies between them. In this paper, a new trust model for cloud services is proposed using Bayesian networks. Bayesian network is a probabilistic graphical model that can be used as one of the best methods to control uncertainty. Using Bayesian network makes it possible to infer more accurate QoS values will which leads to the selection of highly trustworthy services by several cloud service requesters. The results of the experiments show that the proposed trust model is highly accurate and significantly reduces the estimation error.

1. Introduction

Cloud computing presents a new model for IT services. In these models, scalable and virtual resources are provided over the Internet. In such a system, users try to access the service based on their needs, no matter where the service is located or how it is delivered [1]. Cloud computing has a resemblance to a cloud mass through which users can access applications anywhere in the world, so in cloud computing, cloud is a set of distributed nodes that provide computing resources based on demand or user services over the network. The cloud computing approach is relatively new and, in many cases, not yet accepted. Corporate IT departments are still wary of it because the cloud computing platform will not be controlled by that company. Like all emerging approaches, there is a degree of fear, uncertainty and concerns about the development of this technology [2]. One of the problems with cloud computing is the cost of bandwidth. While companies can save on hardware and software with the help of cloud computing, they will have to incur higher bandwidth charges. Also, the relative security of cloud computing is a controversial issue that may delay cloud computing acceptance. In addition to data security, the availability and performance of cloud-hosted applications is crucial for users [3].

However, despite the benefits and rapid growth of cloud computing, it creates security, privacy and trust issues that require immediate action. Trust is an important concept and challenge for cloud computing, as cloud service users are urgently needed to select affordable, reliable and less risky services [4]. The issue of trust is also important for service providers to decide on an infrastructure provider that may suit their needs and to check whether infrastructure providers maintain their consent when using their services. In fact, an effective trust management system helps cloud service providers as well as consumers to reap the benefits of cloud computing technology. Despite the various benefits of trust management in cloud computing, several issues related to public trust assessment mechanisms,

inaccurate feedback, poor feedback detection, participant privacy, and feedback inconsistency still need to be addressed. Traditional trust management approaches such as using a service level agreement for complex cloud environments are inadequate. The vague provisions and unspecified technical specifications of service level agreements can make cloud service customers unable to identify reliable cloud services. Also, given the increasing number of cloud service providers and hence the number of cloud services, it can be concluded that several services will be provided by different cloud providers that are quite similar in performance, but different in the QoS parameters [5]–[9]. Service quality is a set of non-functional properties that reflects the quality offered by a service. In a service, service quality requirements essentially refer to the non-functional quality of service [10].

Traditional service quality includes a wide range of definitions such as response time, availability and reliability. However, in a cloud computing environment, service quality needs to be objectively linked to users' mental understanding. Therefore, from a service consumer perspective, some researchers have concluded that the relationship of trust between cloud service users and cloud service providers is itself an important and essential quality criterion. Given the trust relationship, cloud service users can easily identify the trusted providers they must interact with, as well as the unreliable providers they must engage with. Trust, in a distributed network environment, is considered an essential secure relationship. In general, trust can, as a result of observations, lead to the subjective belief that other measures may be used to achieve an objective in hazardous situations. Trust is updated over time through direct interactions or information provided by others about their experiences. Many researchers have investigated the issue of trust in cloud computing and cloud services. But many of them have not mentioned some of the basic features of trust, such as mental uncertainty. In addition, the various approaches proposed to calculate trust have not addressed the dynamic nature of trust. The dynamic nature of trust actually refers to changes in the level of trust. Trust values calculated in the past usually decrease over time due to various QoS changes and are less effective than trust decisions in the present. Therefore, decreasing trust over time may be an important factor that should be taken into account in approaches to evaluate trust.

Another issue with the cloud service quality parameters is that some of the cloud service quality parameters may not be specified due to some reason. The various trust models provided for cloud services in such circumstances have difficulty in assessing the trust for those cloud services and cannot obtain the exact and actual values of trust for them. It also reveals various models of cloud computing trust, most of which consider QoS parameters values as definite values, while considering uncertain states in Web services and especially cloud services, it is more realistic to model QoS parameters as random variables, and in fact calculate the amount of trust in a probabilistic way. However, there are also methods that model the quality of service using contingency programming and random programming [9], [11]. Although these methods have modeled service quality using random variables, they have a common disadvantage, namely that they do not take into account the dependency between the QoS parameters. In fact, there may be dependencies between the QoS parameters of the service that by taking these dependencies into account the amount of trust can be more accurately calculated and brought closer to reality. For example, a small response time results in a low cost.

According to the items mentioned in the previous section, all methods and models presented to calculate the trust related to cloud services have shortcomings that the innovations of this paper are in order to eliminate these shortcomings. One of these shortcomings is the difficulty in calculating trust for cloud service for which the values of some service quality parameters are not known. Previous approaches in such situations due to the lack of values of some service quality parameters and also the dependence of the process of calculating the amount of trust in the values of these parameters, face problems. Another shortcoming that can be seen in the proposed models is that they do not take into account the dependencies that can exist between service quality parameters, in calculating the values of these parameters and finally in calculating the level of trust. However, the existing dependencies between service quality parameters affect the trust assessment process. Due to the shortcomings mentioned in the models and approaches of trust assessment for cloud services, in this paper, a new trust model based on Bayesian networks will be used to solve the problems expressed in the assessment of trust for cloud services.

2. Related Works

The authors in [12] propose a dynamic evidence-based trust model in which the reliability of services is calculated in the cloud. In this generalized system, fuzzy inference system and IOWA operator are integrated in order to obtain the value of dynamic trust. Given the flexibility of the system for any current and future new services, it can be said that the proposed model is fully compatible with dynamic environments. The critical role of error detection and compatibility across different domains implies that trust assessment systems must be consistent. Such a system helps cloud service providers to enhance the performance of the services provided. The simulation results show the performance of the proposed model. Although it is flexible, it suffers from high costs and delays.

Researchers have also developed a new model of limited reliability to mitigate malware threats and internal services in the cloud environment, which reduces the implications of streaming networks for reducing the scale of malicious software or services. The proposed model can be used in the following two ways: (1) running a trust service among guest services, as well as assessing threats from anonymous malware; (2) reducing the risk associated with leased services. Cloud environments and reduced resource depletion affected by malicious guest services. Although this model can effectively limit the scale of malicious services and significantly reduce the risk of internal attacks, it is not scalable and suffers from low availability [13].

As another study [14], a trust management framework for service-based systems has been proposed that is adaptable. The proposed framework consists of a meta-model with an official language that is appropriate for the security policies of the situation and the devices to deploy and create agents for evaluating and making trust decisions based on security laws, situational information and credentials. Although this framework improves scalability, accessibility and performance, it suffers from low performance and high time cost. Also, the authors in [15] have introduced a new trust model, depending on previous credentials and current capabilities of cloud providers. Some of the issues involved in

calculating the value of trust include reliability, integrity, accessibility and performance. The method combines the quality of customer service and cloud service capabilities provided by a service level agreement. This research increases reliability and accessibility, though not scalability.

In addition, in [16], the authors consider a generalized boundary method. The requirements of both data providers and data consumers can be met by this method. In this way, by extending data items, both usability and privacy can be achieved. In addition, in this paper, a privacy information access control model is presented in which a combination of trust-based decision-making policy and access control policy creates a security protection system. In addition, the proposed model offers effective generalization methods for privacy access control systems. Although this research can improve accessibility and reliability, it is not scalable and also has a high cost.

In addition, researchers in [17] have developed a prototype that logically provides a solution for evaluating and managing the trust and credibility of web services. A suite of service requestor feedback, trust management, and trust assessments are integrated with a proposed prototype that helps provide an effective way to select reliable service for applicants. In order to model the service trust as accurately as possible, a mathematical expression for the different types of data to describe service trust including probabilistic values, small values, fuzzy values and discrete values is presented in this research work. This research improves the success rate of implementing reliable and scalable services, although it is costly.

In addition, the authors [18] have provided a framework for enhancing trust management practices in cloud computing. In fact, they have presented a credit model that not only can detect malicious trust credentials from attackers, but also distinguishes credible trust feedbacks. In addition, they have developed an alternate assignment model that is able to dynamically detect the number of alternatives to a trust management service. Although this research increases accessibility, it suffers from high costs and low scalability.

In [19], Chiraghi et al. have presented an approach that evaluates the validity of cloud services and identifies trusted services in cloud environments. Validity is assessed using three parameters including accessibility, reliability and capability. In this research work, a method for trusted services using three topological indices including input, output, and validity is proposed. The proposed method in this paper has been evaluated under various challenging conditions and the results show that the accuracy of the proposed method increases with the recommendation of trusted cloud providers. In [20], Chong et al. proposed a multi-sided trust management system architecture for the cloud computing market to support customers in identifying trusted providers. This article presents important threats to a trust management system as well as ways to deal with these threats. This article defines the prominent features of a trust management system. In this research work, security components are also used to identify the trust and credibility of e-commerce participants to assist online customers in deciding whether or not to make a transaction. Based on the framework mentioned, this paper also proposes an approach to filter malicious feedback and a trust metric to assess the level of trust and trustworthiness of cloud service providers.

The results of various simulation experiments show that the proposed multilateral trust management system can be very effective in identifying risky transactions in electronic markets.

In [21], a trust assessment model based on D-S evidence theory and slider windows for cloud computing is presented. The timeliness of cross-evidence as first-hand evidence is reflected by the introduction of slider windows. The direct trust of entities is calculated based on the cross-evidence of D-S evidence theory. The computation of trusts is based on D-S theory with the help of cross-circuits. The value proposition of trust from various institutions has been considered as second-hand evidence. The combination of recommended trust values constitutes the validity of the territory. Finally, experiments were conducted to evaluate the effectiveness and counter-attack of the proposed model. The following are the advantages and disadvantages of the trust models presented for the cloud computing environment.

3. Proposed Trust Model

The various trust models proposed for the cloud computing environment use deterministic values for the QoS parameters related to the cloud services. However, web services, especially cloud services, are in an uncertain state, so it is better to model QoS parameters as random variables. Different QoS parameters related to a cloud service can be interdependent and this dependency can have a significant effect on calculating the values of QoS parameters. For example, the reliability and accessibility parameters of a cloud service depend on the cost parameter of that service. Given the above, we propose to use a probabilistic model to calculate the more accurate values of QoS parameters as well as accurate value of trust. Therefore, we intend to use Bayesian network to obtain the dependencies between the QoS parameters and also to calculate their values to predict the accurate trust value.

Bayesian network is one of the famous mathematical models for controlling uncertainty in various problems which is based on probability theory. Bayesian network is a graphical model that finds possible dependencies between system variables. The question that arises here is why we have chosen Bayesian networks to solve the mentioned problems, despite the different methods of controlling the uncertainty in artificial intelligence. To answer this question, we can mention two important features of Bayesian networks. First, Bayesian networks examine incomplete datasets without the slightest problem, as they are able to detect dependencies between variables. When one of the inputs is not seen, most models end up with an incorrect estimate. This is because they do not calculate the dependencies between the input variables. Bayesian networks suggest a natural way to encode these dependencies. Second, anyone can learn about causal relationship using Bayesian networks. There are two important reasons for learning about causal relationship. This process is valuable when we want to understand the scope of the problem, for example when exploratory data is analyzed or when an agent explores the environment. In addition, if there is intervention, one can make estimates using the knowledge of causal relationship.

Proposed probabilistic trust model consists of several parts, which are as follows:

- Discretization of continuous QoS values in the data set

- Creating the structure or topology of Bayesian network using the structure learning algorithm

- Calculation of conditional probability distributions for each node in the network using parameter learning algorithm

- Estimating the trust new values when the value of some QoS parameters changes

Figure 1 demonstrates different parts of the proposed probabilistic trust model. In the following, these various parts will be presented in full detail.

3.1. Data set

Structure and parameter learning algorithms in Bayesian networks require a data set to perform the learning operation. Considering that in the proposed model, the created Bayesian network should indicate possible dependencies between QoS parameters related to cloud services, the data set used to learn the structure and parameters of the desired Bayesian network should also contain QoS values of different cloud services. Tala et. al have conducted a comprehensive investigation of cloud services available on the Web [22]. They have developed a cloud services crawler engine that collects, validates,

and categorizes cloud services, and produced a number of datasets that store the information of the collected cloud services. The collected datasets include meta-data of nearly 10080 real-world cloud services. The QoS parameters of cloud services contained in this data set are: *Availability, Security, Response_Time, Accessibility, Speed, Storage_Space, Features, End_of_Use, Technical_Support, Customer_Service, Level_of_Expertise* and *Trust.* We have used this dataset to learn the structure and parameters of the Bayesian network in the proposed trust model.

3.2. Discretization

Several approaches have been proposed in the literature to learn the structure and parameters of Bayesian networks from data. These approaches operate on the assumption that all values of the variables in the domain are discrete or continuous and follow a normal or Gaussian distribution. In fact, existing learning algorithms for Bayesian networks are not able to work with continuous data with abnormal distribution. One of the best solutions to this problem is to divide the continuous values into several bins. The process of dividing continuous values into different bins is called discretization. Examining the dataset reveals that all quality parameters discussed in the previous section have continuous values. Given that Bayesian network learning algorithms work only with discrete data, the dataset can be used to learn Bayesian networks if the values of the parameters in it are discrete.

In the proposed trust model, discretization is performed in two different stages: before structure learning of the Bayesian network and before parameters learning of the Bayesian network. In the proposed model for learning Bayesian network structure from the desired dataset, WEKA machine learning software is

used. The WEKA desktop includes a comprehensive set of machine learning algorithms and data processing tools [23]. Due to the use of WEKA for structure learning, the WEKA discretization algorithm has been used to discretize the values of parameters in the desired dataset. WEKA discretization algorithm uses the Fayyad and Irani MDL method to discretize continuous variables [24]. This discretization algorithm tries to place the same number of values in each bin or interval. If there is an attribute value of n points in the whole range and we divide it into k distances, then each bin will have n / k points. This type of discretization is named equal frequency discretization algorithm.

3.3. Structure Learning

Learning the structure of Bayesian networks from data is one of the most challenging problems. Several approaches have been proposed in the literature to learn the structure of Bayesian networks from data. In the proposed trust model, the *K2* algorithm is used to learn the Bayesian network structure. The *K2* algorithm is the one that has been used more than any other approach for structure learning in Bayesian networks [25]. *K2* is a score-based structure learning algorithm that takes a greedy approach to learn network structure from data. *K2* is trying to find a network structure that maximizes the likelihood of delay by having a test dataset. To obtain the Bayesian network structure in the proposed trust model, WEKA algorithm *K2* is applied to the desired dataset. For this purpose, the *K2* algorithm is first quantified with a random initial order for the nodes and the resulting structure of which is shown in Fig. 2.

3.4. Parameter Learning

The next step in modeling a problem using Bayesian network after creating the network structure is to learn its parameters. The learning of the parameters of a Bayesian network determines the conditional probability distributions for each of its nodes. In the proposed trust model, Maximum-likelihood estimation (MLE) algorithm is used to learn the parameters of the Bayesian network [26]. MLE is a selected technique of estimating the parameters of a statistical model given data. MLE selects the set of values of the model parameters that maximizes the probability feature. This is a way of finding out the set of values of model parameters for which discovered data satisfactory” fit” the model, inside the feel that the likelihood of the empirical data is maximum. Conditional probability tables of *Response_Time* and *Price* are shown in Table 1 and Table 2.

Table 1
Conditional probability table of *Response_Time* node

Speed	Features	(-Inf-2.5]	(2.5–3.75]	(3.75–4.25]	(4.25–4.75]	(4.75-Inf]
(-Inf-0.05]	(-Inf-1.5]	0.647059	0.176471	0.058824	0.058824	0.058824
(-Inf-0.05]	(1.5–2.5]	0.2	0.2	0.2	0.2	0.2
(-Inf-0.05]	(2.5–3.5]	0.2	0.2	0.2	0.2	0.2
(-Inf-0.05]	(3.5–4.5]	0.2	0.2	0.2	0.2	0.2
(-Inf-0.05]	(4.5-Inf)	0.013953	0.004651	0.004651	0.004651	0.972093
(0.05–1.9]	(-Inf-1.5]	0.771429	0.142857	0.028571	0.028571	0.028571
(0.05–1.9]	(1.5–2.5]	0.62963	0.185185	0.111111	0.037037	0.037037
(0.05–1.9]	(2.5–3.5]	0.411765	0.176471	0.058824	0.058824	0.294118
(0.05–1.9]	(3.5–4.5]	0.538462	0.076923	0.230769	0.076923	0.076923
(0.05–1.9]	(4.5-Inf)	0.007092	0.007092	0.007092	0.007092	0.971631
(1.9–3.1]	(-Inf-1.5]	0.272727	0.090909	0.454545	0.090909	0.090909
(1.9–3.1]	(1.5–2.5]	0.463415	0.317073	0.170732	0.02439	0.02439
(1.9–3.1]	(2.5–3.5]	0.306667	0.413333	0.2	0.013333	0.066667
(1.9–3.1]	(3.5–4.5]	0.085714	0.2	0.542857	0.028571	0.142857
(1.9–3.1]	(4.5-Inf)	0.010989	0.032967	0.010989	0.010989	0.934066
(3.1–3.95]	(-Inf-1.5]	0.2	0.2	0.2	0.2	0.2
(3.1–3.95]	(1.5–2.5]	0.2	0.2	0.2	0.2	0.2
(3.1–3.95]	(2.5–3.5]	0.2	0.2	0.2	0.2	0.2
(3.1–3.95]	(3.5–4.5]	0.2	0.2	0.2	0.2	0.2
(3.1–3.95]	(4.5-Inf)	0.016393	0.016393	0.016393	0.016393	0.934426
(3.95–4.05]	(-Inf-1.5]	0.142857	0.142857	0.428571	0.142857	0.142857
(3.95–4.05]	(1.5–2.5]	0.294118	0.411765	0.176471	0.058824	0.058824
(3.95–4.05]	(2.5–3.5]	0.022222	0.111111	0.688889	0.022222	0.155556
(3.95–4.05]	(3.5–4.5]	0.008547	0.042735	0.692308	0.008547	0.247863
(3.95–4.05]	(4.5-Inf)	0.005348	0.005348	0.294118	0.005348	0.68984
(4.05–4.95]	(-Inf-1.5]	0.2	0.2	0.2	0.2	0.2
(4.05–4.95]	(1.5–2.5]	0.2	0.2	0.2	0.2	0.2

Speed	Features	(-Inf-2.5]	(2.5–3.75]	(3.75–4.25]	(4.25–4.75]	(4.75-Inf]
(4.05–4.95]	(2.5–3.5]	0.2	0.2	0.2	0.2	0.2
(4.05–4.95]	(3.5–4.5]	0.2	0.2	0.2	0.2	0.2
(4.05–4.95]	(4.5-Inf)	0.007519	0.007519	0.007519	0.007519	0.969925
(4.95-Inf)	(-Inf-1.5]	0.013699	0.013699	0.013699	0.013699	0.945205
(4.95-Inf)	(1.5–2.5]	0.015873	0.079365	0.015873	0.015873	0.873016
(4.95-Inf)	(2.5–3.5]	0.021898	0.007299	0.007299	0.007299	0.956204
(4.95-Inf)	(3.5–4.5]	0.012048	0.004016	0.108434	0.004016	0.871486
(4.95-Inf)	(4.5-Inf)	4.93E-04	0.002686	0.006085	0.003015	0.98772

Table 2
Conditional probability table of *Price* node

Speed	(-Inf-1.05]	(1.05–3.05]	(3.05–4.05]	(4.05–4.95]	(4.95-Inf)
(-Inf-0.05]	0.515419	0.189427	0.101322	0.110132	0.0837
(0.05–1.9]	0.323944	0.333333	0.201878	0.107981	0.032864
(1.9–3.1]	0.133047	0.484979	0.201717	0.064378	0.11588
(3.1–3.95]	0.016393	0.180328	0.377049	0.344262	0.081967
(3.95–4.05]	0.025496	0.150142	0.461756	0.048159	0.314448
(4.05–4.95]	0.052632	0.067669	0.142857	0.56391	0.172932
(4.95-Inf)	0.100998	0.131836	0.15243	0.007096	0.60764

3.5. Inference Algorithm

After the learning process, the learned Bayesian network can be used to estimate unseen data. The estimation step in Bayesian networks is usually called inference. Inference in Bayesian networks deals with the recognition of different probabilities of interest from the model [27]. In other words, the inference algorithm calculates the values of other domain variables by having observations for a subset of variables.

Junction tree algorithm, developed by Lauritzen and Spiegelhalter, is one of the most popular algorithms for exact inference in Bayesian networks [28]. It is based on a deep analysis of the connection between graph theory and probability theory. Instead of working with the original DAG, the junction tree algorithm

uses an auxiliary data structure, called junction tree. Considering the aforementioned discussion, junction tree algorithm used for inference in the proposed trust model.

4. Experimental Results

The Bayesian Network Toolbox (BNT) for MATLAB is used to evaluate the proposed probabilistic trust model which uses the Bayesian network to estimate the trust of cloud services. The BNT Toolbox is a Matlab open-source package for directed graph-based models which is widely used in teaching and research. BNT supports a variety of nodes (probability distributions), accurate and approximate inference, static and dynamic learning of parameters and structures, and models. The computer used to simulate the proposed trust model and perform various experiments has a Core i5 2.0 GHz CPU and 4 GB of RAM.

To evaluate the proposed trust model and in fact the constructed Bayesian network, holdout and K-fold cross-validation methods are used and compared. In the holdout method, the main dataset is divided into two datasets called the training dataset and the test dataset and the Bayesian network is constructed using training dataset and then evaluated by test dataset. The most important advantage of this method is the simplicity and high speed of the evaluation operation. How the data set is divided depends on the analyst's discretion, and in the experiments of the proposed trust model, 70% of the dataset is considered as training dataset and 30% as test dataset.

K-fold cross-validation is a model evaluation method that determines how generalizable and independent the results of a statistical analysis on a data set are from educational data. This method is especially used in forecasting applications to determine how useful the model will be in practice. In general, a round of cross-validation involves separating data into two complementary subsets, performing analysis on one of those subsets (training data), and validating the analysis using data from the other set (validation or test data). To reduce the scatter, the validation operation is performed several times with different divisions and the results of the validations are averaged. In K-fold cross-validation, the data is split into a *K* subset. Of these *K* subsets, one is used at a time for validation and the other *K-1* for training. This procedure is repeated *K* times and all data is used exactly once for training and once for validation. Finally, the average result of this *K* validation load is chosen as a final estimate. The 5-fold or 10-fold cross-validation method is typically used in modeling and forecasting research and in our experiments 10-fold cross-validation is used.

An important goal of inference algorithms in Bayesian networks is the high accuracy of inferred values. Therefore, the purpose of this experiment is to determine how the proposed trust model acts in estimating the QoS parameters values as well as trust values. Several test scenarios have been considered for this experiment, which are described below. Given that the number of bins for discretization phase is considered 10 and 20, as well as using of holdout and K-fold cross-validation techniques, four different test scenarios are considered. These test scenarios are named as follows: *TRUST_CV_10BIN*, *TRUST_HO_10BIN*, *TRUST_CV_20BIN* and *TRUST_HO_20BIN*. *CV* and *HO* refer to holdout and cross-validation techniques respectively.

In each test scenario, 20 cloud services are randomly selected from the test dataset, and it is assumed that the trust value of those services is requested. Therefore, the proposed trust model is used to estimate the trust values of cloud services in each scenario. After estimating the trust values, the estimated trust values are compared with the actual trust values. The results of experiments using the discussed test scenarios are shown in Table 3, Table 4, Table 5 and Table 6.

Table 3
Results of test scenario *TRUST_CV_20BIN*

Service ID	Estimated Trust Value	Actual Trust Value
1	4.7471	4.8
2	0.9566	1
3	2.0565	2
4	2.3545	2.4
5	3.6392	3.6
6	4.5596	4.6
7	1.3544	1.4
8	3.0535	3
9	4.967	5
10	2.7584	2.8
11	1.6423	1.6
12	2.5614	2.6
13	3.2303	3.2
14	3.954	4
15	1.9668	2
16	5.0446	5
17	0.9682	1
18	4.3417	4.4
19	1.0582	1
20	4.9424	5

Table 4
Results of test scenario *TRUST_HO_20BIN*

Service ID	Estimated Trust Value	Actual Trust Value
1	4.712	4.8
2	0.9647	1
3	2.0677	2
4	2.3637	2.4
5	3.6233	3.6
6	4.5274	4.6
7	1.3194	1.4
8	3.0551	3
9	4.9494	5
10	2.7644	2.8
11	1.6734	1.6
12	2.5242	2.6
13	3.2764	3.2
14	3.94	4
15	1.9188	2
16	5.04	5
17	0.9438	1
18	4.344	4.4
19	1.0676	1
20	4.9728	5

Table 5
Results of test scenario *TRUST_CV_10BIN*

Service ID	Estimated Trust Value	Actual Trust Value
1	4.7377	4.8
2	0.9288	1
3	2.0863	2
4	2.3585	2.4
5	3.6313	3.6
6	4.548	4.6
7	1.3423	1.4
8	3.0547	3
9	4.9558	5
10	2.7212	2.8
11	1.6737	1.6
12	2.53	2.6
13	3.2506	3.2
14	3.9608	4
15	1.9577	2
16	5.0403	5
17	0.9718	1
18	4.3001	4.4
19	1.0506	1
20	4.9177	5

Table 6
Results of test scenario *TRUST_HO_10BIN*

Service ID	Estimated Trust Value	Actual Trust Value
1	4.7324	4.8
2	0.9466	1
3	2.0596	2
4	2.3208	2.4
5	3.664	3.6
6	4.556	4.6
7	1.3521	1.4
8	3.0729	3
9	4.9332	5
10	2.7058	2.8
11	1.6584	1.6
12	2.4997	2.6
13	3.3029	3.2
14	3.924	4
15	1.9193	2
16	5.0966	5
17	0.9708	1
18	4.3049	4.4
19	1.0394	1
20	4.9552	5

Three criteria of MSE, RMSE and MAE were used to evaluate the results obtained from the experiments. MSE measures the average of the squared difference between the predicted value and the actual value. RMSE is the measure of the distance between the actual values and the predicted value which is calculated using the square root of MSE. Also, MAE represents the difference between the actual and predicted values extracted by averaged the absolute difference over the dataset. Table 7 represents the evaluation results using MSE, RMSE and MAE criterions.

Table 7
Evaluation results using MSE, RMSE and MAE criterions

Test scenario	MSE	RMSE	MAE
TRUST_CV_20BIN	**0.001926**	**0.001926**	**0.043195**
TRUST_HO_20BIN	0.002922	0.002922	0.0506
TRUST_CV_10BIN	0.003668	0.003668	0.0554
TRUST_HO_10BIN	0.0046	0.0046	0.0638

After reviewing the evaluation results in Table 7, it can be concluded that discretization with 20 bins has had a significant effect on improving the quality of estimation as well as reducing different types of errors. Also, using10-fold cross-validation instead of holdout method could increase the accuracy of estimation and also reduce different types of errors.

5. Conclusions

Trustworthiness plays an important role in determining the quality of cloud services for designing efficient and flexible systems based on services. The uncertain nature of cloud computing as a service-oriented environment affect the performance of trust-based cloud service selection models. Predicting trust value of cloud services is a classification problem which can be modeled as a proper solution to the problem of choosing a cloud-based cloud service by predicting the reliability of cloud services based on information about their historical QoS. Therefore, several researchers focused on the development of multiple trust and QoS prediction models based on machine learning and statistical techniques. In this paper, a new trust model using Bayesian network for cloud service-oriented environment is proposed. Bayesian network is a probabilistic graphical model that can be used as one of the best methods to control uncertainty. Using Bayesian network makes it possible to find the dependencies between different service quality parameters and to infer more accurate and realistic values of QoS parameters. More accurate and realistic inference of QoS parameter values will allow the trust assessment to be more accurate with respect to QoS parameter values, leading to the selection of highly trustworthiness cloud services by different users and applications. Evaluation of experiments performed in the Matlab environment indicates that the proposed trust model has a high accuracy level and tries to estimate the trust level for cloud services with the least error.

Declarations

Funding

Not applicable

Conflicts of interest

The authors declare that they have no conflict of interest.

Availability of data and material

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Code availability

All code for data analysis associated with the current submission is available from the corresponding author upon reasonable request

Funding

Not applicable

Conflicts of interest

M. Hosseinnezhad declares that he has no conflict of interest.

M.R.E. Dishabi declares that he has no conflict of interest.

M.A. Azgomi declares that he has no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] P. Mell and T. Grance, “The NIST definition of cloud computing,” 2011.

[2] W. Voorsluys, J. Broberg, R. Buyya, and others, “Introduction to cloud computing,” *Cloud Comput. Princ. Paradig.*, pp. 1–44, 2011.

[3] T. Velte, A. Velte, and R. Elsenpeter, *Cloud computing, a practical approach*. McGraw-Hill, Inc., 2009.

[4] A. S. Horvath and R. Agrawal, “Trust in cloud computing,” in *SoutheastCon 2015*, 2015, pp. 1–8, doi: 10.1109/SECON.2015.7132885.

[5] F. Rosenberg, P. Celikovic, A. Michlmayr, P. Leitner, and S. Dustdar, “An end-to-end approach for QoS-aware service composition,” in *Proceedings of the 13th IEEE international conference on Enterprise Distributed Object Computing*, 2009, pp. 128–137.

[6] J.-Z. Luo, J.-Y. Zhou, and Z.-A. Wu, “An adaptive algorithm for QoS-aware service composition in grid environments,” *Serv. Oriented Comput. Appl.*, vol. 3, pp. 217–226, 2009, [Online]. Available: http://dx.doi.org/10.1007/s11761-009-0047-6.

[7] A. Liu, L. Huang, and Q. Li, “QoS-Aware Web Services Composition Using Transactional Composition Operator,” in *Advances in Web-Age Information Management*, vol. 4016, J. Yu, M. Kitsuregawa, and H. Leong, Eds. Springer Berlin / Heidelberg, 2006, pp. 217–228.

[8] F. Lécué, “Optimizing QoS-Aware Semantic Web Service Composition,” in *The Semantic Web, ISWC 2009*, vol. 5823, A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, Eds. Springer Berlin / Heidelberg, 2009, pp. 375–391.

[9] W. Wiesemann, R. Hochreiter, and D. Kuhn, “A Stochastic Programming Approach for QoS-Aware Service Composition,” in *Proceedings of the 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid*, 2008, pp. 226–233, doi: http://dx.doi.org/10.1109/CCGRID.2008.40.

[10] A. Mani and A. Nagarajan, “Understanding quality of service for Web services,” *IBM Dev.*, 2002.

[11] S.-Y. Hwang, H. Wang, J. Tang, and J. Srivastava, “A probabilistic approach to modeling and estimating the QoS of web-services-based workflows,” *Inf. Sci.*, vol. 177, no. 23, pp. 5484–5503, 2007, doi: http://dx.doi.org/10.1016/j.ins.2007.07.011.

[12] A. Selvaraj and S. Sundararajan, “Evidence-based trust evaluation system for cloud services using fuzzy logic,” *Int. J. Fuzzy Syst.*, vol. 19, no. 2, pp. 329–337, 2017.

[13] Y. Wang, S. Chandrasekhar, M. Singhal, and J. Ma, “A limited-trust capacity model for mitigating threats of internal malicious services in cloud computing,” *Cluster Comput.*, vol. 19, no. 2, pp. 647–662, 2016.

[14] S. S. Yau, Y. Yao, and A. B. Buduru, “An adaptable distributed trust management framework for large-scale secure service-based systems,” *Computing*, vol. 96, no. 10, pp. 925–949, 2014.

[15] P. D. Manuel, S. T. Selvi, and M. I. Abd-El Barr, “Trust management system for grid and cloud resources,” in *Advanced Computing, 2009. ICAC 2009. First International Conference on*, 2009, pp. 176–181.

[16] M. Li, S. Yu, N. Cao, and W. Lou, “Authorized private keyword search over encrypted data in cloud computing,” in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011, pp. 383–392.

[17] P. Zhang and Z. Yan, “A QoS-aware system for mobile cloud computing,” in *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, 2011, pp. 518–522.

[18] T. H. Noor, Q. Z. Sheng, and A. Bouguettaya, *Trust Management in Cloud Services*. Springer, 2014.

[19] M. Chiregi and N. J. Navimipour, “Trusted services identification in the cloud environment using the topological metrics,” *Karbala Int. J. Mod. Sci.*, vol. 2, no. 3, pp. 203–210, 2016, doi: http://dx.doi.org/10.1016/j.kijoms.2016.06.002.

[20] S.-K. Chong, J. Abawajy, M. Ahmad, and I. R. A. Hamid, “Enhancing Trust Management in Cloud Environment,” *Procedia - Soc. Behav. Sci.*, vol. 129, pp. 314–321, 2014, doi: http://dx.doi.org/10.1016/j.sbspro.2014.03.682.

[21] X. Wu, R. Zhang, B. Zeng, and S. Zhou, “A Trust Evaluation Model for Cloud Computing,” *Procedia Comput. Sci.*, vol. 17, pp. 1170–1177, 2013, doi: http://dx.doi.org/10.1016/j.procs.2013.05.149.

[22] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. H. Ngu, “CloudArmor: Supporting reputation-based trust management for cloud services,” *IEEE Trans. parallel Distrib. Syst.*, vol. 27, no. 2, pp. 367–380, 2015.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009, doi: 10.1145/1656274.1656278.

[24] U. M. Fayyad and K. B. Irani, “Multi-interval discretization of continuous-valued attributes for classification learning,” in *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, IJCAI ’93*, 1993, pp. 1022–1029.

[25] B. Lerner and R. Malka*, “Investigation of the K2 algorithm in learning Bayesian network classifiers,” *Appl. Artif. Intell.*, vol. 25, no. 1, pp. 74–96, 2011.

[26] I. J. Myung, “Tutorial on maximum likelihood estimation,” *J. Math. Psychol.*, vol. 47, no. 1, pp. 90–100, 2003.

[27] D. Heckerman, “A Tutorial on Learning with Bayesian Networks,” in *Innovations in Bayesian Networks*, vol. 156, D. Holmes and L. Jain, Eds. Springer Berlin / Heidelberg, 2008, pp. 33–82.

[28] S. L. Lauritzen and D. J. Spiegelhalter, “Local computations with probabilities on graphical structures and their application to expert systems,” *J. R. Stat. Soc. B*, vol. 50, pp. 157–224, 1988.

Figures

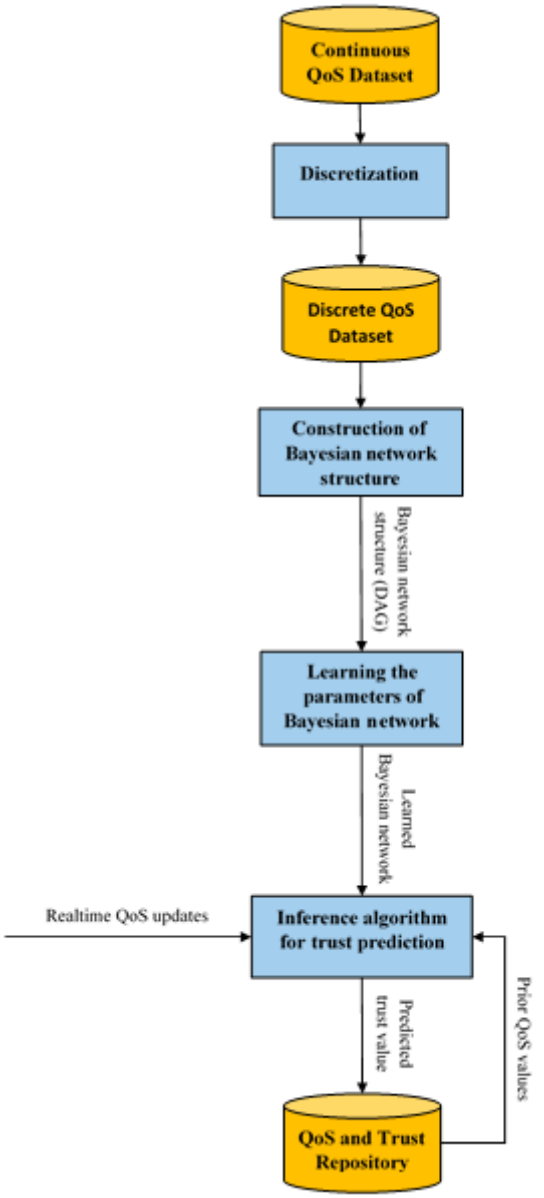

Figure 1

Proposed trust model

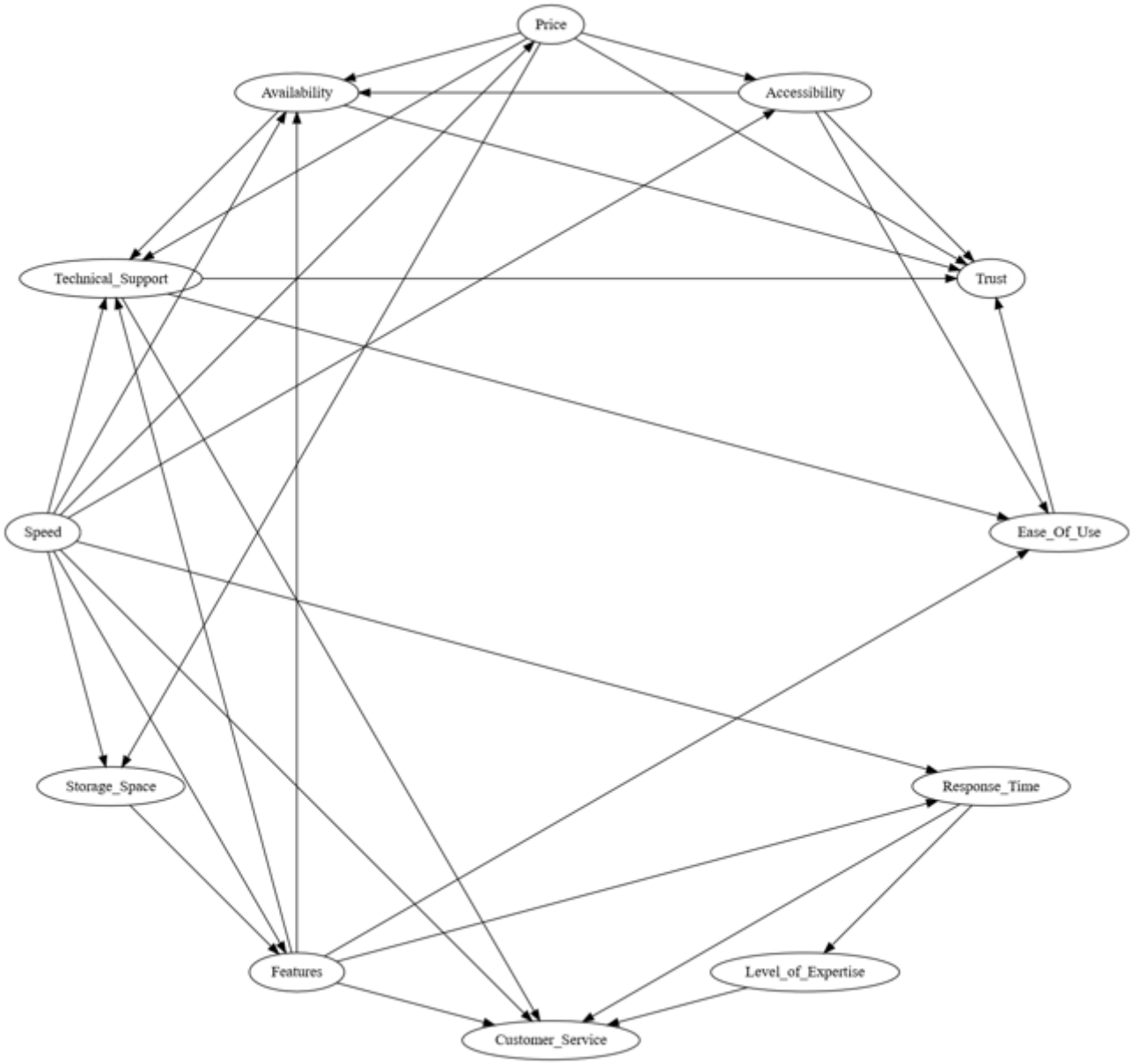

Figure 2

Structure of the Bayesian network in proposed trust model