Rizvi et al. Journal of Cloud Computing: Advances, Systems and Applications (2020) 9:42
https://doi.org/10.1186/s13677-020-00192-9

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

# A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers

Syed Rizvi[1*], John Mitchell[1], Abdul Razaque[2], Mohammad R. Rizvi[3] and Iyonna Williams[1]

## Abstract

Cloud computing is a model for on-demand delivery of IT resources (e.g., servers, storage, databases, etc.) over the Internet with pay-as-you-go pricing. Although it provides numerous benefits to cloud service users (CSUs) such as flexibility, elasticity, scalability, and economies of scale, there is a large trust deficit between CSUs and cloud service providers (CSPs) that prevents the widespread adoption of this computing paradigm. While some businesses have slowly started adopting cloud computing with careful considerations, others are still reluctant to migrate toward it due to several data security and privacy issues. Therefore, the creation of a trust model that can evolve to reflect the true assessment of CSPs in terms of either a positive or a negative reputation as well as quantify trust level is of utmost importance to establish trust between CSUs and CSPs. In this paper, we propose a fuzzy-logic based approach that allows the CSUs to determine the most trustworthy CSPs. Specifically, we develop inference rules that will be applied in the fuzzy inference system (FIS) to provide a quantitative security index to the CSUs. One of the main advantages of the FIS is that it considers the uncertainties and ambiguities associated with measuring trust. Moreover, our proposed fuzzy-logic based trust model is not limited to the CSUs as it can be used by the CSPs to promote their services through self-evaluation. To demonstrate the effectiveness of our proposed fuzzy-based trust model, we present case studies where several CSPs are evaluated and ranked based on the security index.

**Keywords:** Cloud computing, Trust model, Cloud service user, Cloud service provider, Fuzzy-logic

## Introduction

Cloud computing is a distributed computing environment capable of storing large quantities of data, increasing data processing efficiency, and scaling an application based on demand [1]. It utilizes virtualization to provision resources and offers web-based services at a fixed cost. Collectively known as the SPI model, the three main services that this model offers are: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Google Docs, Google App Engine, and Amazon Elastic Compute Cloud (EC2) are examples of SaaS, PaaS, and IaaS, respectively. Furthermore, the cloud supports a variety of deployment models such as private, community, public, or hybrid models. Some of the benefits of employing a cloud service or deployment method include flexibility, scalability, pay-per-use, and accessibility through a web browser [2]. In addition, the amalgamation of IT resources, data warehousing, computation outsourcing, and multi-tenancy are several important concepts that increase cloud efficiency. On the other hand, these features also provide several opportunities for attackers to perform malicious activities such as compromising the integrity of the outsourced data or unauthorized access to cloud services, etc. Thus, the fundamental concepts present a challenge in the security and privacy domain that must

* Correspondence: srizvi@psu.edu
[1]Department of Information Sciences and Technology, Pennsylvania State University, Altoona, PA 16601, USA
Full list of author information is available at the end of the article

be resolved in order to establish trust among the cloud stakeholders (e.g., CSUs, CSPs, third parties, etc.).

In the present scenario, trust plays a key role in influencing the decision to utilize cloud services for enterprises. Trust is based on one entity's own experiences within a specific amount of time and involves the belief in a second entity's capabilities, reliability, and honesty [3]. Trust can be dissected into two types: subjective and objective. Subjective trust focuses on the user's perception of a reputation that develops from utilizing a service. Subjective trust is pertinent as it emphasizes the gradual formulation of trust based upon continuous interaction and encompasses one of the main trust components which is the user's perception of the intentions of the other party [4–6]. The subjective trust used in this paper combines direct, recommendation, and reputation trust in order to create a comprehensive trust but with a degree of uncertainty [3]. The second type is an objective trust that involves quantitative data. Objective trust only takes into account the data, graphs, and equations involved to assess the mathematical trust value. Objective trust portrays a black and white numerical value of trust rather than considering the grey areas that arise when dealing with cloud service users choosing a CSP. Since the input of the CSUs are the source of our data sets, the use of subjective trust can be well justified with our fuzzy-logic based trust model. The operationalization of subjective trust is found in quantifying the overall trustworthiness of a CSP based on the input of cloud users.

Trustworthiness of a CSP and the willingness of the CSU are two important factors to analyze when developing a trust model. Factors that influence trust when dealing with technology vary as there are influences that deal with systemized information about a particular device or service, and aesthetics or design that appeals to the user in order to formulate persuasion of gained user trust. While the popular constituents of subjective trust encompass its quantitative approach, there are qualitative characteristics that heavily influence subjective trust such as aesthetics. Gaining a user's trust not only focuses on rationality but if the user enjoys the product, its design, and the ease of operation [7]. Researchers have previously found that understanding how emotions relate to trust is a key component in gaining full trust of a user. Allowing a user to be emotionally expressive will incline them to increase their trustworthiness of a service [8]. Our use of fuzzy logic addresses the emotional aspect of subjective trust as it analyzes the uncertainties that may arise when users assess the reliability of a CSP as well as their inclination to trust a certain service because of aesthetics. Trust values can be obtained from our proposed model that will allow for a comparison between the CSPs. To calculate trust, factors that influence the trustworthiness of a CSP must be identified [9].

As stated earlier, trust cannot be established until the security concerns associated with cloud computing are evaluated. The security of a CSP is one of the most significant attributes that should be analyzed in order to determine the trust value. The security issues generating the most concerns include data breaches or losses. In general, the main problem is that most of the CSUs presently do not feel confident in outsourcing their sensitive information into the cloud due to security and privacy concerns [10]. Specifically, the most pertinent security issues are; lack of security standards, lack of service-level agreements (SLAs), vulnerabilities with the internet protocol, malicious insiders, account hijacking, insecure API interface, multi-tenancy, and data privacy, to name a few. For instance, the abstraction of SLAs damages the reliability of CSPs since it does not provide guarantees in regard to ownership of data, availability of services, transparency in the deployed security controls, etc. against many different types of attacks. For example, Internet vulnerability protocols are network-based and may allow for distributed denial-of-service (DDoS) attacks [11]. Similarly, malicious insiders could erase user data and potentially sell sensitive information to rival organizations. Furthermore, the virtualization aspect of the cloud facilitates the most serious attacks since virtual machines are utilized by attackers [12].

To address these concerns, several frameworks and models have recently proposed to evaluate CSPs and establish trust between customers and service providers that can evolve. In general, the existing trust models [13–15] for cloud computing are not fully capable of providing an accurate method for measuring trust, especially in the presence of incomplete sets of input data or observations. The CSPs differ with each other in terms of their security strengths, which present a challenge for researchers trying to develop an efficient and accurate model for security evaluation. Several limitations must be overcome to develop a strong trust model that can accurately reflect a true comparison of competing CSPs using partial sets of input data or observations. For example, some trust models only measure the indirect trust based on the user's recommendations. The disregard for direct evidence results in the lack of suitable evaluation factors or a dependence on recommendations to quantify results [3]. Another limitation is the fixed weight assigned to each factor [13]. The expert knowledge used for weight allocation does not allow for adaptability in a practical environment. We believe that the preferences of the CSUs should be taken into account in determining the importance and weight of the factors. Currently, many trust models [13–15] used for evaluating CSPs or similar distributed systems are dependent on certain values or subjective logic. However, there is a general vagueness that surrounds the

concept of trust which promotes uncertainties during the evaluation process. For instance, one CSU could consider a CSP to be trustworthy, while the same CSP appears entirely untrustworthy to another CSU. Subjective logic is adopted in other trust models to accept uncertain values and increase the accuracy of the final results. On the other hand, subjective logic fails to accept values that are represented as vague statements. Furthermore, linguistic statements enable the CSU to input data more accurately since linguistics are easy for humans to understand [10].

Motivated by this, we develop a trust model that can assist a CSU in comparing the potential CSPs using their own security preferences. This research work is the continuation of our initial findings published in [16]. To further extend our trust model, we implement fuzzy logic to process the subjectivity of the CSUs and obtain the most accurate results. Fuzzy logic uses values that are not restricted to a binary decision of yes or no [2]. Instead, the attribute being evaluated is assigned a membership that ranges from low to high. Therefore, fuzzy logic can accept impartial data sets, which is a high possibility when evaluating trustworthiness due to ambiguity and uncertainty being common in these circumstances. There are three stages of fuzzy logic which include Fuzzification, inference engine, and defuzzification [17]. Fuzzification involves the CSU inputting the data which is then matched with the factors used for evaluation and then converted into a fuzzy data set. The inference engine contains nonlinear mapping and the mapping rules that determine the output. Finally, the output (Security Index) is approximated after defuzzification when the data sets are turned into crisp data. To support the three stages of fuzzy-logic, we identify the evaluation factors using linguistic variables which make our trust-model user friendly. In addition, we develop mapping rules for the inference engine to process the input data set.

Research Contributions: The following are some of the contributions that make our research unique from others existing work:

1. Use of Fuzzy logic to evaluate CSPs with a new trust model: Our first main contribution is utilizing the process of fuzzification and defuzzification on the set of input data in order to evaluate service providers. The set of input data is often incomplete with ambiguity and uncertainty – making it difficult for to apply conventional methods of evaluation. Our contribution differs from other research as we use the fuzzy logic process for CSP security assessment and to establish trust between a CSU and CSP even in the presence of impartial data sets.
2. Developing experimental case studies: Our second main contribution is developing case studies based on hypothetical scenario to exemplify how our proposed trust model can be used in real world situations. Our first case study analyzes the event of a CSU evaluating their current CSP to determine its security strength and weaknesses. Our second case study shows how a CSP can use our proposed model to self-evaluate. Our last case study exemplifies the concept of a CSU trying to find the best CSP for security and using our model for a better security assessment.
3. Trust level validation of a CSP: Our third main contribution is the addition of mathematical model to evaluate the trust of a CSP. Using our fuzzy-logic trust model, we can use the equations to not only perform a security assessment, but also audit CSPs to determine their trust level.

## Related work

In the context of cloud computing, there are several trust models recently proposed using a variety of methods to maintain the confidence of clients. Currently, most of the trust models proposed in literature are dependent on the assumptions or inefficient parameters which limit the practicality. This section highlights a few notable works, closely related to ours, whose authors have presented significant contributions in the trust management domain.

Rathi and Kolekar [18] specifically narrow the various parameters involved in trusting a CSP by separating the components that are most valuable when choosing a faithful service. The generated trust model that is implemented focuses on safety of data, individuality management, authorization, authentication, and virtualization to provide maximum efficient security involving a CSP. The proposed trust model analyzes security parameters evaluated by a cloud consumer as the input value and the trust value is calculated by a point-based system that takes the ratio of user gained points to total points possible. This allows the user to choose their cloud service that can cater to their specified needs. The problem with this approach is that the cloud user may not necessarily choose the CSP that has maximum benefits in each parameter; what may be trusted in a specific area may not be as effective in others. Ragavendiran et al. [10] propose a model to develop a trust score of a CSP by comparing Service Broker and Load Balancing policies and using the fuzzy inference system. In this approach, three Service Broker Policies and three Load Balancing Policies are taken into account when calculating the trust score of a CSP. The architecture of the trust model includes User Bundles and Data Centers of the service providers to individually compare the services for optimal selection for the user. The Service Broker and Load Balancing policies are used to consider the Data Center sectors such

as IP, network, storage and applications. The policies then operate on user input such as their region, which then calculates the number of Data Centers in that particular region, and finally sends the request to that specific Data Center. Fuzzy inference allows aggregation and defuzzification to then determine which factors of the CSPs from the Data Centers are directly proportional to determine the overall trust score. The drawback to this approach is that cloud service configuration varies, and the trust model assumes that all configurations across the regions are alike. Farrokhi and Nalbandian [19] propose a trust using fuzzy logic similar to [10], however, the main focus is on the management of trust. The trust assessment is based on an equation that calculates the difference of direct and indirect trust along with trust weight and interaction weight. The downside to using this method is that calculating trust in a fuzzy environment involves cloud network types, trust management types, and trust scores; that is not addressed by the authors.

Mohammed et al. [9] proposed a trust model similar to [18] where the trust percentage is calculated for the consumer based on the trust metric stage. However, the parameters include reputation, customer experience, majority consensus, and cloud service consumer's capability which differs from [18]. The trust metric stage is where the trust percentage for each consumer is calculated and uses techniques such as Particle Swarm Optimization, Multiple Regression, Analytic Hierarchical Process, and Particle Swarm Optimization-Multiple Regression. However, although these techniques seem to be accurate, there still is a percentage of average deviation that would only be fixed by developing a scheduling process for the consumers to test the model based on their trust values. Wang et al. [3] proposed a trust model that encompasses both weights and gray correlation analysis. The generalized trust is converted into subsections that include recommendation trust, comprehensive trust, and overall trust. Using the rough set theory and analytic hierarchy process-based method, direct trust is calculated which then can provide an accurate overall trust. A flaw in this proposed trust model is that the satisfaction factor that is implemented with recommendation trust. Even if the reputation seems to be good, there is not a guarantee that the service provided is efficient due to the voting being based on human interaction.

Wen et al. [20] proposed a dynamic cloud service trust evaluation model that operates on service-level agreement and privacy-awareness. The proposed model has components similar to [3] where trust is broken down into various evaluations including direct, indirect, and reputation trust to calculate the final trust. The model encompasses a dynamic trust update mechanism to consistently update direct trust and deliver a dataset to the user that is based on user preferences, satisfaction, accuracy, and feasibility. This approach can be inaccurate due to the user feedback mechanism being inconsistent. Rizvi et al. [21] proposed a trust model where a set of third-party auditors are utilized to establish trust between CSUs and CSPs using both direct and indirect trust characteristics. Specifically, the role of different auditors is to evaluate the security strength of CSPs using a pre-defined criterion and provide a fair third-party assessment that can be later used to produce a trust value. Although their approach well justified the advantages of third-party assessment, they did not provide any mechanism to use that as an assessment. Thus, they were unable to compute a final trust score for each CSP.

Alruwaythi et al. [22] presents the effects of user behavior in modeling trust which is associated when evaluating the principles between different sets of trust models. User behavior is considered in this model where the principles include expired, recent, and abnormal user behavior. Using these principles, trust value either decreases or increases depending on the number of malicious behaviors that have been detected. The evidence of user behavior is calculated through evidence, logins, operations, reliability, and performance. This trust model is unreliable because it is completely dependent on user behavior which makes accuracy sporadic as the behavior would have to be of an exemplary user. Chen et al. [23] proposed a trust model that is similar to [22] as it is based on user behavior; however, the priority is to ensure safety of users which makes the main focus on degrees of security. Recommendation, direct, and comprehensive trust are calculated by interaction between users and other cloud users along with the weighted average method. The proposed model, being based on user behavior, monitors abnormal behavior more effectively by calculating the trust value based upon cloud user behavior. This model has its benefits including the fact that the trust value can change with the environment because it is based on user input rather than generalization. However, abnormal users affect the security and trust value. Werff et al. [24] proposed an experimental analysis approach to cloud trust. The experimental analysis includes various cloud solutions that explore the influence of information to consumers when they are choosing a CSP. A survey was conducted to conclude that positive and negative labels on cloud trust affect the selection of user chosen CSPs. The fault of this experiment, to be used when evaluating trust, is that the sample of people being surveyed could not be large enough to detect a consistent pattern in choosing the right CSPs for different users.

Xu et al. [25] incorporates quality-of-service (QoS) data to determine which CSPs are reliable and the effect

that unreliable QoS data can have on the criteria for CSP selection. The model presents a cloud service selection framework that is based on a reputation approach. This approach encompasses a new user reputation calculation named MeURep. The algorithms follow parameters based on QoS such as robustness, parallelization, and efficiency. However, this model does not consider other factors that are pertinent in choosing a CSP such as security, reliability, etc. that have been mentioned in previous trust models. Lu et al. [26] proposed a trust assessment model that is based on recommendations and dynamic self-adaptation in cloud services. Trust is broken down into indirect and direct trust as exemplified by previous trust models. The recommendation mechanism is operated by third-party nodes that make recommendations based on situations where it either accepts or rejects circumstances after interaction. This process helps mitigate the risk of malicious node attacks which, in turn, protect the cloud user and CSP. A disadvantage to this approach is that it does not protect against more complex malicious attacks which can have a detrimental impact on the cloud user.

Huang et al. [27] propose a trustworthy framework for cloud security called the trustworthy computing framework (TwCF). The framework highlights the policies that are modified to adapt to a users' security contents. The authors also utilize a trust management framework (TMF) that aids in decision-making of security policies and the platform security mechanism (PSM) is used to ensure that the correct security policies are being deployed. The authors highlight the fact that due to the lack of trust in cloud infrastructure, many CSPs are unable to accurately determine their security status. The need for Tailored Trustworthy Spaces (TTS) to provide an adaptive and flexible environment for a user is an important component in securing trust between a CSU and CSP. The authors dissect TTS and combine it with multiple security mechanisms to formulate a comprehensive combination of trust and security in cloud infrastructure. While this approach of TTS and subjective trust is fundamental in our proposed fuzzy-logic trust-based model, its disadvantage lies in the lack of fuzzy theory. Without the analysis of the grey areas, it is hard for a CSU to make a truly accurate decision when choosing a CSP. Kurdi et al. [28] introduce a lightweight trust management algorithm that is based upon subjective logic (InterTrust) in order to enhance trust in interconnected cloud computing environments. The authors conducted an experiment that portrays the InterTrust capability of producing trust information efficiently in comparison to other algorithms and subjective trust without an algorithm. Subjective logic is explained in the literature as a proponent of trust that has an input of uncertainty and incomplete knowledge. The authors conclude that the InterTrust algorithm is a possible tool that can be used for trust management for a multitude of cloud environments. While certain components of this research are utilized within our fuzzy-logic based trust model, such as subjective trust, it is not concentrated on the security of cloud environments and mostly has its focus on proving the superiority of InterTrust as solely a trust management system.

Within our proposed model, we have utilized the work from the Mamdani inference system [29] in order to perform fuzzification using their inference rules to provide an efficient application of human reasoning as part of our proposed model. We have also used Kurdi et al. [28] to identify our subjective and objective trust definitions. Other cited works within our proposed model were not utilized explicitly but taken into account by analyzing their limitations to suggest a solution to these disadvantages. Our proposed model is unique compared to other related works because we consider the input from cloud service users and their definition of trust to then synthesize the information into a coherent computational form, and finally determining the trust level and reputation of a cloud service provider. After the trust level is evaluated it can be used for cloud service users to determine which service would best fit their prospected needs.

## Fuzzy-logic based trust model

In the proposed fuzzy-logic based trust model, the security of a CSP is evaluated based on several influential factors using a fuzzy inference system. Security is often influential when determining the trustworthiness of a CSP since data protection is one of the primary concerns for enterprises that are still hesitant to migrate to the cloud. The steep learning curve associated with cloud computing can result in uncertainties and ambiguities among CSUs. This leads to inaccurate decision making and a lack of trust when CSUs attempt to choose the most efficient CSP in terms of security. Therefore, we implement a fuzzy inference system to allow for an accurate selection of CSPs based on the security evaluation performed by the CSU. We used the Mamdani inference system [29] over the Takagi and Sugeno method [30] since it is more efficient for applying intuition and human reasoning with the knowledge of experts as discussed in [31]. Utilization of fuzzy CSU data allows our proposed model to take into account the multiple possibilities or uncertainties that a CSU may have and quantify that data in order to achieve a coherent security analysis of a CSP that would be catered to a CSU. Figure 1 illustrates the basic building block of the inference process that is used to calculate the security index of a CSP. It is shown that the CSU provides numerical (crisp) input, or fuzzy expressions, that are sent and converted into fuzzy

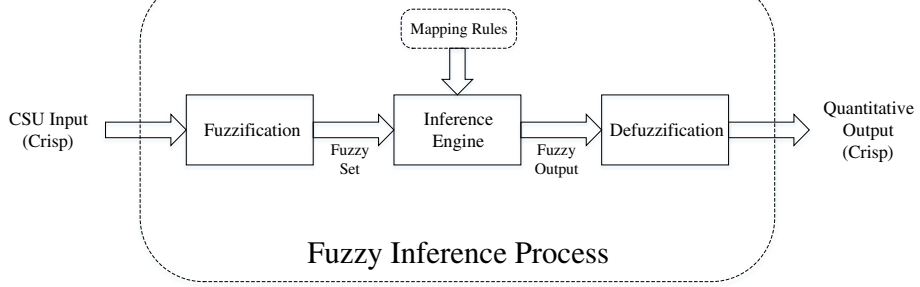


**Fig. 1** Overview of Fuzzy Inference System

quantities (fuzzification). Rules are then applied to the fuzzy set to generate fuzzy output (inference engine), and further converted from fuzzy output to crisp output (defuzzification). The inference process of Fuzzification, Inference Engine, and Defuzzification are further elaborated in the subsequent subsections.

### Fuzzification

Fuzzification is the process of transforming crisp values into fuzzy values. Specifically, the crisp data provided by the CSU is mapped to a fuzzy set which contains the membership functions and linguistic values [32, 33]. A fuzzy set is defined by the members it contains as shown in $x \in X$ where $x$ is the element. The symbol ∈ demonstrates that $x$ is a member of the specified set $X$. In particular, a fuzzy set is defined by ordered pairs as shown below:

$$A = \{(x, \ddot{E}A(x)) \mid x \in U\} \quad (1)$$

where $U$ is the universe of discourse which contains all of the elements that may be used in fuzzy set $A$. The membership functions are read as μA(x) wherein a membership grade is assigned in the closed interval [0,1] to each element in $U$. We utilize this formula in our proposed model in order to take the CSU fuzzy input for the process of fuzzification. Figure 1 shows the simplified fuzzy inference process that we have utilized in our fuzzy-logic model in order to process the information of the CSU.

In our proposed approach, the linguistic variables represent the fuzzy sets $f_i$ (i = h, r, a, g, e) which consist of harmful, risky, average, great, and exceptional. Each CSU could interpret the meaning of these variables differently. Therefore, the membership functions are employed in the fuzzy sets to define the meaning of the linguistic variables. In our approach, CSUs can utilize these linguistic variables in order to make a decision on which CSP would better fit their security needs. Table 1 shows the membership degrees of the variables. Once the values are assigned, the linguistic variables can be used by the CSUs to help evaluate the security of the CSPs.

There are multiple factors that affect the security of a CSP which can be evaluated by the CSU to perform an overall security assessment. We identify four factors used in our fuzzy-logic approach: compliance, access controls, auditability, and encryption. For simplification, we do not utilize the sub-factors in our approach due to the dependence and relationship among the aforementioned factors. Our proposed model uses compliance to measure the membership degree in which CSPs protect their CSU's data, auditability in order to make sure the CSP has the adequate management controls to protect the CSU, and encryption to ensure CSU data safety. Figure 2 illustrates the membership functions of the possible CSU input. The linguistic variables correlate to their respected membership degree with a corresponding description for clarification. These linguistic variables are scaled on a membership degree with a range of [0,1] in order to eliminate vagueness and allow an infinite spectrum between membership degrees. Specifically, we use the triangular membership functions in Fig. 2 that can be defined as follows:

$$\mu(x) = \begin{cases} \dfrac{x-a}{b-a}, & \text{if } a \le x \le b \\ \dfrac{x-c}{b-c}, & \text{if } b \le x \le c \end{cases} \quad (2)$$

The triangular membership function requires three parameters represented as $a$, $b$, and $c$ which dictates the

**Table 1** Definition of Linguistic Variables

| Linguistic Variables | Membership Degree | Description |
|---|---|---|
| Exceptional | 80–100 | Near flawless |
| Great | 60–90 | Better than most |
| Average | 30–70 | Uncertain about security |
| Risky | 10–40 | Potentially harmful |
| Harmful | 0–20 | Presents a danger |

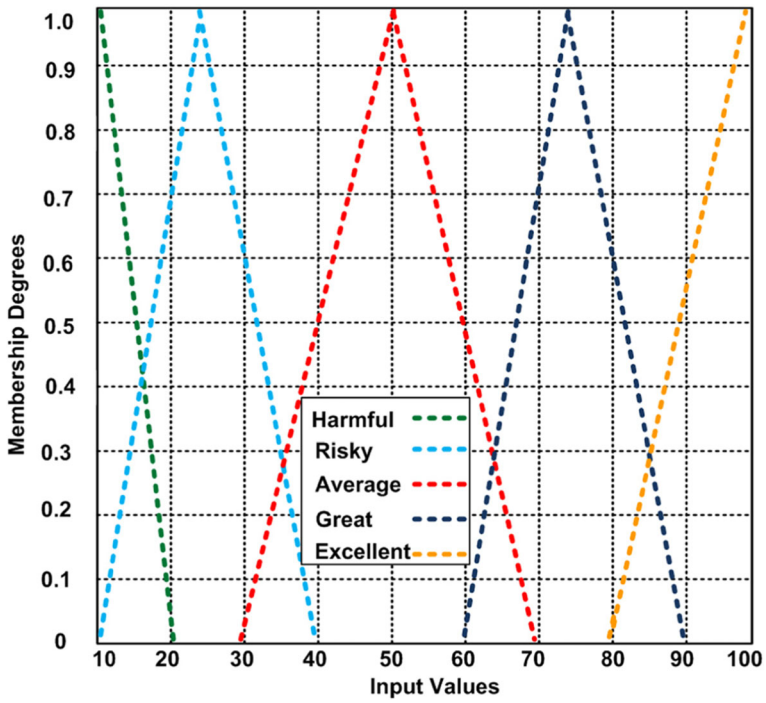


**Fig. 2** Membership Functions of Linguistic Variables

three corners of the triangle. For instance, triangle (x; 30, 50, 70) in Fig. 2 is the representative of the average fuzzy set. First, compliance is an important security issue in cloud computing since non-compliance can facilitate threats that propagate to the other security factors. The governing bodies and international standard entities that represent the foundation of cloud security require compliance from the CSP. Without compliance, these entities are ineffective to prevent and mitigate the damage caused by an attacker. Furthermore, service level agreements (SLAs) and relevant policies/procedures that describe the performance, availability, payment, and delivery will be ineffective if the CSP does not comply with its own policies [11]. Compliance risks contribute to an inadequate security index-score as well as the enlargement of distrust between the CSUs and CSPs. Another influential security factor is the access controls used by the CSP. Thus, the authentication mechanisms are a vital component for maintaining the confidentiality and integrity of data.

In addition, many attacks are aimed at exploiting the access controls of the cloud such as phishing attacks. Malicious insiders pose a constant threat to CSUs since the insiders often possess authentication and authorization credentials. Auditability refers to the CSPs undergoing security assessments on themselves which include performing employee background checks, analyzing the access controls, and initiating the vulnerability scans. This will help to prevent malicious insider attacks and unauthorized users from gaining access to cloud services. Moreover, the cloud is a dynamic environment where modifications are made frequently and could promote additional security threats. Thus, the CSP should provide an auditing service to monitor security. Data encryption is another important security requirement that will restore the confidence in CSUs since data privacy and protection are among the primary concerns. If a CSP can provide the option for encrypting the data of a CSU at rest and in transit, it will increase the reliability of the CSP and mitigate future threats such as data breaches or losses.

### Inference engine

The inference engine is responsible for applying the inference rules to the fuzzy input in order to generate the fuzzy output. In particular, the inference rules are used to evaluate the linguistic values and map them to a fuzzy set which requires the defuzzification process to transform it into a crisp value. One of the fundamental concepts of the Mamdani method [29] is the inference rules which provide the computation functionality of the system. These rules can be based on previous experiences,

observations, and the knowledge of experts. Each fuzzy inference rule consists of two concepts which include the if-then statements and the linguistic variables. The if-then rules contain the antecedents and the consequence, where the linguistic input from the CSU is the antecedent and the consequence is the linguistic output based on the input.

When fabricating an inference rule, operators such as "and," "or," and sometimes "not" are used [30]. The amalgamation of operators is referred to as the *t*-norms. The definition of the fuzzy "and" operator is given as:

$$\mu A \cap B(x) = \min[\mu A(x), \mu B(x)] \tag{3}$$

The membership in class A is specified as μA while μB represents the membership in class B. This rule extracts the minimum number of the membership values of the fuzzy sets to compute the "and" operation. The fuzzy "or" operator is defined as:

$$\mu A \cup B(x) = \max[\mu A(x), \mu B(x)] \tag{4}$$

In both eqs. (3) and (4), the $x$ represents the degrees of the membership functions that correspond to the fuzzy sets. For example, $A(x)$ refers to the membership degrees of fuzzy set $A$. The "or" operation is computed by extracting the maximum value of the membership values of the fuzzy sets. When formulating our inference rules, we used the "and" operator since the evaluation factors are dependent on each other. The "or" operator is mostly used for independent factors that are not closely coupled. The operators allow for the CSU to compute a fuzzy output distribution based on the rule strength. The rule strength enables the aggregation of the fuzzy outputs into a distribution. We present several of the rules used in our evaluation of cloud security in linguistic form (see Table 2). The full abbreviation of each variable used in established rules is presented in Table 3.

### Defuzzification

In defuzzification, the fuzzy output of the inference engine is mapped to a crisp value that provides the most accurate representation of the fuzzy set [31]. The classifications for the fuzzy output are identical to the linguistic variables used for the fuzzy input as shown in Table 1. The fuzzy outputs are represented as $G_i$(i = h, r, a, g, e) and share the same membership functions. The crisp output is obtained in our proposed fuzzy approach by using the centroid method [34], which is defined below:

**Table 2** Rules for Evaluating the Service Providers Security using Linguistic Form

| Rule | Description | Rule | Description |
|---|---|---|---|
| 1 | $if\langle CP \triangleq H\rangle \wedge \langle AC \triangleq H\rangle \wedge \langle AU \triangleq H\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong H\rangle$ | 2 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq AV\rangle \wedge \langle AU \triangleq AV\rangle \wedge \langle EC \triangleq R\rangle \underset{then}{\rightarrow} \langle SC \cong AV\rangle$ |
| 3 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq H\rangle \wedge \langle AU \triangleq H\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong H\rangle$ | 4 | $if\langle CP \triangleq G\rangle \wedge \langle AC \triangleq H\rangle \wedge \langle AU \triangleq H\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong H\rangle$ |
| 5 | $if\langle CP \triangleq EX\rangle \wedge \langle AC \triangleq H\rangle \wedge \langle AU \triangleq H\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong H\rangle$ | 6 | $if\langle CP \triangleq R\rangle \wedge \langle AC \triangleq R\rangle \wedge \langle AU \triangleq H\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong H\rangle$ |
| 7 | $if\langle CP \triangleq R\rangle \wedge \langle AC \triangleq R\rangle \wedge \langle AU \triangleq R\rangle \wedge \langle EC \triangleq H\rangle \underset{then}{\rightarrow} \langle SC \cong R\rangle$ | 8 | $if\langle CP \triangleq R\rangle \wedge \langle AC \triangleq R\rangle \wedge \langle AU \triangleq R\rangle \wedge \langle EC \triangleq R\rangle \underset{then}{\rightarrow} \langle SC \cong R\rangle$ |
| 9 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq R\rangle \wedge \langle AU \triangleq R\rangle \wedge \langle EC \triangleq R\rangle \underset{then}{\rightarrow} \langle SC \cong R\rangle$ | 10 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq AV\rangle \wedge \langle AU \triangleq R\rangle \wedge \langle EC \triangleq R\rangle \underset{then}{\rightarrow} \langle SC \cong R\rangle$ |
| 11 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq AV\rangle \wedge \langle AU \triangleq AV\rangle \wedge \langle EC \triangleq R\rangle \underset{then}{\rightarrow} \langle SC \cong AV\rangle$ | 12 | $if\langle CP \triangleq AV\rangle \wedge \langle AC \triangleq AV\rangle \wedge \langle AU \triangleq AV\rangle \wedge \langle EC \triangleq AV\rangle \underset{then}{\rightarrow} \langle SC \cong AV\rangle$ |
| 13 | $if\langle CP \triangleq G\rangle \wedge \langle AC \triangleq AV\rangle \wedge \langle AU \triangleq AV\rangle \wedge \langle EC \triangleq AV\rangle \underset{then}{\rightarrow} \langle SC \cong AV\rangle$ | 14 | $if\langle CP \triangleq G\rangle \wedge \langle AC \triangleq G\rangle \wedge \langle AU \triangleq AV\rangle \wedge \langle EC \triangleq AV\rangle \underset{then}{\rightarrow} \langle SC \cong AV\rangle$ |
| 15 | $if\langle CP \triangleq G\rangle \wedge \langle AC \triangleq G\rangle \wedge \langle AU \triangleq G\rangle \wedge \langle EC \triangleq AV\rangle \underset{then}{\rightarrow} \langle SC \cong G\rangle$ | 16 | $if\langle CP \triangleq G\rangle \wedge \langle AC \triangleq G\rangle \wedge \langle AU \triangleq G\rangle \wedge \langle EC \triangleq G\rangle \underset{then}{\rightarrow} \langle SC \cong G\rangle$ |
| 17 | $if\langle CP \triangleq EX\rangle \wedge \langle AC \triangleq G\rangle \wedge \langle AU \triangleq G\rangle \wedge \langle EC \triangleq G\rangle \underset{then}{\rightarrow} \langle SC \cong G\rangle$ | 18 | $if\langle CP \triangleq EX\rangle \wedge \langle AC \triangleq EX\rangle \wedge \langle AU \triangleq G\rangle \wedge \langle EC \triangleq G\rangle \underset{then}{\rightarrow} \langle SC \cong G\rangle$ |
| 19 | | 20 | $if\langle CP \triangleq EX\rangle \wedge \langle AC \triangleq EX\rangle \wedge \langle AU \triangleq EX\rangle \wedge \langle EC \triangleq EX\rangle \underset{then}{\rightarrow} \langle SC \cong EX\rangle$ |

**Table 3** Abbreviations used in Established Evaluations Rules

| | |
|---|---|
| Access Control | *AC* |
| Auditability | *AU* |
| Average | *AV* |
| Compliance | *CP* |
| Encryption | *EC* |
| Exceptional | *EX* |
| Great | *G* |
| Harmful | *H* |
| Security | *SC* |

$$z = \frac{\sum_{j=1}^{n} z_j \mu_c(z_j)}{\sum_{j=1}^{n} \mu_c(z_j)} \quad (5)$$

The centroid method uses the center of mass, which is represented as $z$, in a fuzzy output distribution to determine a single scalar value. The membership of the fuzzy sets is presented in $u_c$, whereas the value of the membership is represented as $z_j$. Finally, the crisp output from the defuzzifier is an approximation that is used to represent the security index of a CSP based on the evaluation of the top-level factors by the CSU. The CSU can then use the index of a CSP to review if the security of the CSP is sufficient enough for their needs.

## Experimental evaluation using case studies

In this section, we present three case studies that demonstrate the effectiveness of our proposed fuzzy-logic approach. The first case entails a single evaluation of a CSP from the viewpoint of a CSU whereas the second case encompasses the self-evaluation of a CSP by multiple CSUs. Lastly, the third case presents a CSU who is trying to choose the most secure CSP for data migration and ranks them according to the security index. For the case study evaluations, we assume that the fuzzy sets described in the previous section will be utilized. In addition, the universe of discourse U = {0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100} is discrete, and the membership degrees for each fuzzy set are shown in Table 4. The evaluation factors are abbreviated as follows: compliance = CP, access control = AC, auditability = AU, and encryption = EC.

### Case 1 – CSU evaluating security strength of a CSP

In this scenario, a CSU has already been using the cloud to archive important data. However, the extent

**Table 4** Fuzzy Set Classifications

| Name | Elements in Set | Membership Degrees |
|---|---|---|
| Harmful ($f_h$) | 0 | 1 |
| | 5 | .75 |
| | 10 | .5 |
| | 15 | .25 |
| | 20 | 0 |
| Risky ($f_r$) | 10 | 0 |
| | 15 | .33 |
| | 20 | .66 |
| | 25 | 1 |
| | 30 | .66 |
| | 35 | .33 |
| | 40 | 0 |
| Average ($f_a$) | 30 | 0 |
| | 35 | .25 |
| | 40 | .5 |
| | 45 | .75 |
| | 50 | 1 |
| | 55 | .75 |
| | 60 | .5 |
| | 65 | .25 |
| | 70 | 0 |
| Great ($f_g$) | 60 | 0 |
| | 65 | .33 |
| | 70 | .66 |
| | 75 | 1 |
| | 80 | .66 |
| | 85 | .33 |
| | 90 | 0 |
| Exceptional ($f_e$) | 80 | 0 |
| | 85 | .25 |
| | 90 | .5 |
| | 95 | .75 |
| | 100 | 1 |

**Table 5** Membership of Input Values

| Evaluation Factors | Input Values | Fuzzy Sets | Membership Degrees |
|---|---|---|---|
| Compliance | 15 | Harmful | .25 |
| | | Risky | .33 |
| Access Control | 35 | Risky | .33 |
| | | Average | .25 |
| Auditability | 10 | Harmful | .5 |
| Encryption | 75 | Great | 1 |

of cloud security risks are increasing and poses a constant threat to anyone outsourcing the information to a CSP. This causes the CSP to become untrustworthy to the CSU. To determine if their CSP is trustworthy, the CSU evaluates the security of the CSP based on the previously identified security factors to ensure that their data is protected. Suppose the input values are CP = 15, AC = 35, AU = 10, and EC = 75. Table 5 shows the corresponding membership values and fuzzy sets.

Let us assume that there are two rules which fulfill the conditions stated in Table 5 which are:

$$if \langle CP \hat{=} H \rangle \wedge \langle AC \hat{=} R \rangle \wedge \langle AU \hat{=} H \rangle \wedge \left\langle EC \hat{=} G \right\rangle \underset{then}{\rightarrow} \langle SC \cong R \rangle$$

$$if \langle CP \hat{=} R \rangle \wedge \langle AC \hat{=} AV \rangle \wedge \langle AU \hat{=} H \rangle \wedge \left\langle EC \hat{=} G \right\rangle \underset{then}{\rightarrow} \langle SC \cong AV \rangle$$

For the above two evaluation rules, Eq. (3) is used to compute the "and" operators and create the fuzzy output set for each rule as demonstrated below:

$$\mu\text{security} = f_r = \min[\mu c = f_h(15), \mu ac = f_r(35), \mu au = f_h(10), \mu e = f_g(75)] = \min[.25, .33, .50, 1] = .25 \quad (6)$$

$$\mu\text{security} = f_a = \min[\mu c = f_r(15), \mu ac = f_a(35), \mu au = f_h(10), \mu e = f_g(75)] = \min[.33, .25, .50, 1] = .25 \quad (7)$$

The membership degrees generated by the fuzzy inference rules are used to determine the rule strength for each given rule. The CSU can aggregate the fuzzy outputs into a distribution which can then be used for extracting the security index score. Figure 3 illustrates the rule strength of the risky fuzzy output set. Figure 4 displays the rule strength for the average fuzzy output set and shows the interconnection between the risky and average fuzzy sets portrayed by the arrow. Furthermore, Fig. 5 shows the output aggregation that forms from both fuzzy output sets at a closer view from Fig. 4.

Once the output distribution is fabricated, the CSU can use Eq. (5) to extract the security index of the CSP. According to the centroid method [34], the final security index score can be computed as follows:

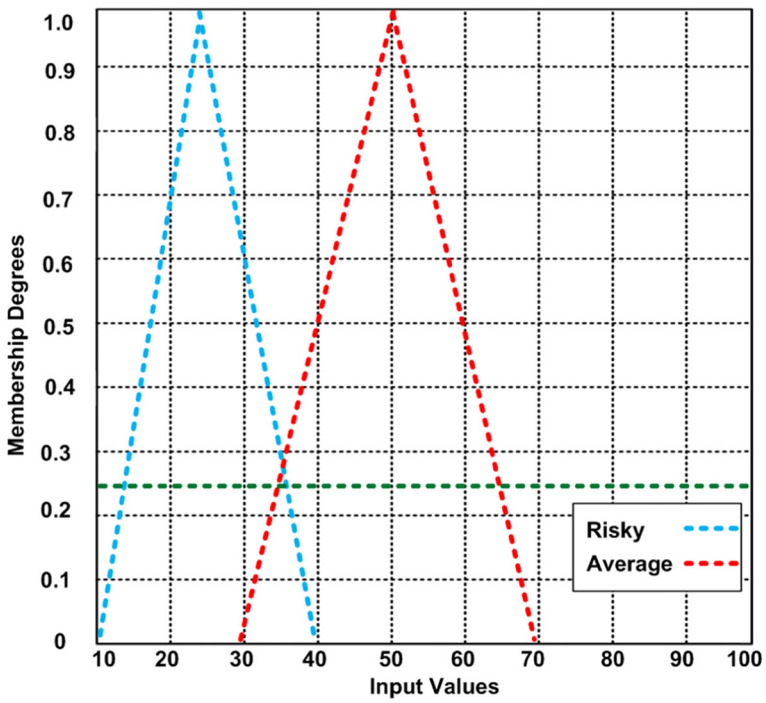


**Fig. 3** Rule Strength for Risky Fuzzy Set

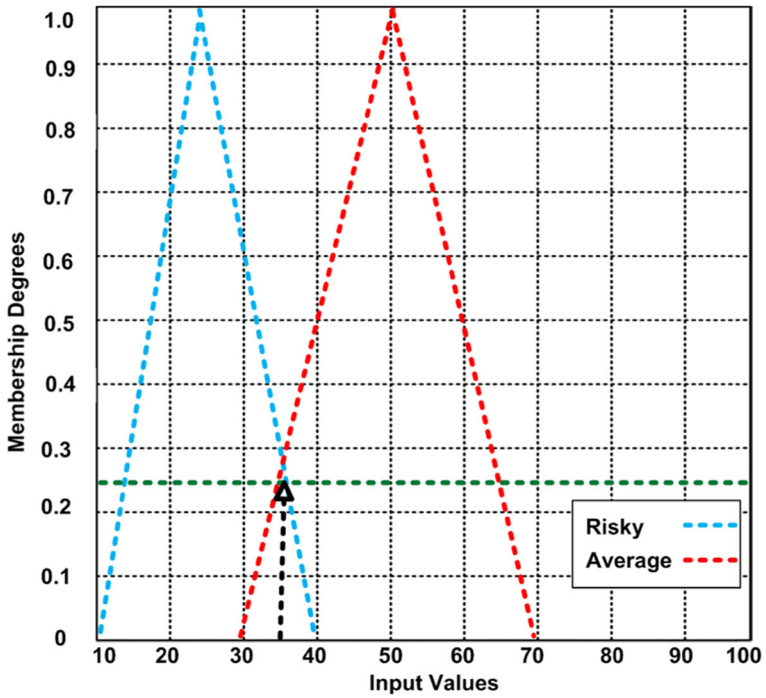


**Fig. 4** Rule Strength for Average Fuzzy Set

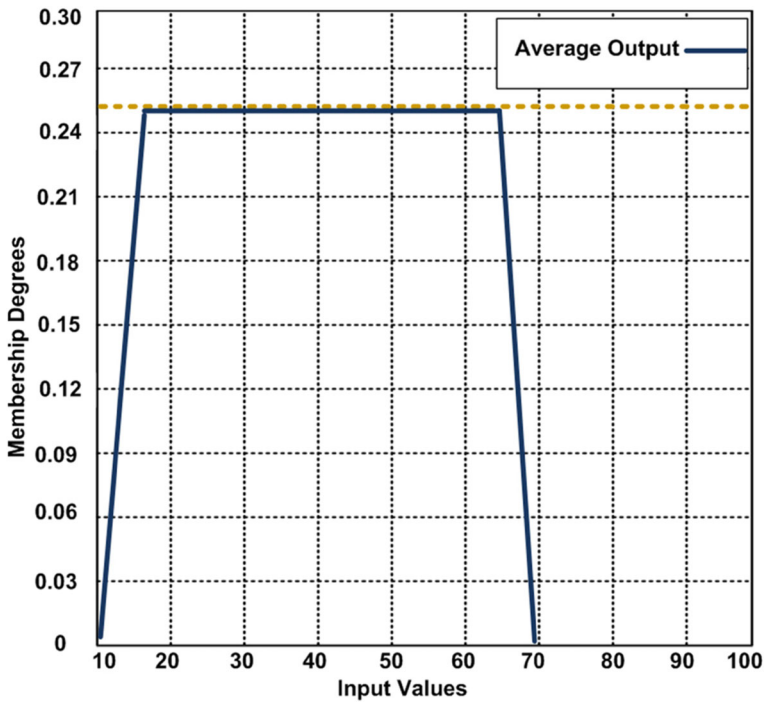


**Fig. 5** Output Distribution of Both Fuzzy Sets

$$Security\ Index\ (SI) = \frac{[((10+15+20+25+30+35+40)\times 0.25 + (30+35+40+45+50+55+60+65+70)\times 0.25)]}{.25\times 7 + 0.25\times 9} = 39.06$$

With the security index score, the CSU can decide if the given CSP offers a sufficient level of security to protect the customer's information in the cloud storage from threats and attacks. Since a security index of 39.06 is not exceptionally well, the CSU may want to change their CSP in order to receive better protection from other potential service providers. This particular case study has shown that our proposed fuzzy approach can be effectively used by a CSU that is uncertain about the level of security offered by a CSP. .

#### Case 2 – self-security assessment by CSP

Suppose that a CSP wants to evaluate its own security strength. In order to prevent personal bias from affecting the security index, the CSP promotes a security evaluation of itself from the CSUs. Let us assume that there are three CSUs that are chosen to evaluate the security of the CSP using our fuzzy-logic approach. The CSUs perform their evaluations separately and discontinue the approach once they complete the fuzzy inference process. Each CSU presents one fuzzy output set which consists of the input value and the membership degree that was computed based on the fuzzy inference rules. The fuzzy output sets are received as $G_g = (85, .33)$, $G_e = (90, .5)$, and $G_a = (65, .25)$. These fuzzy values are used by the CSPs to compute a crisp security index. The CSPs continue the fuzzy approach by aggregating the fuzzy output values into an output distribution. Figure 6 displays the fuzzy output distribution which will be used for the defuzzification process. The graph of the output distribution in Fig. 6 is sectioned in increments of five to be utilized in the Security Index as shown by the green lines between the input values from 30 to 100. Next, the centroid method is used to extract the security index from the center of the newly formed output distribution. Using this method, the aggregated security index score can be computed as follows:

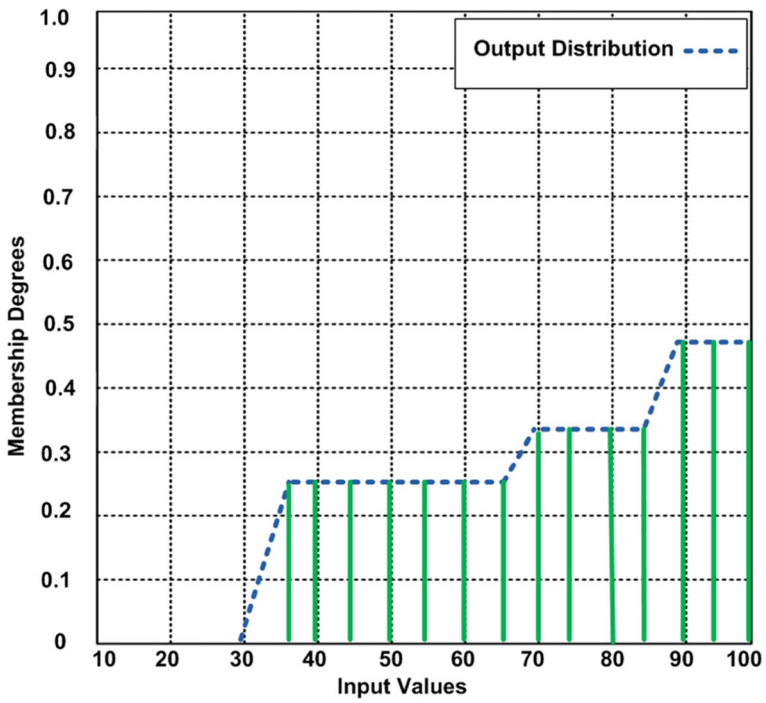


**Fig. 6** Case 2 Output Distribution

$$\text{Security Index (SI)} = \frac{[((30+35+40+45+50+55+60+65+70)\times 0.25+(60+65+70+75+80+85+90)\times 0.33(80+85+90+95+100)\times 0.5)]}{0.25\times 9+0.33\times 7+0.5\times 5} = 72.34$$

The CSP has obtained a security index of 72.34 which proves that the security level is better than most of the other CSPs. This case study has shown that our fuzzy approach is beneficial for both the CSU and CSP. It can be used for self-evaluation and to motivate a CSP to improve its security compliance, auditability, encryption, service quality, etc. Furthermore, the CSP can calculate a more accurate security index if more CSUs are providing the fuzzy output sets. The computed aggregated security index score can be used by the CSP to advertise its security strength including the compliance and other relevant information so that more cloud customers can be attracted.

### Case 3 – ranking the CSPs

Let us consider that a CSU desires to have the most efficient CSP in terms of security. In addition, there are several CSPs that appear to have a credible security presence. Our proposed fuzzy approach allows the CSU to evaluate each CSP and rank them according to the security index. Suppose that the CSU has computed the fuzzy output sets for five different CSPs named $CSP_1$, $CSP_2$, $CSP_3$, $CSP_4$, and $CSP_5$. The output sets for each CSP are as follows: $CSP_1$ = {(80, .66), (95, .75)}, $CSP_2$ = {(95, .75), (90, .5)}, $CSP_3$ = {(85, .33), (65, .33)}, $CSP_4$ = {(90, .5), (70, .66)}, $CSP_5$ = {(95, .75), (85, .25)}. Table 6 displays the CSP fuzzy output values in accordance to the output fuzzy sets. By using Eq. (5), the CSU can calculate the security indexes (SI) for the selected CSPs.

**Table 6** Fuzzy Output Classification of each CSP

| Name | Input Values | Fuzzy Sets | Membership Degrees |
|---|---|---|---|
| $CSP_1$ | 80 | Great | 0.66 |
| | 95 | Exceptional | 0.75 |
| $CSP_2$ | 95 | Exceptional | 0.75 |
| | 90 | Exceptional | 0.05 |
| $CSP_3$ | 85 | Great | 0.33 |
| | 65 | Great | 0.33 |
| $CSP_4$ | 90 | Exceptional | 0.05 |
| | 70 | Great | 0.66 |
| $CSP_5$ | 95 | Exceptional | 0.75 |
| | 85 | Exceptional | 0.25 |

Each SI corresponds to a particular CSP in the following manner: $SI_1 = CSP_1$, $SI_2 = CSP_2$, $SI_3 = CSP_3$, $SI_4 = CSP_4$, and $SI_5 = CSP_5$. The computation of security score for $CSP_1$ is as follows:

$$SI_1 = \frac{[((60+65+70+75+80+85+90)\times 0.66+(80+85+90+95+100)\times 0.75)]}{0.66\times 7+0.75\times 5} \triangleq 81.72$$

The CSU then computes the second SI given as:

$$SI_2 = \frac{[((80+85+90+95+100)\times 0.75+(80+85+90+95+100)\times 0.5)]}{0.75\times 5+0.5\times 5} \triangleq 90$$

The third calculation of security score is computed as:

$$SI_3 = \frac{[((60+65+70+75+80+85+90)\times 0.33+(60+65+70+75+80+85+90)\times 0.33)]}{0.33\times 7+0.33\times 7} \triangleq 75$$

The fourth calculation of security score yields:

$$SI_4 = \frac{[((80+85+90+95+100)\times 0.5+(60+65+70+75+80+85+90)\times 0.66)]}{0.5\times 5+0.66\times 7} \triangleq 80.27$$

Finally, the fifth security index computation is as follows:

$$SI_5 = \frac{[((80+85+90+95+100)\times 0.75+(80+85+90+95+100)\times 0.25)]}{0.75\times 5+0.25\times 5} \triangleq 90$$

The results of this security index computation are summarized in Table 7. The SIs allow the CSU to rank the CSPs based on their security strength and determine the most secure CSP. Table 7 shows the ranking order of the CSPs based on the computations performed earlier using Eq. (5). There is a tie for first place which can be resolved with a larger universe of discourse used for the membership functions. A higher set of numbers for possible input values promotes a more accurate SI. These rankings identify reliable and secure CSPs for the CSUs while encouraging improvement among the CSPs to increase their ranks.

**Table 7** Rankings of the CSPs

| Rank | Name | Security Index |
|---|---|---|
| 1 | $CSP_2$ | 90 |
| 1 | $CSP_5$ | 90 |
| 3 | $CSP_1$ | 81.72 |
| 4 | $CSP_4$ | 80.27 |
| 5 | $CSP_3$ | 75 |

## Trust level validation of cloud service provider

The CSP validation requires that the fundamental security factors (such as access control, auditability, and encryption) must satisfy the customer's security preferences. The trust-value can be calculated based on the level of satisfaction a CSU receives from a given service provider. More specifically, we derive the closed-form expressions for the trust-level for each of these important factors in the next subsequent subsections.

### Trust-level auditability

The trust-level for auditability of a CSP can be determined by using two types of characteristics. First, a CSP may receive a high reputation (or we can refer to its ranking as stated earlier in the previous section) if it is trusted by a large number of CSUs. Second, that reputation can be used by CSUs to make informed decisions for choosing cloud services from one or more CSPs. However, the second characteristic seems to be more important than the first one since it allows the CSU to make rational decisions to select cloud services. We derive the closed-form expression for the trust-level auditability of CSP using the second type of characteristics by applying the following theorem:

**Theorem 1:** *In our proposed fuzzy-logic approach (as discussed in Section 3), CSUs provide the crisp data as an input that can be mapped to a fuzzy set during the Fuzzification process. For this Theorem 1, we refer to this crisp data as sample input data provided by a CSU. We sample input data for an audit if all the samples given to the auditor are correctly aligned with the service level agreement (SLA). If so, then the CSU is considered a reliable party*.

**Proof:** First, let us prove that the chosen samples of CSUs given to auditors for checking the trust-level of a CSP are correctly aligned with the SLA. This implies that if the given samples of services are properly stored on the server of the CSP, the confirmation process of samples versus original service according to SLA can be expressed as follows:

$$T_p \leftarrow e(\Delta\beta, \Delta\gamma) \in C_u \tag{8}$$

In (8), CSUs give the sample of services (few attributes of used service) to an auditor to verify the SLA.

$$V = \prod_{i=1}^{n} e(t_i, U) \cong e\left(\prod_{i \in Q}^{n} i(C_u) \in e(\Delta\beta, \Delta\gamma)\right) \tag{9}$$

In (9), the total number of CSPs and its provided cloud services to CSUs according to SLA, including the identity of CSUs, is considered.

$$V = \prod_{i=1}^{n} e(t_i, U).\sum_{j=1}^{\infty} e(\Delta\partial)^k \forall \underset{\approx}{\subseteq} e\left(\prod_{i \in Q}^{n} i(C_u) \in e(\Delta\beta, \Delta\gamma)\right) \tag{10}$$

Equation (10) shows the matching process, in which a CSP checks the provided services to CSUs, and the auditor compares the services with the provided samples.

$$V = \prod_{i=1}^{n} e(t_i, U) \quad \approx e\left(\prod_{j=0}^{n} i(C_u)\right) \tag{11}$$

Equation (11) confirms the matching process of samples with the original services given to CSUs. Based on our analytical model we can identify the trust-level of CSPs and its broader impact. Table 8 shows the used notions and their descriptions.

**Table 8** Notations and their definitions for trust-level validation of CSPs

| Notations | Definitions |
|---|---|
| $T_p$ | Auditor that gets a sample from CSU |
| $C_u$ | CSU gives the sample to validate the trust of CSP |
| $\Delta\beta$ | Service level agreement done between CSUs and CSP |
| $\Delta\gamma$ | Number of samples to examine |
| $e$ | Checkup process |
| $t_i$ | Total services |
| $k$ | Number of CSUs |
| $U$ | Total number of CSUs on the server |
| $V$ | Validation of CSUs and provided services |
| $\Delta\partial$ | CSP collects the money for this service |

### Results and discussion

We analyze the performance of the proposed fuzzy logic for CSPs using the fuzzy inference system to show that the CSPs are trustworthy. We focus on the auditability, access control, and encryption in this experiment. The experiment involves the MATLAB version 8.5 with an *Intel Core* i3 Processor running at 2.40GHz with 4GB RAM. We used four physical servers. Two servers are set for the CSP which provide the web service and database service. The third server works as the third-party auditor (TPA) and the fourth server works as the CSUs. We consider the data collection that is significant for the fuzzy-logic model. Thus, the fuzzy-logic model with fuzzy inference features should be trained using training data to specify the greatest possibility for obtaining the required results. The used data was collected using cloud users and cloud experts.

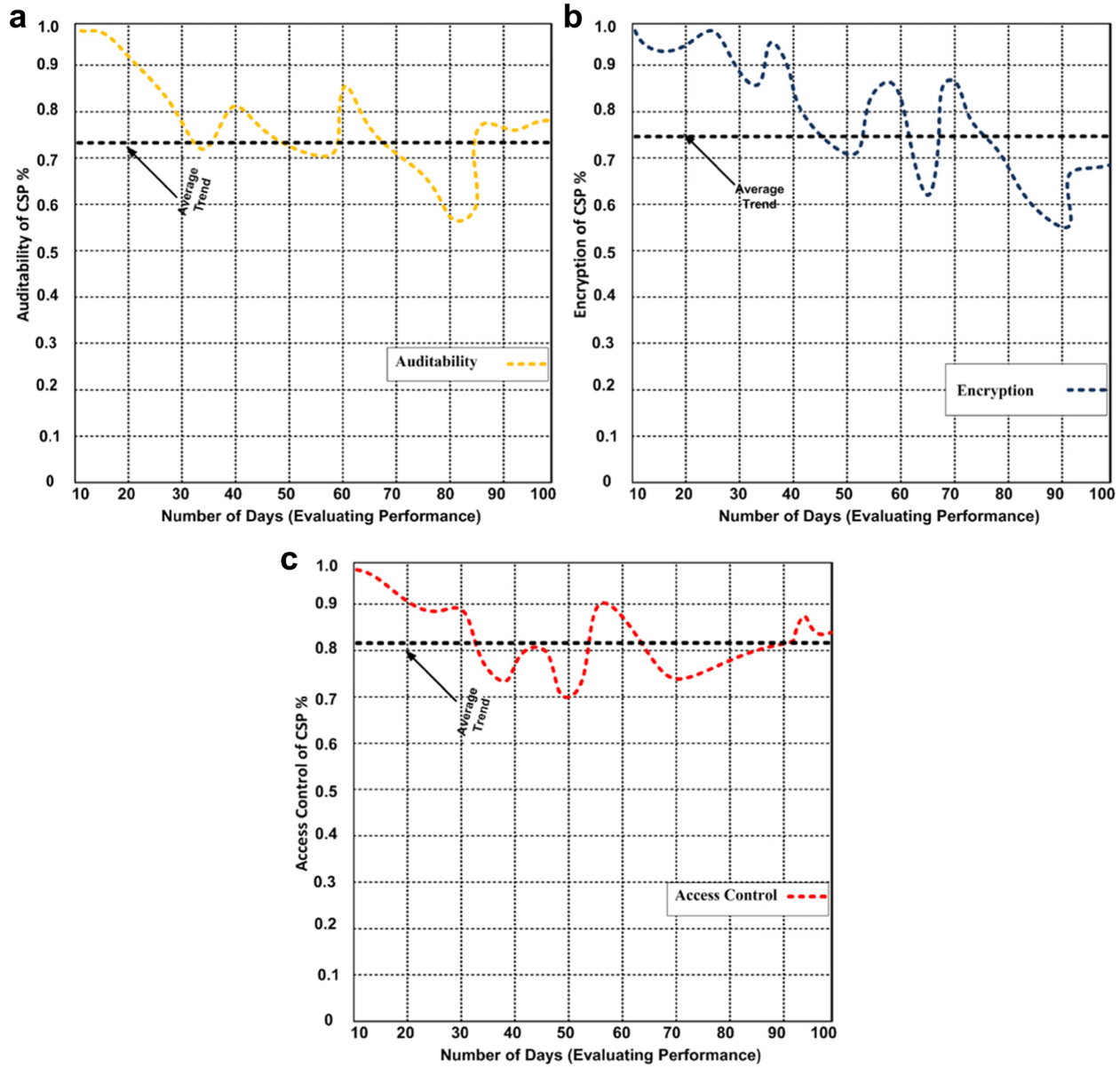


**Fig. 7** **a** showing the auditability; **b** the encryption and **c** access control of CSPs on different periods of times

We used the Zoomerang online survey tool to collect data sets from different sites. The survey consisted of the research questions showing the values for the most significant variables which were chosen to represent the access control, encryption, and auditability. The proposed fuzzy-logic model uses different input fuzzy sets and five fuzzy sets for the parameters of output (e.g., harmful, risky, average, great, and exceptional). After setting up the input and output fuzzy sets, the first step in the simulation is to focus on the Fuzzification to be converted to input membership functions. This process is done by applying the membership function editor available in MATLAB. Each variable used in the experiment is quantified into harmful, risky, average, great, and exceptional for auditability, access control, and encryption.

In Fig. 7, auditability, encryption, and access control are detected to ensure the trust-level of the CSPs. The performance of these three key variables remains unstable during the different time periods (days). The performance initially is satisfactory from day ten to twenty, but subsequently the performance is affected for all three key variables (encryption, authorization, and access control). The reason for the decline of performance could be the technical side including the obsolete programming languages, poor design and architecture, and the lack of knowledge and technology support by CSPs.

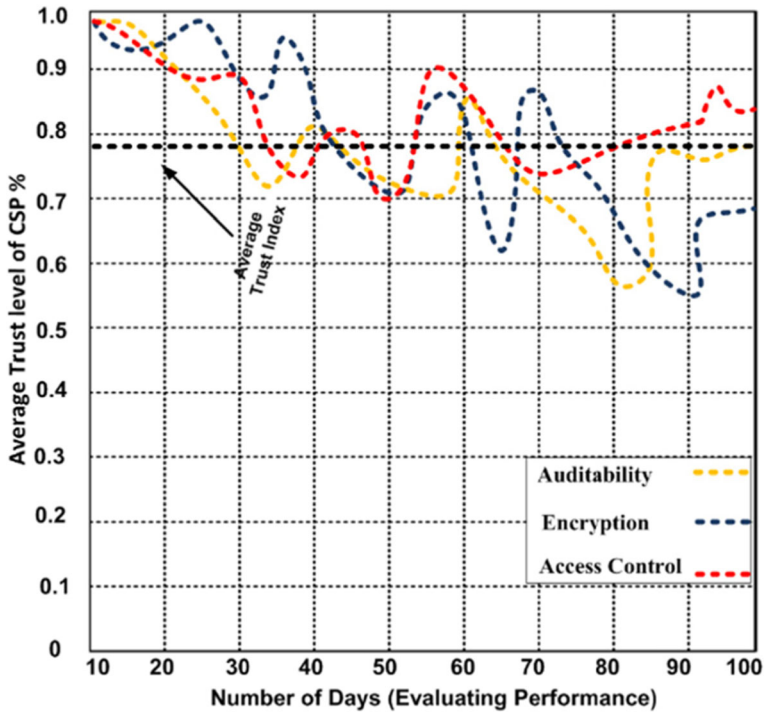


**Fig. 8** Trust-level of CSPs based on the three key variables: auditability, encryption, and access control

In these experiments, we assigned the weightage for these values to determine the realistic behavior of CSP. In Fig. 7, 7, and 7a, b, c, we have the average trends of each key variable. The lines in Fig. 7, 7, 7a, b, c visualize the fluctuation between auditability, encryption, and access control. The black dotted line allows the reader to see the average trend for each key variable and make a comparison between the variable and its average value. We observed that access control has an 81.8% maximum average trend that is more than other key variables. An average trend validates that the access control of the CSPs remains better than authentication and encryption. However, the combined performance of all three key variables for identifying the trust-level of CSPs is decreased, which is depicted in Fig. 8. We noticed that the combined trust-level of CSPs was around 77.1% during the evaluation period.

## Conclusion

In this paper, we presented a fuzzy-logic approach for evaluating the security of a CSP. Our approach addresses uncertainties associated with measuring trust, which is significant because the goal of security evaluation is to restore the confidence of the CSU. It is essential to maintain trust between the CSU and CSP to facilitate the more widespread adoption of cloud computing and similar distributed systems. Our approach is intuitive and easy to understand so that it can be used for a variety of purposes. In the future, to get an even more concise evaluation of a CSP by evaluating the sub-factors of security, a CSU can choose the most trustworthy CSP. Furthermore, our fuzzy approach can be used by the CSP for performing a self-evaluation and identifying any areas of risk that might require improvement. The validity of the fuzzy-logic approach is confirmed through case studies where real-life scenarios are tested. We hope to extend our work in the future by including the sub-factors of the evaluation factors used in our fuzzy-logic approach. This would improve the accuracy and precision of the security index. In addition, we anticipate many new factors besides security, which could be used to determine the most trustworthy CSP. Moreover, we want to implement a weighting system that can be included in our fuzzy-logic approach. The ability to apply weights to certain factors would emphasize those factors and allow CSUs to have their preferences make an impact on the security index. We plan to apply our proposed approach to case studies for specific CSPs for security evaluation. This is important for CSPs to establish trust and build a reputation to garner more CSUs. Our fuzzy-logic model can also be accompanied with machine learning in order to learn techniques to make an accurate assessment of a CSP that would be favoring to the CSU. This would allow CSPs to develop an even more efficient way of catering to their CSUs.

**Acknowledgements**
Not applicable.

**Authors' contributions**
All authors have equal contribution. Specifically, Dr. Rizvi has worked on the first four sections of the paper with three other authors (John Mitchell, Mohammad R. Rizvi and Iyonna Williams). Dr. Razaque has worked on Section 5 to validate the trust level of service providers using extensive simulation and numerical analysis. All authors read and approved the final manuscript.

**Funding**
This research project was not funded.

**Availability of data and materials**
All findings of this research work are presented in this article.

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]Department of Information Sciences and Technology, Pennsylvania State University, Altoona, PA 16601, USA. [2]Department of Computer Science, New York Institute of Technology, New York, NY, USA. [3]PricewaterhouseCoopers (PWC), Dallas, TX, USA.

Received: 30 November 2019 Accepted: 15 July 2020
Published online: 29 July 2020

**References**

1. Stergiou C, Psannis KE, Kim B-G, Gupta B (Jan. 2018) Secure integration of IoT and cloud computing. Futur Gener Comput Syst 78:964–975
2. Hayat B, Kim KH, Kim K-I (2017) A study on fuzzy logic based cloud computing. Clust Comput 21(1):589–603
3. Wang Y, Wen J, Wang X, Tao B, Zhou W (Jan. 2019) A cloud service trust evaluation model based on combining weights and gray correlation analysis. Security Commun Net 2019:1–11
4. van der Werff L, Real C, Lynn T (2018) Individual trust and the internet. In: Searle R, Nienaber A, Sitkin SB, editors. Trust. Oxford: Routledge. Available: http://doras.dcu.ie/22321/1/Individual_Trust_and_the_Internet_Repository.pdf
5. Söllner, M.; Pavlou, P. & Leimeister, J. M. (2016): Understanding the development of trust: comparing trust in the IT artifact and Trust in the Provider. In: Academy of management annual meeting (AOM), Anaheim, CA, USA
6. Mcknight DH, Carter M, Thatcher JB, Clay PF (2011) Trust in a specific technology: an investigation of its components and measures. ACM Transact Manag Inform Syst 2(2):12
7. Li YM, Yeh YS (2010) Increasing trust in mobile commerce through design aesthetics. Comput Hum Behav 26(4):673–684
8. Jensen, T., Khan, M. M. H., Albayram, Y., Fahim, M. A. A., Buck, R., & Coman, E. (2019). Anticipated emotions in initial trust evaluations of a drone system based on performance and process information. International Journal of human-computer interaction, 1-10
9. A. M. Mohammed, E. I. Morsy, and F. A. Omara, "Trust model for cloud service consumers," 2018 *International Conference on Innovative Trends in Computer Engineering (ITCE)*, 2018
10. Prabu Ragavendiran, S. D, Sowmiya N, Santhiya P. "Analysis of Trust Score of CSPS by Comparing Service Broker Policies and Load Balancing Policies using Cloud Analyst and Fuzzy Inference System", *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) RTICCT* – 2019 (Volume 7 – Issue 01)
11. A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," 2019 Amity International conference on artificial intelligence (AICAI), Apr. 2019
12. N. Gregorio, J. Mathanamohan, Q. H. Mahmoud, and M. Altaei, "Hacking in the cloud," *Internet Technology Letters*, vol. 2, no. 1, 2018
13. Cayirci E, Oliveira ASD (2018) Modelling trust and risk for cloud services. J Cloud Comput 7(1)
14. E. G. Abdallah, M. Zulkernine, Y. X. Gu, and C. Liem, "TRUST-CAP: A Trust Model for Cloud-Based Applications," 2017 IEEE 41st annual computer software and applications conference (COMPSAC), Jul. 2017
15. S. R. Rathi and V. K. Kolekar, "Trust Model for Computing Security of Cloud," 2018 Fourth international conference on computing communication control and automation (ICCUBEA), Aug. 2018
16. Mitchell J, Rizvi S, Ryoo J (2015) "A Fuzzy-Logic Approach for Evaluating a Cloud Service Provider," 2015 1st International Conference on Software Security and Assurance (ICSSA), Suwon. p. 19-24 https://doi.org/10.1109/ICSSA.2015.014
17. Srivastava R, Daniel AK (2018) Efficient model of cloud trustworthiness for selecting services using fuzzy logic. Adv Intelligent Syst Comput Emerg Technol Data Mining Inform Security:249–260
18. S. R. Rathi and V. K. Kolekar, "Trust Model for Computing Security of Cloud," 2018 *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018
19. B. Farrokhi and S. Nalbandian, "A Survey on Fuzzy Trust Management in Cloud Computing," 2018 *1st International Conference on Advanced Research in Engineering Sciences (ARES)*, 2018
20. Y. Wang, J. Wen, W. Zhou, and F. Luo, "A Novel Dynamic Cloud Service Trust Evaluation Model in Cloud Computing," 2018 *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018
21. S. Rizvi, J. Ryoo, J. Kissell, and W. Aiken. A Stakeholder-Oriented Assessment Index for Cloud Security Auditing. *The 9th ACM International Conference on Ubiquitous Information Management and Communication*
22. Alruwaythi M, Kambhampaty K, Nygard KE (2019) User Behavior and Trust Evaluation in Cloud Computing. Proceedings of 34th International Conference on Computers and Their Applications, vol 58. p.378-386
23. Chen Z, Tian L, Lin C (2018) Trust evaluation model of cloud user based on behavior data. Int J Distributed Sensor Netw 14(5)
24. van der Werff L, Fox G, Masevic I, et al (2019) Building consumer trust in the cloud: an experimental analysis of the cloud trust label approach. J Cloud Comp 8:6 https://doi.org/10.1186/s13677-019-0129-8
25. Xu J, Du X, Cai W, Zhu C, Chen Y (2019) MeURep: A novel user reputation calculation approach in personalized cloud services. Plos One 14(6)
26. Lu Y, Fang Y, Qin J (2019) A trust assessment model based on recommendation and dynamic self-adaptive in cloud service. J Phys Conf Ser 1325:012007
27. Huang C, He L, Liao X, Dai H, Ji M (2016) Developing a trustworthy computing framework for clouds. Int J Embed Syst 8(1):59–68
28. Kurdi H, Alfaries A, Al-Anazi A, Alkharji S, Addegaither M, Altoaimy L, Ahmed SH (2019) A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments. J Supercomput 75(7):3534–3554
29. F. Topaloglu and H. Pehlivan, "Comparison of Mamdani type and Sugeno type fuzzy inference systems in wind power plant installations," 2018 *6th International Symposium on Digital Forensic and Security (ISDFS)*, 2018
30. Hamdaouy AE, Salhi I, Belattar A, Doubabi S (2017) Takagi–Sugeno fuzzy modeling for three-phase micro hydropower plant prototype. Int J Hydrog Energy 42(28):17782–17792
31. A. Ebrahimnejad and J. L. Verdegay, "Fuzzy Set Theory," in Fuzzy Sets-Based Methods and Techniques for Modern Analytics, vol. 364, pp. 1–26
32. Thaker S, Nagori V (2018) Analysis of Fuzzification process in fuzzy expert system. Proced Comput Sci 132:1308–1316
33. Vimercati SDCD, Foresti S, Livraga G, Piuri V, Samarati P (2019) A fuzzy-based brokering service for cloud plan selection. IEEE Syst J:1–9
34. Chakraverty S, Sahoo DM, Mahato NR (2019) Defuzzification. Springer:117–127

## Publisher's Note

# Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com