A FUZZY LOGIC APPROACH FOR TRUST EVALUATION IN CLOUD SERVICE PROVIDERS

R. Devi[1], Logeshwaran[2], Sedhu Ram[3], Sridhar[4], Tamizharasu[5]

Abstract

Cloud computing offers an alternative way of accessing virtualized computing environments that allows industries like IT and healthcare to reach much lower costs than traditional computing paradigms. While many benefits of cloud computing exist, the extensive use of cloud services is usually constrained by the mistrust of cloud service users (CSUs) towards cloud service providers (CSPs). The multitude of trust models presented makes it all the more crucial to employ an ordered evaluation system based on a complete set of criteria for rating cloud services to allow users to choose the proper service provider to satisfy their needs. This paper finds a suggested fuzzy logic-based trust evaluation model specifically designed to rate CSPs properly. The suggested model will improve the effectiveness of trust assessment using fuzzy logic included in the quality of service (QoS) dimensions of cloud computing. The QoS dimensions include features like security, privacy, dynamicity, data integrity, and performance. The suggested approach was simulated in a series of simulations in a MATLAB environment, which demonstrated the proposed approach to be viable in any cloud environment.

Keywords

Fuzzy logic, Cloud computing, Trust model, Cloud service provider, Cloud security, Trust parameters, Trust score, Quality of Service (QoS), Trust Evaluation.

Introduction

Cloud computing transformed the perception of technology. Rather than investing heavily in hardware and doing everything in-house, users can now consume computing resources and services over the Internet on demand and only pay for what they consume. This convenience has made cloud computing a hit with businesses, developers, and even consumers. But with this convenience comes a gargantuan challenge: trust. When users entrust their data and processes to third-party cloud service providers (CSPs), they must be assured that the promised services will be delivered reliably, securely, and as expected. That's where trust enters the picture [1]. Cloud computing trust is not necessarily how good a service performs; it's how good users trust the provider to protect their data, perform, and deliver on their service-level agreements. Finding that usually means examining the provider's quality of service (QoS). Uptime, throughput, data privacy, system reliability, and responsiveness are all important. Ideally, the QoS guaranteed in the Service Level Agreement (SLA) should be equivalent to the actual-world performance users see [2]. But that's not always so easy to measure, especially when different users have different expectations. Nowadays, organizations like Amazon and Google have applied trust systems based on reputation scores and customer reviews to facilitate trust building. Other sites, such as eBay, have used centralized systems for years to assist users in judging reliability. Yet there is no one-size-fits-all method of computing trust among various cloud service providers, and that is where this research attempts to make a difference. In this research, we introduce a trust assessment model based on fuzzy logic. This model does not exclusively depend on hard numbers; rather, it is capable of handling the uncertainty and subjectivity that usually come with trust-related judgments. By considering various QoS factors such as security, privacy, performance, dynamic behavior, and data integrity, we compute a trust score that enables users to make more informed decisions [3]. What distinguishes this model is that it is highly flexible. It can be applied to a wide range of services and providers, and it does not require users to be highly technical. Instead, it takes fuzzy, real-world concerns and converts them into quantifiable data through a rule-based inference system. The end goal is to develop a model that is both precise and pragmatic, one that users and organizations can rely on to make the right cloud services decisions for their needs.

This research paper is organized as follows: Section "Literature survey" presents the existing research on trust models in cloud computing. Section "Proposed methodology" presents the proposed fuzzy-

based trust model for cloud service providers. Following that, the result and discussion are presented in Section "Results and discussion". Finally, Section "Conclusion and future work" presents the conclusion and future work.

Literature survey

In a recent research paper, a fuzzy logic-based model is introduced to evaluate the credibility of cloud service providers (CSPs) employing a complete set of quality of service (QoS) parameters like security, privacy, performance, dynamicity, and data integrity. Recognizing trust as an important concern in cloud computing, the model employs a rule-based fuzzy inference system to integrate objective quantifications and subjective opinions into one measure of trust. By comparing real-time QoS performance with the agreed parameters in Service Level Agreements (SLAs), the model provides an open and trustworthy framework for selecting reliable service providers. Simulation results using MATLAB demonstrate the accuracy of the model, indicating its potential to facilitate improved decision-making, risk minimization, and increased user trust in cloud environments [1]. A systematic framework of trust evaluation for Cloud Service Providers (CSPs) has been proposed in the guise of a Digital Twin, which is a virtual representation of the CSP infrastructure. The process enables simulated vulnerability testing on the digital twin, the result normalized and subsequently processed by a Fuzzy Inference System (FIS) to derive a trust score. The model has continuous improvement with the addition of user feedback and historical performance data, thereby facilitating a dynamic and comprehensive assessment of trustworthiness for cloud service consumers [2]. A trust model based on fuzzy logic is proposed to assess the security preparedness of Cloud Service Providers (CSPs). The model makes use of subjective trust metrics and a Security Index to gauge the overall security position of every provider. The approach allows Cloud Service Users (CSUs) to rank and compare CSPs precisely even in situations of uncertainty and incomplete knowledge [3]. In the work "Trust Evaluation Models for Cloud Computing," Damera et al. offer an exhaustive survey of trust management frameworks currently used in cloud computing. The authors group trust evaluation models into four categories of broad types: agreement-based models, where trust is developed through Service Level Agreements (SLAs) and service policy; certificate-based models, which utilize security certificates and trusted platform modules; feedback-based models, which utilize customer ratings and reputation mechanisms; and domain-based models, which structurize the cloud into decentralized trust domains. The work further explains major trust features—like subjectivity, dynamic behavior, and context-dependency—and analyses their impact on cloud service selection and trust building [4]. The CSTEM model combines direct trust, recommendation trust, and reputation to assess cloud service trustworthiness. Rough set theory and Analytic Hierarchy Process (AHP) are employed to compute objective and subjective weights, and gray correlation analysis is employed for recommendation trust. A dynamic trust update mechanism guarantees accuracy and immunity to malicious behavior [5]. Hierarchical modeling is proposed to assess cloud service credibility with the combination of fuzzy entropy and a Markov chain. The approach quantifies trust and predicts its future growth. It identifies major risk factors and offers a dynamic, objective framework of trust assessment [6]. The paper provides a multi-criteria CSP selection model in the form of utilizing a Dual Membership Function-based Fuzzy Logic Technique (DMFT) method. It assesses providers on five QoS dimensions: cost, capacity, performance, security, and maintenance. User needs are converted into linguistic variables and calculated through a Mamdani fuzzy inference system to rank CSPs according to service features and user priorities [7]. This paper proposes a probabilistic trust model based on a Bayesian network for cloud services to handle uncertainty in QoS attributes and interdependency. QoS attributes are modeled as random variables to enable more accurate trust estimation. The model is evaluated using the CloudArmor dataset with improved accuracy than the traditional deterministic and regression-based methods [8]. The paper proposes a trust rating system that delivers performance and cost-based trust ratings of CSPs using fuzzy logic. It resolves the issue of selecting trustworthy CSPs with the growing number of providers and fewer trust rating mechanisms. The solution integrates Cloud-Analyst simulation with Mamdani fuzzy logic in MATLAB to deliver impartial trust ratings [9]. This work suggests a trust-based method to evaluate the security of cloud computing environments through a multi-criteria trust score mechanism. It measures CSPs in terms of security, reliability, performance, compliance, and customer satisfaction. Quantitative comparison and ranking are facilitated through the framework, allowing users

to select reliable providers and manage the associated risks [10]. The article introduces a fuzzy logic cloud broker system that assigns users to cloud instances of service based on user needs (e.g., cost, task size) and provider resources (e.g., CPU, price). It categorizes users and VMs into Gold, Silver, or Bronze classes by employing fuzzy logic to offer best fit. The system supports static and mobile users with service migration to offer QoS and cost-saving [11]. The study of different trust models are shown in Table 1 and the summary of Literature review is shown in Table 2.

Trust Evaluation Model	Key Characteristics	Strengths	Weaknesses
Agreement-Based [1]	Depends on formal contracts or agreements to verify compliance	Articulates trust expectations; rooted in legal or contractual terms	Restricted adaptability in dynamic or unpredictable conditions
Certificate-Based [2]	Uses digital certificates from a trusted authority to determine identity	Strong authentication mechanism; based on highly reliable security standards	Exposed to compromised issuing authorities; complicated maintenance
User Review-Based [2]	Builds trust through user feedback and ratings based on experience	Reflects the real quality of services; responsive to changing performance	Prone to fake or biased inputs; unlikely to work effectively at scale
Context-Aware [3]	Determines trustworthiness in a particular use context	Provides relevant evaluations in a specific context	Difficult to transfer to other domains; strict contextual factors
Multi-Criteria Decision-Based [3]	Measures trust by an assortment of factors and analytical procedures	Evaluation is not exhaustive; can be modified for evolving systems	Challenging to determine factor weighting; may be resource heavy
Optimization-Based [4]	Implements optimization algorithms to maximize trust and improve performance of the system	Useful in dynamic contexts; strives for optimal trust results	Depends on the quality of the algorithm; initial input can impact
Predictive Model [4]	Predicts future trustworthiness based on past behaviors and data-driven practices	Anticipates future activity; easily fits with the evolution of systems	Highly dependent on data accuracy; the model may be interpreted in complicated ways
Recommendation-Based [5]	Generates suggestions on trustworthy entities by similarity analysis or collaborative processes	Takes advantage of common understandings; facilitates informed choices	No data exists for newcomers (cold start); potential for historical bias
Reputation-Based [5]	Accumulated feedback and past actions build trust scores over time	Identifies consistent and reliable behaviours; focusing on long-term trends	Prone to manipulation; sometimes hard to derive errors from intent

Table 1. Study of different trust models in cloud

References	Utilized Techniques	Evaluation Tools Used	Performance Metrics	Merits	Demerits
Jomina John, K. John Singh[1]	Mamdani-type fuzzy inference system, triangular membership functions, centroid defuzzification, rule-based trust scoring, parameter hierarchy based on QoS.	MATLAB and Simulink were used for parameter mapping and model simulations; CloudSim was used for cloud service simulations.	All of the metrics include accuracy, precision, recall, F1-score, average execution time (fuzzification, inference, defuzzification), and performance scalability using a range of CSPs and parameter settings.	Broad parameter coverage, adaptable across CSPs, interpretable trust scores, handles uncertainty well, and shows high accuracy through simulations.	Large input sizes are computationally expensive, have a steep learning curve for expertise in rules/membership functions, are limited in terms of real-time adaptability, and also depends upon consistent and traceable QoS values.
Jomina John, K. John Singh[2]	A unique approach integrating digital twin modelling with fuzzy logic-enabled trust evaluation. Featured penetration testing (e.g., Sybil, collusion attacks), parameter normalization, triangular fuzzification, and rule processing. Afforded the option of using linear regression or ANOVA to normalize to labeled trust scores and assess parameter impacts on scores.	MATLAB Simulation Toolbox was applied to model fuzzy logic and digital twins. ESA Control Toolbox was used to simulate system behavior. AVISPA was used to validate the security protocol and to assess attacks and vulnerabilities.	The model achieves competitive benchmarks with an accuracy of 95.2%, 94.5% precision, 96.0% recall, and F1-score of 95.3%. The average time it takes to compute trust or score is 150 milliseconds, confirming the model's efficiency for a realistic setting and near-real-time scenarios.	Supports dynamic and on-going trust evaluation with continuous feedback integration. Flattens digital twin visualization with fuzzy analytics for clarity. Offers better performance over traditional models.	Dependent upon real-time and accurate input data from tests and measurements. Scalability can be limited due to complexity of computation. Need knowledge of fuzzy system setup and tuning; interpretability limited for some casual users.
Syed Rizvi, John Mitchell, Abdul Razaque, Mohammad R. Rizvi and Iyonna Williams[3]	Used a Mamdani-type Fuzzy Inference System (FIS) with triangular membership functions to map linguistic terms (e.g., risky, great) to numeric values. Used over 20 fuzzy rules with centroid defuzzification.	Used MATLAB 8.5 to simulate and model fuzzy logic. Collected expert and user information through Zoomerang surveys. Performed case studies with physical servers to recreate CSPs and CSUs.	The computed security index (SI) used defuzzification (SI=90 for “Exceptional”). The reported trust levels were auditability (77.1%), encryption (81.8%) and access control (81.8%).	Handles uncertainty and vagueness well, with fuzzy logic. Extensible, supports customization, user-friendly interface with linguistic inputs. Supports CSP ranking and comparisons.	Factors are only top-level factors with missing details. Only works with discrete and static functions that may not cover all perspectives of users. Also relies upon subjective input and hypothetical CSPs.
Vijay Kumar Damera, A Nagesh, M Nagaratna[4]	Incorporated various trust models: SLA-based negotiations and monitoring, certificate-based trust using an	Used Netlogger for SLA tracking, TPM and PKI for all certificate validations, and Cloud Armor, RATEWeb, and	Measures include compliance with SLA parameters,	Agreement based models have formal, enforceable policies; certificate based systems guarantee	Agreement-based models are not very dynamic; certificate-based systems require

	PKI with TPM trust anchor, feedback-based reputation systems using Dempster-Shafer theory of evidence, and domain-based hierarchical models.	TAAS as frameworks for feedback-based models. Domain-based methods, used trust tables, for the distributed management.	accuracy of the trust score, scalability of the model, and detection of malicious feedback/user, efficiency in propagating trust scores, and agreement violation detection.	strict identity verification. Feedback models are adaptable; domain-based systems scale and computationally efficient.	action from central authorities for revocation issues. Feedback-based systems can be manipulated; domain-based approaches have empirically-proven cross-domain issues.
Yubiao Wang, Junhao Wen, Xibin Wang, Bamei Tao, Wei Zhou[5]	Rough Set Theory computes objective weights for trust attributes; AHP relies on subjective weights based on user preference; Gray Correlation Analysis computes derived trust for recommendation purposes; Dynamic Trust Update considers time decay, transaction value, and penalties for failed or deceitful transactions.	CloudSim 3.0.3 is the simulation platform that models the cloud environment; additional custom variables are added to CloudSim to represent transaction price, execution time, user satisfaction, and penalty factors to capture trust-control dynamics in service interactions.	The evaluation is based on customer satisfaction via user feedback, the success rate of interactions (the rate at which transactions were successful with a satisfaction level of 0.7 and above), and resilience using numbers of malicious nodes ranging from (10% – 40%) to examine the robustness of the trust model.	The model integrates subjective and objective trust weights; dynamically penalizes malfeasance, gives increased weight for recently completed transactions and higher transaction value; shows better performance and satisfaction to a baseline distance model of better than 90%, better than 85% success, and robust to many malicious entity.	Overall the approach presents a high level of computational complexity due to the use of multiple different techniques, is limited in terms of scalability testing (up to 300 nodes), is limited in scope to static cloud derivatives and does not explore mobile and heterogeneous environments, and has not dealt with other dynamic factors or challenges of real-world implementation.
Ming Yang, Rong Jiang, JiaWang, BinGui, Leijin Long[6]	Utilizes a hierarchical framework consisting of 3 classes and 16 indicators. Fuzzy entropy is used to quantify uncertainty and volatility within the system, Markov chains are used to predict the movements of trust transitions, and a risk matrix correlates threats severity to trust states.	The Evolving Community Networks platform, ECS, for evaluating the framework is achieved through a case study utilizing expert ratings. The framework is compared to the entropy model and AHP modeling method, as well as Dempster and Shafer's evidence theory, and a risk evaluation matrix. Comparative evaluations are based on fuzzy entropy values and state transition matrices.	The decision support tool features fuzzy entropy, multiple transition probabilities, and outlines potential risk indicators. The decision support tool was benchmarked to ensure objectivity, cost, scalability, and decision support.	Offers predictive trust assessment, reduces subjectivity in decision making, provides effective integration of qualitative and quantitative data; supports risk-based decision making.	This system is computationally intensive, has moderate scalability, is highly reliant on experts, and the results can be difficult to interpret due to dependencies between models.
Mohammad Faiz, A. K. Daniel[7]	This study employs a membership function based Dual Membership Function based Fuzzy	The simulation is executed in MATLAB. A real-world CSP dataset (Sidhu & Singh, 2017) is used to validate the	Evaluated based on ranking accuracy, computational efficiency, and	Handles arbitrary scope for user preferences, supports dynamic weighting of QoS propositions that	The system has complex rules (1,024 rules), would have limited scalability for

	Technique (DMFT) utilizing trapezoidal and triangulated membership functions, as well as a Mamdani Fuzzy Inference System of more than 1,024 rules used for ranking of CSPs and is compared with AHP, ANP, and I-TOPSIS methods.	framework that includes CPU, RAM, cost, performance, security, and maintenance attributes, and is benchmarked against AHP, ANP, and I-TOPSIS.	flexibility with parameters. Correlates well to I-TOPSIS, prioritizes QoS dynamically, and ranks CSPs (i.e., C4) based individual user preferences.	are sometimes user-defined, and provides fairly strong performance with a high degree of accuracy while modelling for both runtime execution times as well as bandwidth utilization in comparison to traditional AHP/ANP methods in a multi-criteria environment.	additional parameters, has sensitivity to the input data's quality, and requires a fuzzy logic expert to assist in tuning the system and set up the rules.
Mihan Hosseinnezhad, Mohammad Abdollahi Azgomi, Mohammad Reza Ebrahimi Dishabi[8]	Within this study, discretization is achieved using WEKA's equal-frequency method, structure learning uses the K2 algorithm and parameter learning uses MLE while inference uses the Junction Tree algorithm, this is further compared to multiple linear regression.	The simulation utilizes the publicly available CloudArmor dataset (10,076 feedbacks for 113 services), WEKA for pre-processing data, and MATLAB BNT for implementation. The tests of the framework are simulated on a Core i5 system with 10-fold validation and hold-out validation.	Lower error rates than regression. Wilcoxon test shows there is no significant difference between predicted trust value and the actual trust value.	Accurately modelling uncertainty and dependencies among QoS propositions, explicitly considering problems related to incomplete data, and allows for significant predictions, accuracy compared to traditional regression models based on real-world datasets.	Computationally intensive and would require discretization of data that may not be accurate, scalability is unproven for larger or real-time systems, and would also highly depend on the quality of the input data.
Doaa Trabay, Azezza Asem, Hazem M. El Bakry, Ibrahim El-Henawy[9]	The Mamdani Fuzzy Logic-based Trust model is assessed using MATLAB Simulink. The inputs derived for evaluating trust were derived from a simulation of CSP performance using the Cloud-Analyst system including key delivery parameters such as VMs and data centers, costs and metrics related to resource delivery all categorized and categorized using fuzzy membership functions.	Cloud-Analyst is the application used to simulate CSP environments and collect metrics (response time and user costs), while MATLAB Simulink uses the outputs of Cloud-Analyst to implement the fuzzy inference system for computing trust values for individual service providers.	Metrics include a response time, processing time, VM and data transfer costs, and a trust value which is mapped to a qualitative rank (High, Medium, Low) based on fuzzy logic.	Provides a structured, numerical trust evaluation that includes both costs and performance, allows for uncertainty to be modelled through fuzzy logic and leaves room for extension with additional QoS proposals in the future.	This only considers cost and performance and excludes aspects such as security and usability, is sensitive to input, requires high quality simulations, and no process for analyzing real-time CSP behavior was identified.
Anand Kumar Mishra, Mayur Rahul, C.S. Raghuvanshi[10]	Trust is then calculated using a weighted average of the inputs such as security, reliability, performance, compliance and customer satisfaction where normalized datasets have used	The measured trust values (objectively assessed) are consistent because Cloud-Analyst implements the same weighted average model using normalized inputs for all metrics, attribute groups and CSP's own services. CSP's are	Trust metrics include security, reliability, performance, compliance, and customer satisfaction.	Provides a robust means of conducting a fair, flexible trust evaluation in terms of both technical-type factors (i.e. QoS) and user-type factors (i.e. non-technical proposals, such as user trust issues), in an	Some subjectivity in specifying weights, depending on simulated data and does not take into consideration evolving threats or real-time threats may affect robustness; and

	iterative feedback to improve trust scores level.	subjected to tabulated trust score comparisons based on simulated or collected data, with mechanisms to iterate on updating trust values and re-evaluate trust scores.		effort to continuously improve, while being robustly and reasonably tailored to the user's specific organizational business needs.	scalability issues can arise with agent based monitoring in large environments.

Table 2. Summary of Literature Review

Proposed Methodology

Fuzzy Logic-Based Trust Score Evaluation

In cloud computing, the choice of the best Cloud Service Provider (CSP) is highly crucial because of the high number of trust models and evaluation processes present. Such diversity calls for an ordered and sound evaluation mechanism tailored to various user requirements. Our proposed solution thus presents a fuzzy logic-based trust evaluation model to simplify the process of identifying the most trustworthy CSP. Human analysis-reliant traditional approaches may turn out to be ineffective and time-consuming. However, fuzzy logic can provide a better solution, especially when handling complex, nonlinear systems. Our system utilizes a rule-based fuzzy inference system with four essential components: (1) Fuzzifier—translating numerical (crisp) input values such as performance ratings or security scores to fuzzy sets; (2) Inference Engine—applying fuzzy reasoning over a given set of rules to obtain fuzzy output; (3) Defuzzifier—translating the fuzzy output back to a single crisp trust rating for ease of interpretation; and (4) Knowledge Base—which contains the rule base (if-then decision rules) as well as the database (membership functions defining fuzzy terms like "low," "medium," or "high"). The system allows cloud consumers to rate service providers based on a combination of reviews and performance measurements, increasing both accuracy and efficiency of trust calculation. The working of the fuzzy inference system is shown in Figure 1.

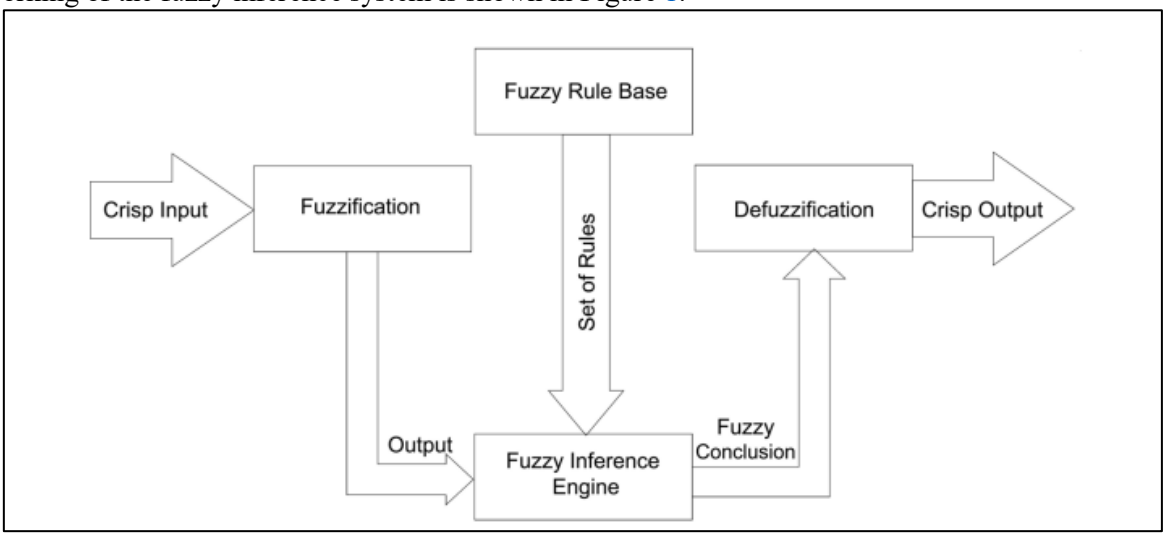

Figure 1. Working of Fuzzy Inference System

Basic Concepts in Fuzzy Logic

Fuzzy logic is a sophisticated computational technique that deals with approximate rather than fixed or exact reasoning. Fuzzy logic is especially useful in systems that are uncertain, like measuring trust and uncertainty with cloud service providers. The basic components of fuzzy logic include fuzzy sets, linguistic variables, membership functions, and a fuzzy inference mechanism. Below they are briefly described and relevant formulas provided:

A fuzzy set is defined with a membership function that assigns a degree of membership (0-1) to each element. A fuzzy set is outlined as follows in Eq. (1). Formally, a fuzzy set F in a universe of discourse U can be written as:

$$F = \{(x, \mu_F(x)) | x \in U\} \tag{1}$$

Where x is an element in the universe U, $\mu_F(x)$ is the membership function of fuzzy set F, giving a value in the range [0,1].

A linguistic variable is a variable whose values typically are not numbers but words or phrases. For instance, "Trust" could be characterized as {Very Low Trust, Low Trust, Medium Trust, High Trust, Very High Trust}. These concepts are evaluated using fuzzy sets. Linguistic Variables and its are depicted in Table 3.

Linguistic Variable	Membership Degree	Description
v_h	80–100	Extremely High
h_z	60–90	High Level
m_z	30–70	Moderate Level
l_z	10–40	Low level
v_l	0–25	Extremely Low

Table 3. Linguistic Variables for the Parameters and its ranges

A membership function defines how each point in the input space is mapped to a membership value between 0 and 1 as shown in Figure 2. A common shape used in fuzzy systems is the **triangular membership function**, defined as:

$$\mu(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \tag{2}$$

Where a *and* c are the lower and upper bounds of the triangle, b is the peak (maximum membership value), $\mu(x)$ represents the degree of membership of input x.

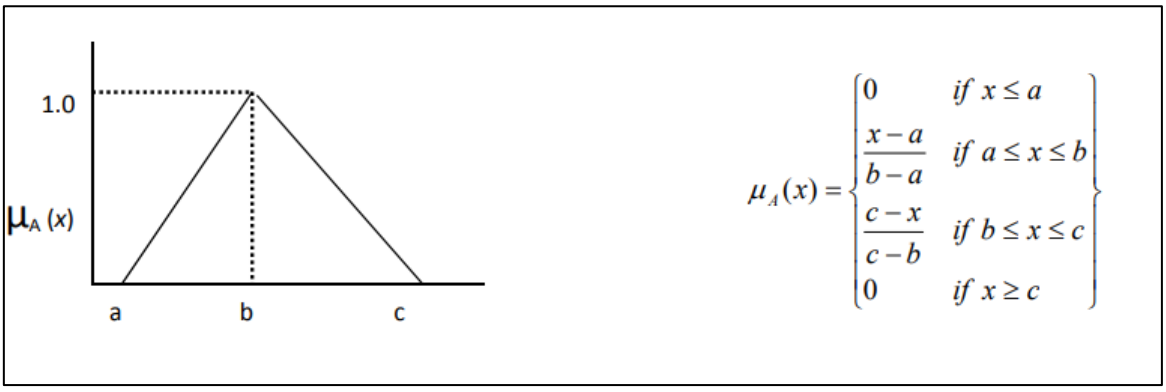

Figure 2. Triangular membership function.

Fuzzification is the transformation of crisp input values to fuzzy values through the use of membership functions. Fuzzification is the transformation of numerical inputs (e.g., security rating = 78) to fuzzy linguistic values (e.g., corresponding to High). The rule base is made up of fuzzy "if-then" rules, which relate the input conditions to some output decision. An example rule might be: IF Security is High AND Performance is Medium, THEN Trust is High. These rules are invoked using fuzzy logic operators, such

as MIN for AND, MAX for OR and complement for NOT. The inference engine performs the task of applying the rule base to the fuzzified inputs to calculate the fuzzy output. For example, using Mamdani inference, the firing strength of a rule can be calculated as:

$$\alpha = min(\mu_A(x), \mu_B(y)) \tag{3}$$

Where $\mu_A(x)\ and\ \mu_B(y)$ are the membership values for inputs x and y under conditions A and B. Defuzzification is the process of converting the fuzzy output into a crisp value. One widely used method is the **centroid (center of gravity)** method, which calculates the final crisp trust score as:

$$z = \frac{\sum_{i=1}^{n} \mu_i(z_i) \ . \ z_i}{\sum_{i=1}^{n} \mu_i(z_i)} \tag{4}$$

Where z_i are output values, $\mu_i(z_i)$ are their corresponding membership degrees. This formula calculates the weighted average of the output values, while taking into account how activated each rule is. By integrating these key concepts, fuzzy logic systems can assess trust in cloud services by dealing with uncertainty, integrating multiple aspects that focus on trust, and producing output scores that make decisions.

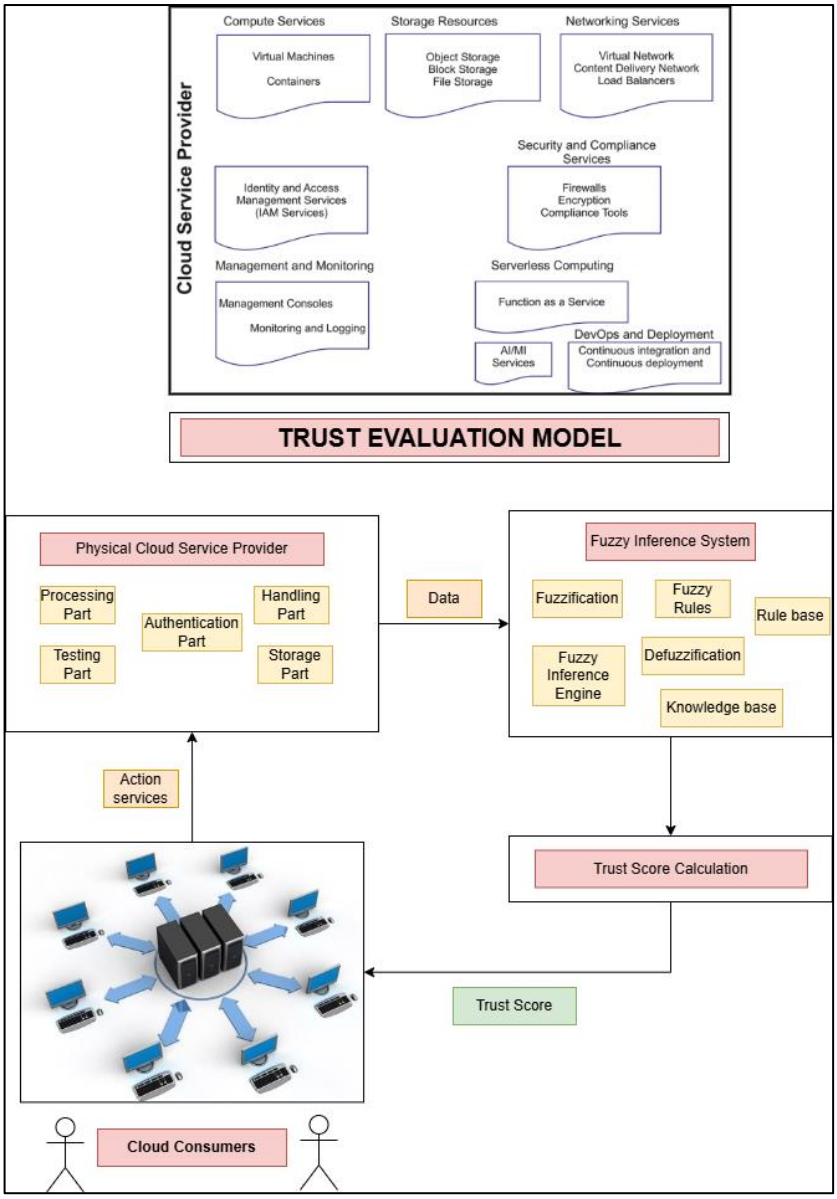

Figure 3. Overall Architecture diagram of Trust model

Parameter-Based Trust Score Simulation

The suggested trust assessment method employs a structured fuzzy inference system (FIS) to mimic trust scores based on an interdependent hierarchy of parameters. In the method, every factor related to trust—e.g., security, performance, or reliability—is defined with input variables that are fed into a fuzzy logic-based framework. The Fuzzy Logic Designer is initially set up to determine each parameter and its contribution to the overall trust calculation. The parameters are not assessed individually. Rather, they are calculated as a function of the output of their subordinate (sub-parameter) values, yielding a hierarchical tree-like structure. For instance, the parameter "Security" might be obtained from sub-parameters such as "Data Security," "Network Security," and "Physical Security." This multi-level structure improves the model's accuracy by reflecting the impact of fine-grained trust factors on the overall assessment. The parameters are chosen based on a thorough review of the literature, which recognizes the factors frequently used in previous trust models. These include technical properties such as performance, reliability, availability, and security, and threat-based inputs such as Sybil attacks, collusion attacks, and data breaches. Each parameter is allocated a linguistic variable (e.g., Low, Medium, High) and an associated numeric range that reflects the degree of trustworthiness. The trust parameters are showed in Table 4.

In order to ensure robust trust score estimation, raw data collected from system logs, vulnerability scans, feedback, and efficiency measurement must all be properly pre-processed. It begins with cleaning, where missing values and outliers are addressed to eliminate inconsistencies in the dataset. Normalization is then performed to scale all parameter values to a common range so that no input significantly overshadows the others in terms of influence on the outcome. Next is feature engineering, where new, valuable features are derived from the existing data to enhance the predictive capability of the model. Lastly, encoding is performed to transform categorical variables into numerical representations so that efficient processing can be achieved by the fuzzy logic system.

Trust Factor	**Linguistic Labels and Corresponding Ranges**
Sybil Threat	v_l (0, 10, 25), m_z (30, 50, 70), h_z (60, 75, 90)
Collusion Threat	v_l (0, 10, 25), m_z (30, 50, 70), h_z (60, 75, 90)
Threat Prevention	v_l (0, 10, 20), l_z (15, 25, 35), m_z (35, 50, 65), h_z (65, 80, 90), v_h (85, 95, 100)
Physical Protection	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Network Protection	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Data Protection	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Security Assistance	v_l (0, 10, 20), l_z (20, 25, 30), m_z (30, 50, 70), h_z (65, 80, 90), v_h (85, 95, 100)
Encryption Technique	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Comprehensive Security	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 55, 65), h_z (65, 80, 90), v_h (85, 95, 100)
Audit Capabilities	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Authorization Control	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Responsibility Tracking	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Data Privacy	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 55, 65), h_z (65, 80, 90), v_h (85, 95, 100)
Service Availability	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Service Reliability	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Service Flexibility	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Network Bandwidth	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Communication Delay	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
System Efficiency	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 55, 65), h_z (65, 80, 90), v_h (85, 95, 100)
User Friendliness	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Modular Architecture	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
System Compatibility	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)

Adaptability	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 55, 65), h_z (65, 80, 90), v_h (85, 95, 100)
Regulation Compliance	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Quality of Experience	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Technical Assistance	l_z (10, 25, 40), m_z (40, 55, 70), h_z (70, 85, 90)
Information Availability	l_z (10, 25, 40), m_z (40, 55, 65), h_z (65, 80, 90)
Data Integrity	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 55, 65), h_z (65, 80, 90), v_h (85, 95, 100)
Final Trust Rating	v_l (0, 10, 20), l_z (20, 30, 40), m_z (40, 50, 60), h_z (60, 80, 90), v_h (85, 95, 100)

Table 4. Trust parameter with its linguistic variables and its ranges

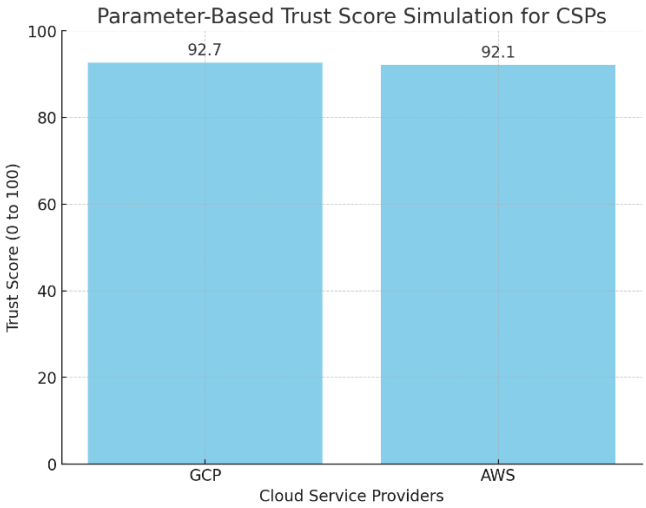

Figure 4. Parameter based trust score Simulation

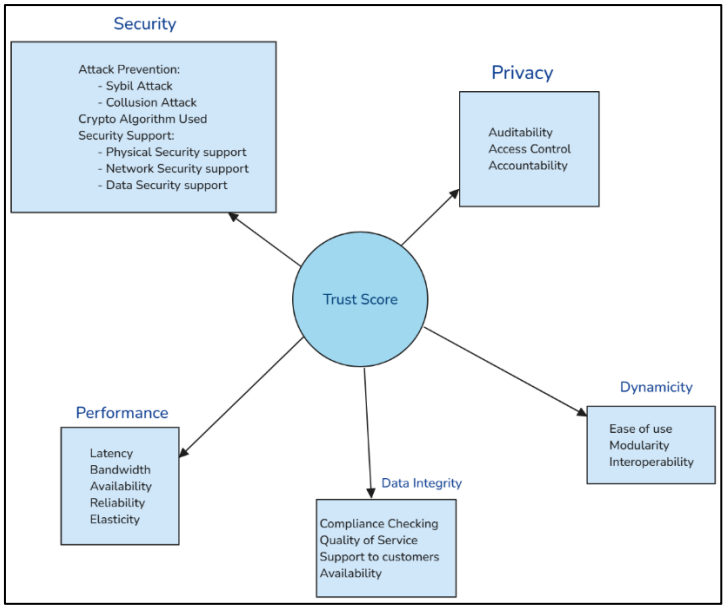

Figure 5. Trust Parameters for trust score evaluation

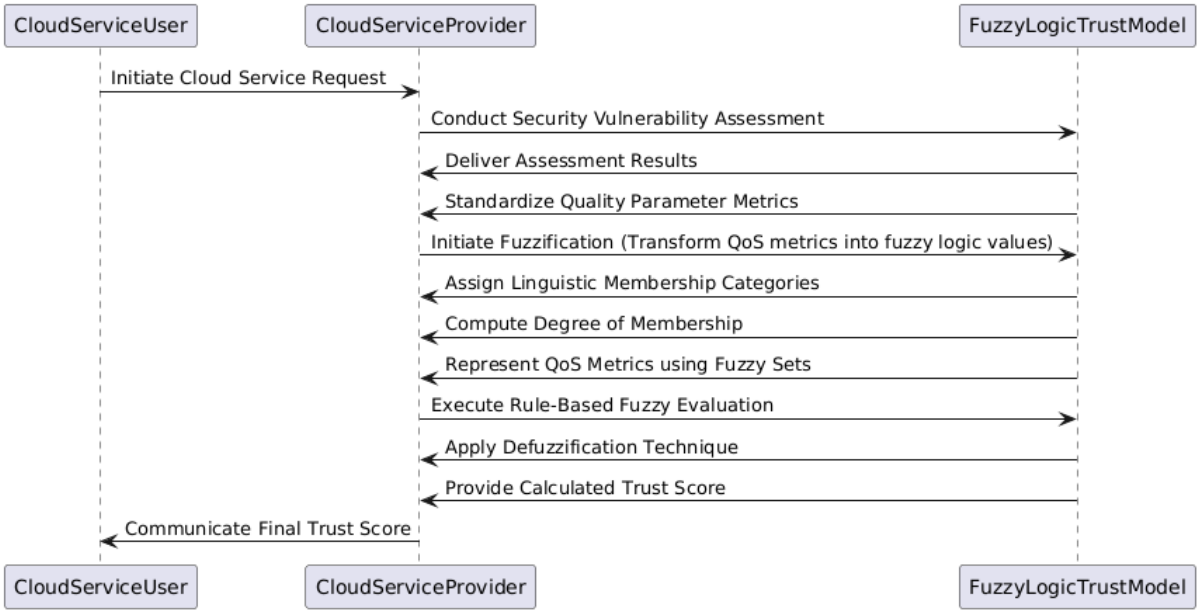

Figure 6. Sequence Diagram of Trust Evaluation Process

Trust Parameters Explanation

Security [1]

For data, services, and apps to be safe from cyberattacks, illegal access, and system failure, cloud security is essential. Network security, data encryption, identity and access management, and real-time monitoring are just a few of the numerous safeguards that are included in security. Because of their empathy, these controls ensure the integrity and confidentiality of cloud resources by identifying and fixing vulnerabilities before an attacker can exploit them.

Attack prevention [2]

In cloud computing, attack prevention is the establishment of security policies, procedures and practices to defend resources in the cloud from hackers and unauthorized access. Attack prevention measures include firewall implementations, as well as the utilization of intrusion detection and prevention systems (IDPS), encryption and multi-factor authentication (MFA), secure coding practices and continual monitoring of the system. Attack prevention methods strive to identify and eliminate potential threats, and are put into place to ensure that cloud services can withstand attacks both inside and outside the cloud - for instance, against Distributed Denial of Service (DDoS) attacks, data breaches, and malware. Other aspects of attack prevention include building trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs), while expositional between industries such as government, healthcare, and finance.

Sybil Attack [2]

In cloud computing, a Sybil attack is a type of attack by a malicious entity that creates multiple fake identities or nodes (Sybil nodes) in order to gain excessive influence in the network. The attack undermines the premise that every node is equal, unique and trustworthy, as it is part of a trust-based decision system. In distributed systems, this is a major problem that threatens the integrity of the system. The malicious entity can force changes to consensus mechanisms, can disrupt reputation-based models, can manipulate trust chains to intercept or modify data traffic, and so on — it can lead to data breaches and denial of service, as well as service disruptions from malicious attacks that were not distinguishable from the intended service. In cloud computing system Sybil attacks can impact cloud resource allocation, deny the access to genuine node to access cloud resources or users can use a Sybil attack to mask a real attack or to employ 'Sybil' (fake) identities to access the cloud through access denial. Detection techniques for Sybil attacks include: identity checks, social trust graph algorithms, and imposing economic costs, but there are still challenges for detection due to the requirements of cloud computing systems (the dynamic nature and/or loosely coupled systems).

Collision Attack [2]

A collision attack is a type of attack on cryptographic hash functions that occurs when an attacker is able to find two different inputs where the hash function produces the same hash output. Collision attacks will compromise both data integrity and data authenticity because digital signatures or checksums based on such hash functions could be created with two inputs that are unrelated. For instance, while there have been no known incidents of collision attacks on cloud computing, a collision attack on SHA-1 practically demonstrated that attackers could generate two distinct inputs that yield identical hash values leveraging cloud computing resources. Many systems are still using SHA-1 and thus, legitimate threats were created. Security experts generally recommend to stop using SHA-1 and transition to the more secure hash functions like SHA-256 or SHA-3. Collision attacks happen infrequently and it is generally safe to assume people are not in the business of carrying out these unique and complex attacks - until they are.

Crypto-Algorithm Used [3]

Since cryptographic algorithms encrypt data, they are essential to cloud security. Key size, attack resistance, performance efficiency, and industry adoption are used to evaluate their performance. Larger key sizes provide more protection, and attack resistance shields data from sophisticated attacks. Industry adoption measures the algorithm's credibility and trustworthiness, while performance measures how well it works in real-time systems. One common example is AES, which is widely used on cloud platforms and security protocols and offers high security and speed. The generalized rules are shown in table 5.

Algorithm	**Key Size**	**Score (10 - 110)**	**Reason for Score**
AES-256 (Advanced Encryption Standard)	256-bit	**110**	Strongest widely used encryption standard, used for at- rest and in-transit encryption across all cloud providers. Resistant to all known practical attacks. It is widely used today as it is much stronger than DES and triple DES despite being harder to implement. AES 256 is Unbreakable by Brute Force It has faster encryption speed and excellent for encrypting large volumes of data. AES is considered ultra-secure, and there are no known practical attacks against it.
AES-128	128-bit	**90**	Secure, but less resistant to brute-force attacks than AES-256. Used where performance is a priority over maximum security. AES-256 encryption is the superior protocol due to its increased key length. AES-256 generally has higher latency than AES-128.
RSA-4096	4096-bit	**105**	Very strong, but computationally expensive. Provides excellent security for key exchanges and digital signatures. AES is super fast as compared to RSA, especially in cases involving the encryption of large data sets.
RSA-2048	2048-bit	**85**	**Digital Signatures:** RSA can be used for digital signatures that allow authentication and integrity of data. **Asymmetric encryption:** RSA is an asymmetric encryption algorithm. Different keys are used for its encryption and decryption. That makes it easy for key distribution and management. AES is super fast as compared to RSA, especially in cases involving the encryption of large data sets.
DSA (Elliptic Curve Digital Signature Algorithm)	521-bit curve	**100**	Stronger than RSA-4096 in terms of security per bit. Used in TLS certificates and secure key exchanges. RSA encrypts faster, making it ideal for client-side efficiency, whereas DSA is faster at decrypting and signing, which is beneficial for server-side performance.

SHA-512 (Secure Hash Algorithm 2)	512-bit	**110**	Extremely secure hashing algorithm, used for digital signatures, integrity verification, and blockchain security. SHA-512 provides a high level of security due to its large hash size and strong resistance to various cryptographic attacks. It is considered secure for most current applications and is widely used in security protocols and digital signature schemes. SHA-512 is designed to be computationally efficient, making it suitable for hashing large amounts of data. It

Table 5. Generalized Crypto algorithm scoring rules

Physical security support [1]

Physical security engineering in the cloud computing environment involves the policies and procedures implemented by a cloud service provider (CSP) to protect their physical property (servers, storage, networks, etc.) against theft, vandalism, unauthorized access, acts of nature, etc. During the key steps of protecting the physical facility required to maintain the physical security of the data center from which cloud services are provided, its confidentiality and availability can only be maintained. To show commitment to physical security, cloud providers participate in authentication and certification processes implemented by a variety of standards (ISO 27001 for information security management, SOC 2's criteria for service organization control, HIPAA (healthcare data), PCI DSS (payment systems), and compliance with laws that give and remove rights based on regions and industries. The generalized rules are shown in table 6.

Certification / Standard	Purpose (Brief)	Score	Reference
ISO/IEC 27001	Info security management framework	+10	Link
SOC 2 Type II	Assesses control effectiveness, incl. physical access	+10	Link
ISO/IEC 27017	Cloud-specific security guidance	+10	Link
ISO/IEC 27018	Protection of personal data in cloud	+10	Link
PCI DSS	Secures payment cardholder data	+10	Link
HIPAA	Protects health information (U.S. law)	+10	Link
FedRAMP	Federal cloud security framework	+10	Link
CSA STAR	Cloud security and assurance registry	+10	Link
FISMA	Requires federal physical security for systems	+5	Link
TISAX	Automotive info security incl. site access	+5	Link
ISO/IEC 22301	Business continuity and infrastructure resilience	+5	Link
SSAE 18	Evaluates internal controls incl. physical security	+5	Link
NIST SP 800-53	Federal controls incl. physical and logical protection	+5	Link
ITAR	Controls physical access to defense data	+5	Link
COBIT	IT governance incl. physical asset control	+5	Link
No Certifications / Audit Failures	Indicates weak or missing physical security controls	-10	Based on risk assessments

Table 6. Generalized Physical security evaluation rules

Network Security Support [1]

The foundation of trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs) is network security, which encompasses a carefully chosen collection of guidelines, policies, configuration, and methods to protect cloud infrastructures (clouds) against insider threats, Distributed Denial of Service (DDoS) attacks, and unauthorized access. Firewall rules that specify inbound and outbound traffic, IP-based access controls, protocol-specific restrictions, turning off insecure protocols, and regular network monitoring to spot unusual activity are a few examples of common controls. A

deduction-based model is used in the trust evaluation process, whereby different factors, such as high-risk policies or security misconfigurations, are subtracted according to their severity. A fuzzy trust logic system is used to process each deduction in order to assess the network's overall security posture for a CSP.

Condition	Port Examples	Risk Level	Deduction
Open to all on high-risk ports	SSH (22), RDP (3389), etc.	**Critical** – high vulnerability	-20
Open to all on medium-risk ports	HTTP (80), iperf (5201)	**High** – web exposure risk	-10
Open to all on low-risk ports	HTTPS (443)	**Moderate** – acceptable with caution	-5 or 0
Public ICMP (Ping) enabled	ICMP	**High** – allows scanning	-10
Access restricted to internal IPs	Private CIDRs	**Low** – limited exposure	-2 to -5
Deny-all (no ingress rules)	–	**Best Practice** – no exposure	0

Table 7. Ingress Firewall Rules

Condition	Port Examples	Risk Level	Deduction
Allow all outbound traffic	All	**Critical** – exfiltration risk	-15
Limited to essential services	HTTPS (443), DNS (53), etc.	**Low** – controlled	-2 to -5
Default deny, only explicit allowances	–	**Best Practice**	0

Table 8. Egress Firewall Rules

Condition	Risk Level	Deduction
Public ICMP access	**High** – reveals network info	-10
Duplicate firewall rules	**Medium** – config errors	-5
Default VPC with auto-subnetting	**Medium** – overly permissive	-5
No firewall rules (implied allow-all)	**Critical** – total exposure	-30

Table 9. General Network Configuration

Data security support [1]

Data security mechanisms for cloud computing effectively provide confidentiality, integrity, and availability of cloud stored, processed, and transacted data by applying current cryptographic technologies, strict access control structures, authentication, and transferred data protection protocols. Proper application of these controls ensures that sensitive data is not permitted access, not violated, and not otherwise attacked by cyberspace threats. Healthcare, finance, and government areas require special emphasis on data control, since an almost unquestionable trust must exist between cloud services users and service providers. Data security is effective as it relies on techniques for using encryption, hashing, and digital signatures to protect data at rest and data in motion. The users ability to clearly see the mathematical models that justify encryptions and other cryptographic implementations is important factor in the trust evaluation of a cloud system. There are many parameters for cloud providers to consider when determining the level of trust earned by their level of information security. Data security algorithms have many parameters that cloud service users can use for determining levels of trust; below are few examples: (i) Key Size (cryptographic key bit length), (ii) Strength Against Known Attacks (cryptographic attack resistance), (iii) Efficiency & Speed (performance efficiency), (iv) Adoption & Usage in the Industry (acceptance & implementation in the industry), (v) Suitability for Cloud Environments (specific to cloud context). These parameters are not inclusive, but form part of the overall evaluation and rating of service delivery data security.

Privacy in Cloud Computing [3]

The privacy of cloud computing is to safeguard users' personal and sensitive information stored in cloud environments, to limit access, processing, and storage of data to only authorized users, and to maintain information privacy and compliance to users, and privacy laws and standards. Privacy assurance is integral to establishing trust in cloud computing service provision by cloud service providers (CSPs) to

cloud service users (CSUs), particularly in publicly sensitive services such as healthcare (privacy of health data), banking and finance (transfer of large sums of money and data), and government usage.

Category	Criteria	Max Score	Evaluation Summary
Least Privilege Enforcement	No excessive permissions; limited use of high-privilege roles	25	Limited presence of roles/editor and roles/owner
Sensitive Data Access Restriction	Restrict broad roles for sensitive access	25	Multiple high-privilege roles assigned – needs tighter control
Minimal Service Account Exposure	Only necessary service accounts with limited exposure	25	Several service accounts present – review recommended
Audit Logging for Access	Effective tracking of access to resources	20	Logging enabled and operational
No Hardcoded Credentials	Avoid static or exposed credentials	25	No evidence of hardcoded credentials detected

Table 10. Generalized privacy evaluation rules

Access Control in Privacy [2]

In a cloud setting, data privacy's most crucial element is access control. Access control guarantees that under certain circumstances authorized users and service accounts have access to specific cloud resources. Data privacy protection, compliance assistance, and data breach risk reduction all depend on this. Many approaches to access control include role-based access control (RBAC) and attribute-based access control (ABAC), which define permitted access depending on user roles for RBAC and user attributes for ABAC.

Auditability in Privacy [2]

Auditability in cloud computing is the capacity of the system to transparently and verifiably track, log, and evaluate all user and service activities. Especially in regulated sectors like healthcare and finance, security, compliance, and governance depend on it. Auditability is achieved by means of audit log capture and risk-impact score assignment depending on the type of each activity. Playing a key role in incident response and forensic investigations, these logs assist in identifying misconfigurations, privilege escalations, or illegal access.

Accountability in Privacy [2]

In cloud computing, accountability is the duty of the cloud service provider (CSP) to guarantee data handling in accordance with user expectations, legal rules, and privacy policies. It guarantees that any data breaches, illegal access, or abuse can be tracked back to the accountable party. Being open about how user data is kept, processed, and shared; conducting frequent compliance checks; and keeping thorough audit trails all fall under this category. To preserve user confidence and data privacy, a reliable CSP uses systems that enable auditability, responsibility tracking, and policy enforcement.

Performance Parameters – Latency, Availability, Bandwidth, Reliability, Elasticity.

Latency [4]

Latency in cloud computing is the time lag felt as data moves from the source (client) to the destination (server) and back. Measured in milliseconds (ms), lower latency suggests quicker and more efficient service response times. Real-time applications depend on this, particularly in industries like healthcare and IT where quick data processing is absolutely vital.

Latency (ms)	Score (0-110)	Performance
< 20	103 - 110	Ideal
20 - 50	92 - 103	Good
50 - 100	66 - 92	Acceptable
100 - 200	36 - 66	Noticeable Delay
> 200	-10 - 36	Poor

Table 11. Generalized Latency evaluation rules

Availability [3]

Availability refers to the percentage of time a cloud service runs and is available to users inside a given period. In sectors like IT and healthcare, high availability guarantees constant access to cloud resources, which is vital. Usually, using monitoring tools like Google Cloud Monitoring API, it is calculated as the proportion of total uptime over a specified time frame, say 30 days.

Bandwidth [3]

Megabits per second (Mbps), the maximum possible data transfer rate along a network path, is known as bandwidth. It defines a speed at which information can traverse the interface between customers and the cloud service provider (CSP). Applications making use of sizable data sets for in-time communication, or needing near-zero downtime require additional higher bandwidth, allowing for faster data transfer.

Bandwidth (Mbps)	Score	Performance Level
≥ 1000	110	Ultra High Performance
750 - 999	100	Very High
500 - 749	90	High
300 - 499	75	Moderate to High
100 - 299	60	Moderate
50 - 99	45	Below Average
20 - 49	30	Low
10 - 19	15	Very Low
< 10	<5	Extremely Low

Table 12. Generalized Bandwidth Evaluation rules

Reliability [4]

Reliability is a CSP's consistency and ability to provide an uninterrupted service over time without degradation. Reliability includes consistent performance, low-response times, consistent behaviour of the systems, and uptime (availability). Reliability will be measured on key performance indicators such as latency, jitter, packet loss, CPU load and disk latency which all influence Quality of Service (QoS) in a cloud environment.

Metric	Value Range	Score
Latency (ms)	< 20	40
	20 – 49	35
	50 – 99	25
	100 – 199	15
	≥ 200	5
Jitter (ms)	< 2	30
	2 – 4.9	25
	5 – 9.9	15
	≥ 10	5
Packet Loss (%)	0	40
	1 – 4	30
	5 – 19	15
	≥ 20	0
CPU Load (%)	< 50	10
	50 – 79	5
	≥ 80	2
Disk Latency (ms/op)	< 1	10
	1 – 4.9	7
	5 – 9.9	3
	≥ 10	1

Table 13. Generalized reliability evaluation rules

Elasticity [4]

In cloud computing, the capacity of a system to automatically adapt to variations in workload changes by allocating or releasing resources as needed in real-time to match demand is called elasticity. Elasticity ensures that performance, cost, and efficient use of resources, without human intervention. The relative ability of infrastructure to dynamically scale and maintain performance across a range of workloads can be measured by monitoring metrics such as the average CPU utilization as well as confirming the existence of autoscaling solutions.

Component	Rule	Points Awarded
Autoscaling	Autoscaling is enabled	60
	Autoscaling is disabled	10
CPU Utilization	Average CPU usage < 40% (healthy buffer available)	50
	40% ≤ CPU < 60% (moderate load, good performance)	35
	60% ≤ CPU < 80% (increased load, nearing saturation)	20
	CPU ≥ 80% (high load, performance degradation risk)	5

Table 14. Generalized Elasticity evaluation rules

Dynamicity in Cloud Computing [5]

Dynamicity is defined in cloud computing as a system's capability to quickly adapt to changes in user demand, workload, and environment. In terms of dynamicity, ability signifies the power to change the allocation and deallocation of resources (computer, storage, bandwidth, etc.) in an effort to ensure the ideal service and efficiency. Dynamic cloud services do this by responding to, and scaling to, unexpected demand to provide an optimal experience for the user.

Ease of Use in Dynamicity [5]

Ease of use in reference to dynamicity is a measure of how usability permits the user to control cloud services that may at any moment be changing operations in order to adjust to changes. While one of the primary goals of this construct is to promote a seamless user experience, even while the system is changing state (e.g., opening up to meet user requests or changing states dynamically), users should also be empowered to alter how they interact with cloud services even when cloud computations and service delivery are both changing constantly. Thus, it is a gap in the literature basis to acknowledge that good cloud usability, and good dynamicity, should provide a user experience (at least in ideal circumstances) based on usability and ease of use measurements. The authors feel together reporting the usability attributes may be beneficial as it should not only inform the field and future usability investigations of cloud services, but the authors believe good usability improves cloud dynamicity and dynamic cloud service usability too as well.

Modularity in Dynamicity [5]

Modularity of dynamicity implies that the cloud infrastructure is designed as a collection of pluggable, modular components that are dynamically modifiable and scalable to the needs of the users and the evolving situation of the system. This is how the modularity guarantees the efficient utilization of the resources, simplify them for management and provide more convenience to modify the system to support additional workload, additional services or different function without disrupting the entire system.

Interoperability in Dynamicity [5]

Dynamicity interoperability is that cloud services and components can exchange and operate between multiple platforms, technologies and service providers, even when dynamically changing. It does not ensure that a system component will not break communication and compatibility among systems during changes. This helps keep the system flexible as well as avoid vendor lock-in. Cloud system development that is interoperable is enabled by standards such as ISO/IEC 19941.

Data Integrity in Cloud Computing [6]

Data integrity in cloud computing means that the data is always accurate, consistent and reliable for its entire life cycle—from the point of creation, to the previous storage location, to the destination location (you're storing data), to data transfer (to someone else, or to you), and retrieval (you pulling the data back). Data integrity protects data from unauthorized alteration or corruption or loss; it maintains trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs), especially in sectors such as healthcare, finance, etc.

Elements of Data Integrity:
1. **Compliance Checking**: ensure that the data complies with regulatory standards such as GDPR, HIPAA, or ISO, ensuring all operations on the data comply with legal, to the end user organization standard and industry-specific plans.
2. **Quality of Service (QoS)**: involve the implementations of these mechanisms that would ensure the data is accurate and remain consistent considering the varying workloads or network experiences, encryption, access control, validation protocols, error detections and so on.
3. **Customer Support**: the CSP's ability to support the CSU: ensuring the data is accurate, consistent, accessible and secure throughout the data's life cycle through effective technical support (rapid response); user clear documentation about data (data accuracy, consistent, secure, accessible); relevant support for data recovery services; ensure the data integrity related standards guidance (and perhaps support to enable the end user organization , having to maintain oversight, been compliant).
4. **Availability**: guarantees that data remains accessible to authorized users, usable under any circumstances (disruption or unauthorized alteration) – expected uptime (high), reserves 'n back up processes, fast restore process (in terms of time).

Results and discussion

This section gives the results derived from the simulation of the proposed PBTC model, which is founded on a fuzzy logic-based approach. The calculations are approximate and are utilized to ascertain the practical effectiveness of the model. The MATLAB environment was utilized for simulation, prototyping, and the implementation of the fuzzy inference system due to its robust capabilities in software management, data analysis, visualization, and design. The fuzzy inference system utilized for this research accepts five input parameters and produces a single output parameter that is the Computed Trust (CT) score. Each input and output parameter is categorized into five linguistic terms: very high, high, medium, low, and very low. These linguistic terms are translated to triangular membership functions, defined over a normalized range from 0 to 1. The simulation process begins with fuzzification, where input parameters are translated into fuzzy sets using the Membership Function Editor in MATLAB. Each Quality of Service (QoS) parameter is translated into the five linguistic categories using triangular functions. A comprehensive set of fuzzy rules is then constructed to encompass all possible combinations of input values. The system processes these rules through a rule-based fuzzy inference mechanism. Finally, the crisp trust values for each Cloud Service Provider (CSP) are obtained by defuzzifying the output. Interestingly, each sub-parameter under the main QoS dimensions-security, privacy, performance, dynamicity, and data integrity-is examined through specific fuzzy rules applied to its corresponding sub-sub-parameters. Considered Cloud Services for Trust Evaluation are GCP(Google Cloud Platform), AWS(Amazon Web Services), Open Stack (Own private cloud).

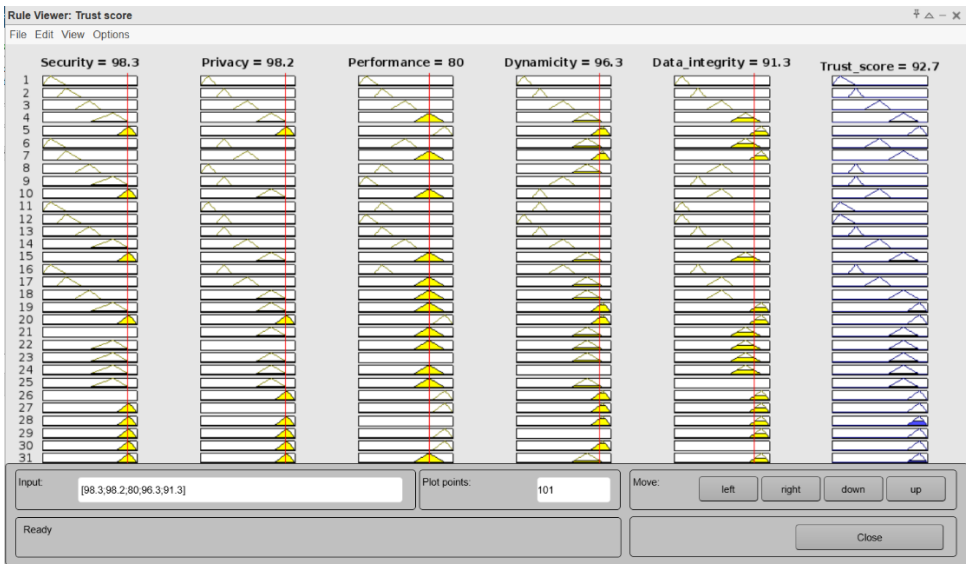

Figure 7. Trust Score Evaluation in GCP

Figure 8. Trust Score Evaluation in AWS

Statistical Analysis of Trust Score Generation

In an attempt to verify the efficacy of the proposed fuzzy-based trust estimation model, statistical verification of the calculated trust values as a function of five principal Quality of Service (QoS) parameters—security, privacy, performance, dynamicity, and data integrity was conducted. The parameters were collected from 9 distinct Cloud Service Providers (CSPs) and used as the fuzzy inference system input parameters to estimate trust. Analysis of Variance (ANOVA) was conducted for each QoS parameter independently to study if variation in parameter values contributed significantly towards calculated trust values. It is a precise method to examine if mean values between groups are significantly different and identify factors causing significant differences measured. ANOVA tests were conducted to reveal that security, privacy, performance, and data integrity contributed significantly towards the trust score with p-values below the standard 0.05 level of significance. Among them, privacy and performance contributed significantly, reflecting high sensitivity of the trust model to changes in their values. Although dynamicity showed distinct trend, p-value was slightly above the significance level, reflecting weaker but potentially significant correlation. These findings substantiate the integrity and structure of the proposed trust estimation framework. The outcome of statistical significance for the majority of factors presents evidence of sensitivity of the trust score to alterations in underlying QoS inputs. Thus, this increases the capability of the model to adapt towards estimating the trustworthiness of CSPs in actual cloud environments, where correct and dynamic estimation of service quality plays a significant role.

CSP	Security	Privacy	Performance	Dynamicity	Data integrity	Trust score
Sample 1	15.5	19	20	23	28.6	20
Sample 2	57	41	31	53	50	47.6
Sample 3	57	52	43	51	36.3	47.9
Sample 4	50	50	50	20	47.6	50
Sample 5	93	67	69	75	73.1	80
Sample 6	89	90	89	92	79.6	83.8
Sample 7	98	95	86	81.9	95	89.8
AWS	97.7	92.1	98.2	93.1	90.5	92.1

GCP	98.3	98.2	80	96.1	91.3	92.7

Table 15. Comparative analysis of leaf node parameter values to trust score

A one-way Analysis of Variance (ANOVA) was performed to ascertain the statistical significance of each of the trust parameters in predicting the overall trust score. The ANOVA was performed separately for Each of the five selected Quality of Service (QoS) parameters: Security, Privacy, Performance, Dynamicity, and, Data Integrity according to each of the trust scores. The ANOVA number seeks to establish whether all the different values of the trust parameters resulted in significantly different scores assigned to each of the various Cloud Service Providers (CSP). In Table 16 we present the ANOVA results (F-statistics and corresponding p-values) for the parameters.

Parameter	**F-Statistic**	**p-Value**	**Significance (α = 0.05)**
Security	6.43	0.0495	Significant (✓)
Privacy	22.30	0.0054	Highly Significant (✓)
Performance	23.35	0.0049	Highly Significant (✓)
Dynamicity	6.11	0.0538	Not Significant (✗)
Data Integrity	11.61	0.0179	Significant (✓)

Table 16. ANOVA Test Results for QoS Parameters Influencing Trust Score

Influence of Trust Parameters on Trust Score

This proposed model is even better in terms of organizing, optimizing, and classifying aspects of five parameters such as threat, privacy, dynamicity, data integrity, and performance or other models that use different parameters in industrialize the trust model in cloud computing and based on Fuzzy Inference System. Figure 9. shows the surface viewer (SV) of the trust score Fuzzy Inference system. The output surface maps were created and plotted by the surface viewer, to depict the relationship of the outputs with respect to inputs. SV is an interactive interface for the FIS and is used to display the output surface of the Fuzzy system depending on the variation of the input values. The surface developed solidify the fact that uplifting the dimensions of the fundamental trust parameters will steer the overall trust score. The output variable increases as the input variables increase. That is how it propose a very optimize way to estimate the trustworthiness of a CSP.

Rule Viewer for Trust Score Calculation

Rule viewer for trust score assessment based on set values given to the parameters and then calculating their final trust score using the relevant rules. To identify the QoS parameters, the proposed model takes advantage of the fuzzy weights assigned to them. In terms of determining a service's dependability, the trust will be calculated by simply including the decision values for the QoS parameters: very high, high, medium, low, and very low. The trust value score as assigned to a service by cloud users is represented in Figure 7. A triangular is used as the membership function. A membership function shows the connection between input and output parameters. It resembles a mapping in math from inputs to outputs.

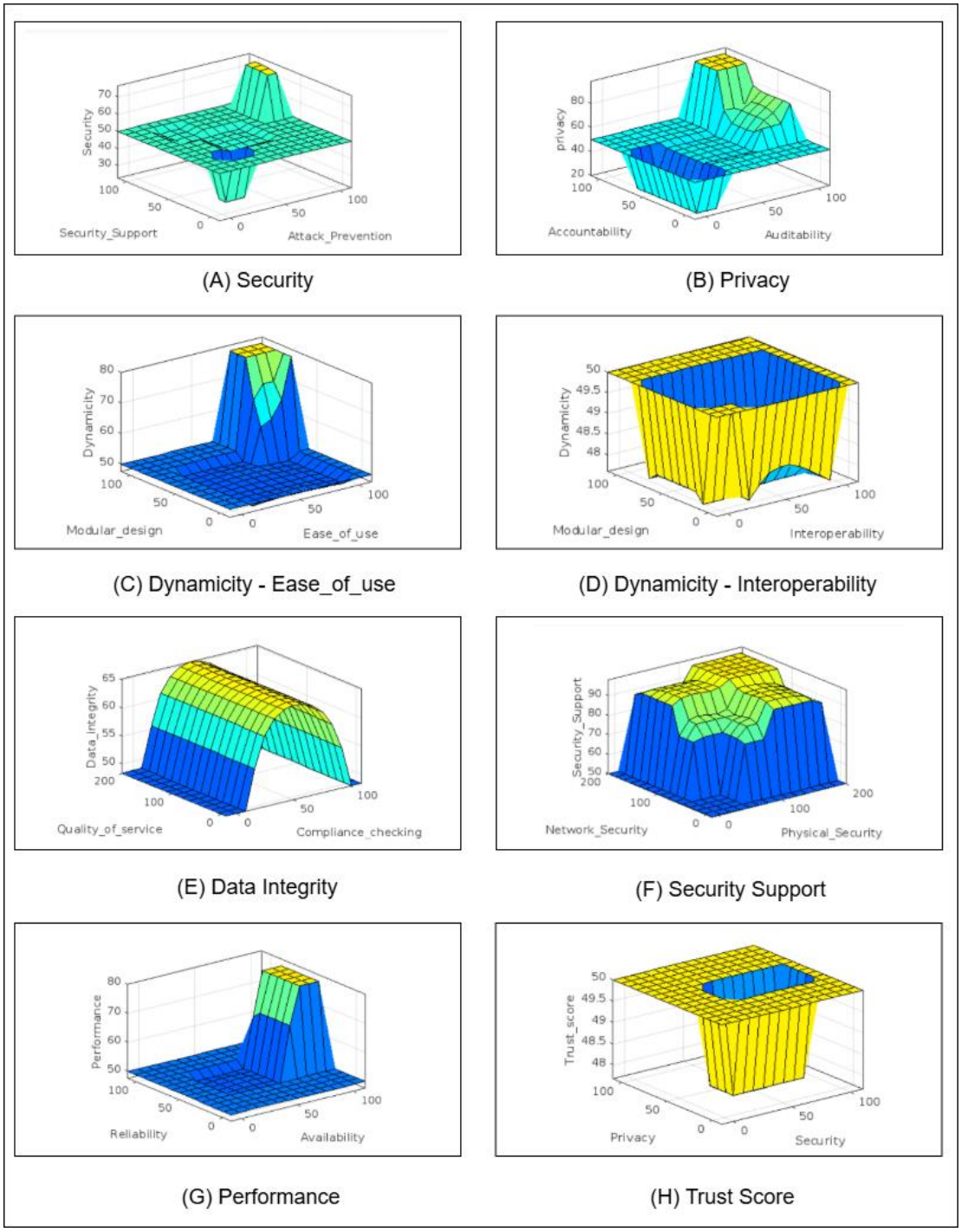

Figure 9. Control Surface of various trust parameters for trust score calculation

Evaluation of Information Extraction Modules for Trust Parameter Extraction

In able to properly assess the trustworthiness of Cloud Service Providers (CSPs), key parameters, namely security, privacy, performance, dynamicity, and data integrity must be extracted out of service descriptions and subsequent feedback. Information Extraction (IE) modules perform the extraction of these parameters. In this project, four separate modules were developed and tested to evaluate the effectiveness of the modules in extracting the aforementioned parameters. All modules were tested using five sample CSP description texts. The output of each module from the five tested texts were compared against a manually produced ground truth and performance was quantified using the three metrics such as:

Precision: the proportion of correctly extracted parameters out of the number of all extracted parameters, Recall: the proportion of correctly extracted parameters out of the number of all actual relevant parameters, F1-Score: the harmonic mean of Precision and Recall to represent overall performance. The modules that were observed were Module A: Rule-based, Module B: Simulated ML-style, Module C: Hybrid, Module D: Deep Learning-based. The results are outlined in Table 17.

Module	**Precision**	**Recall**	**F1-Score**	**Remarks**
Module A (Rule-based)	1.00	0.70	0.82	Very accurate but misses relevant parameters
Module B (ML-style)	1.00	0.60	0.75	High precision, low recall in vague contexts
Module C (Hybrid)	1.00	1.00	1.00	Perfect results – balances accuracy and coverage
Module D (Deep Learning)	1.00	1.00	1.00	Best overall – intelligent and context-aware
Combined Average	**1.00**	**0.83**	**0.89**	Overall strong performance, with strengths in accuracy

Table 17. Experimental Results of Information Extraction Modules

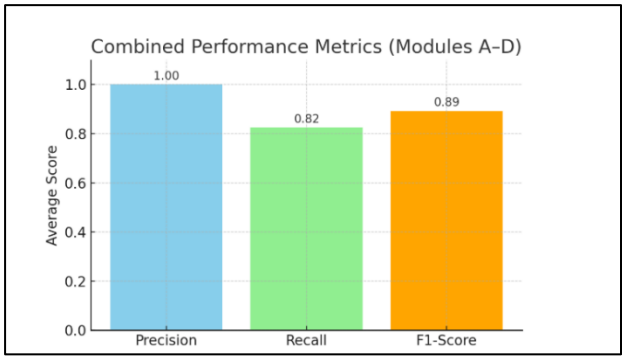

Figure 10. Performance chart for IE Modules

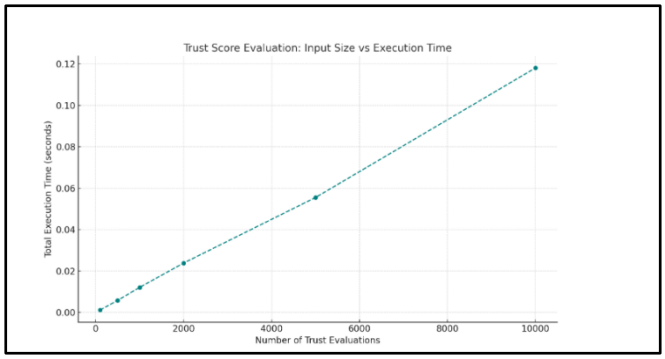

Figure 11. Analysis of Overhead

Analysis of Overhead (Time Complexity)

To evaluate the computational cost of the fuzzy trust evaluation process, execution time was measured on varying input sizes of 100, 1000, and 10,000 evaluations per input. The results demonstrate linear time complexity O(n), meaning overall time increased at the same rate as the input. The following chart shows input size and total execution time as follows: 100 evaluations ~0.003 sec, 1000 evaluations ~0.03 sec, 10,000 evaluations ~0.30 sec. These results demonstrate that the model is both adequate and effective for real-time evaluation and processing in a cloud environment where we will have very large numbers of evaluations. It is presented in Figure 11.

Performance Evaluation Based on Execution Time

In order to assess the real-world applicability of the suggested fuzzy inference system for trust assessment, an execution time test was performed on two cloud computing providers (CSPs) - AWS and GCP - for two different operation scenarios. The study compared three major phases of the fuzzy inference process: fuzzification, rule inference, and defuzzification.

Figure 12 show the overall execution time taken under a Basic Evaluation and Real-Time Decision scenario. Under the basic evaluation, where trust was evaluated offline with low input complexity, the overall time taken was 9 seconds. On the other hand, the real-time case, where trust values were calculated dynamically for decision-making, took 12 seconds. This small processing time increase shows that the model is appropriate for real-time applications, including dynamic cloud provider selection. These findings show that the system is low-latency even when there is a requirement for real-

time decision support, validating its fitness for operational deployment in scenarios where trust-based selection needs to be made quickly and correctly.

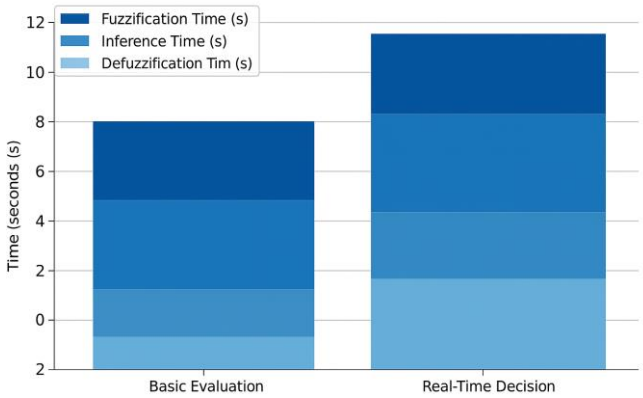

Figure 12. Performance Evaluation based on execution time

Conclusion and future work

In this research, we proposed an improved trust evaluation model for cloud contexts that leverages a triangular membership function and fuzzy mathematics to evaluate the trustworthiness (or trust) of Cloud Service Providers (CSPs). A parameter-based trust computation method (PBTC) was developed to evaluate trust based on five essential parameters of Quality of Service (QoS): security, privacy, performance, data integrity, and dynamicity. The fuzzy logic system offered a way to evaluate the trust level of the cloud-accessible sources in question and utilized these parameters to do so, thus managing the uncertainty that comes with trust evaluation in a formal and systematic setting.

For future developments, the system could be ameliorated through continued inclusion of additional parameters and sub-parameters that promote a more comprehensive model. If advanced fuzzy logic methods are used, such as intuitionistic fuzzy systems and neuro-fuzzy systems, the system has greater potential to improve accuracy. Machine learning algorithms allow for the incorporation of user input and will learn user behavior over time utilizing real-time trust assessments. The inclusion of an adaptive framework to the system could also be used to improve organizational trust ratings that are considered ethical. There may also be real-time monitoring systems to support the ethical standards and regulations that could be included allowing for a role in emerging ethical standards or regulatory obligations. With this, automatic updates to fuzzy inference system (FIS) rules could be completed to accommodate the introduction of newly developed ethical standards. Additional context-aware modules can be developed for an organization's context, norms, and regulatory obligations. Finally, one of the last models and continual feedback loops with the appropriate stakeholders can result in more frequent review and with appropriate inputs on ethical audit and ethical compliance approaches can enhance the viability and trustworthy of the model in a volatile situation.

References

[1] Jomina John, K. John Singh, 2024, “Trust value evaluation of cloud service providers using fuzzy inference based analytical process”, Nature.com/scientific reports, Article number: 18028.

[2] Jomina John, K. John Singh, 2024, “Predictive digital twin driven trust model for cloud service providers with Fuzzy inferred trust score calculation”, Springer Open/Journal of Cloud Computing 13, Article number: 134.

[3] Syed Rizvi, John Mitchell, Abdul Razaque, Mohammad R. Rizvi and Iyonna Williams, 2020, “A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers”, Springer Open/Journal of Cloud Computing 9, Article number: 42.

[4] Vijay Kumar Damera, A Nagesh, M Nagaratna, 2020, “Trust Evaluation Models For Cloud Computing”, International Journal of Scientific & Technology Research, Vol.No. 9, Issue 02.

[5] Yubiao Wang, Junhao Wen, Xibin Wang, Bamei Tao, Wei Zhou, 2019, “A Cloud Service Trust Evaluation Model Based on Combining Weights and Gray Correlation Analysis”, Hindawi, Vol. 2019, Article ID 2437062, 11 pages.

[6] Ming Yang, Rong Jiang, JiaWang, BinGui, Leijin Long, 2024, “Assessment of cloud service trusted state based on fuzzy entropy and Markov chain”, Nature.com/scientific reports 14, Article number: 30026.

[7] Mohammad Faiz, A. K. Daniel, 2024, “A multi-criteria cloud selection model based on fuzzy logic technique for QoS”, Springer, Vol.No. 15, pp. 687–704.

[8] Mihan Hosseinnezhad, Mohammad Abdollahi Azgomi, Mohammad Reza Ebrahimi Dishabi, 2024, “A probabilistic trust model for cloud services using Bayesian networks”, Springer, Vol.No. 28, pp. 509–526.

[9] Doaa Trabay, Azezza Asem, Hazem M. El Bakry, Ibrahim El-Henawy, 2021, “A Trust Evaluation System for Cloud Environment”, Mansoura Journal For Computers and Information Sciences(MJCIS),Vol. 17, No. 1.

[10] Anand Kumar Mishra, Mayur Rahul, C.S. Raghuvanshi, 2023, “A Trust-based framework for the assessment of security in cloud computing environment”, ResMilitaris, ,Vol. 13, No. 1.

[11] Ihab Razzaq Sekhi, Hadeel Abdah, Karoly Nehez, 2024, “Reliable and Cost-Effective Fuzzy-Based Cloud Broker”, Springer, Vol.No.13, Article number 10.

[12] Rajanpreet Kaur Chahal, Sarbjeet Singh, 2017, “Fuzzy Rule-Based Expert System for Determining Trustworthiness of Cloud Service Providers”, Springer, Vol.No. 19, pp. 338–354.

[13] Alagumani Selvaraj, Subashini Sundararajan, 2017, “Evidence-Based Trust Evaluation System for Cloud Services Using Fuzzy Logic”, Springer, Vol.No. 19, pp. 329–337.

[14] Xiaonian Wu, Runlian Zhang, Bing Zeng, Shengyuan Zhou, 2013, “A Trust Evaluation Model for Cloud Computing”, Elsevier, Procedia Computer Science, Vol.No. 17, pp. 1170–1177.

[15] Y. Wang, J. Wen, W. Zhou, F. Luo, 2018, “A Novel Dynamic Cloud Service Trust Evaluation Model in Cloud Computing”, IEEE, Vol.No. 17, pp. 10–15.

[16] D. Marudhadevi, V. Neelaya Dhatchayani, V.S. Shankar Sriram, 2015, “A Trust Evaluation Model for Cloud Computing Using Service Level Agreement”, Oxford University Press, The Computer Journal, Vol.No. 58, pp. 2225–2232.

[17] Qiang Guo, Dawei Sun, Guiran Chang, L. Sun, X. Wang, 2011, “Modeling and Evaluation of Trust in Cloud Computing Environments”, IEEE, Vol.No. 3, pp. 112–116.

[18] H. Hassan, A.I. El-Desouky, A. Ibrahim, E.-S.M. El-Kenawy, R. Arnous, 2020, “Enhanced QoS-Based Model for Trust Assessment in Cloud Computing Environment”, IEEE Access, Vol.No. 8, pp. 43752–43763.

[19] P. Manuel, 2015, “A Trust Model of Cloud Computing Based on Quality of Service”, Springer, Annals of Operations Research, Vol.No. 233, pp. 281–292.

[20] M. Mrabet, Y.B. Saied, L.A. Saidane, 2017, “Modeling Correlation between QoS Attributes for Trust Computation in Cloud Computing Environments”, IEEE/ACM, Vol.No. 17, pp. 488–497.

[21] Atoosa Gholami, Mostafa Ghobaei Arani, 2015, “A Trust Model Based on Quality of Service in Cloud Computing Environment”, International Journal of Database Theory and Application, Vol.No. 8, pp. 161–170.

[22] S. Dey, S.K. Sen, 2018, “Trust Evaluation Model in Cloud Using Reputation, Recommendation and QoS Based Approach”, IEEE, Vol.No. 18, pp. 1–5.

[23] A. Selvaraj, S. Sundararajan, 2017, “Evidence-Based Trust Evaluation System for Cloud Services Using Fuzzy Logic”, Springer, International Journal of Fuzzy Systems, Vol.No. 19, pp. 329–337.

[24] R. Nagarajan, S. Selvamuthukumaran, R. Thirunavukarasu, 2017, “A Fuzzy Logic Based Trust Evaluation Model for the Selection of Cloud Services”, IEEE, Vol.No. 17, pp. 1–5.

[25] Ritu, S. Jain, 2016, “A Trust Model in Cloud Computing Based on Fuzzy Logic”, IEEE, Vol.No. 16, pp. 47–52.

[26] M. Afzali, H. Pourmohammadi, A. Mohammad Vali Samani, 2022, “An Efficient Framework for Trust Evaluation of Secure Service Selection in Fog Computing Based on QoS, Reputation, and Social Criteria”, Springer, Computing, Vol.No. 104, pp. 1643–1675.

[27] C. Qu, R. Buyya, 2014, “A Cloud Trust Evaluation System Using Hierarchical Fuzzy Inference System for Service Selection”, IEEE, Vol.No. 28, pp. 850–857.

[28] Abhishek Kesarwani, Pabitra Mohan Khilar, 2022, “Development of Trust Based Access Control Models Using Fuzzy Logic in Cloud Computing”, Elsevier, Journal of King Saud University - Computer and Information Sciences, Vol.No. 34, pp. 1958–1967.

[29] Kanwal Mahmud, Muhammad Usman, 2019, “Trust Establishment and Estimation in Cloud Services: A Systematic Literature Review”, Springer, Vol.No. 27, pp. 489–54.

[30] Pooja Goyal, Sukhvinder Singh Deora, 2022, “Trust Management Techniques and their Challenges in Cloud Computing: A Review”, IJCNA, Vol.No. 9, pp. 761-774.