A FUZZY LOGIC APPROACH FOR TRUST EVALUATION IN CLOUD SERVICE PROVIDERS

PROJECT WORK

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF **BACHELOR OF TECHNOLOGY** IN **INFORMATION TECHNOLOGY** OF THE ANNA UNIVERSITY

2025

Submitted by

LOGESHWARAN VV

71772118125

SEDHU RAM P

71772118138

SRIDHAR E

71772118145

TAMIZHARASU M

71772118L10

Under the Guidance of

Dr. R. DEVI, M.Tech., Ph.D.,

DEPARTMENT OF INFORMATION TECHNOLOGY

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institution affiliated to Anna University)

COIMBATORE - 641 013

DEPARTMENT OF INFORMATION TECHNOLOGY
GOVERNMENT COLLEGE OF TECHNOLOGY
(An Autonomous Institution affiliated to Anna University)
COIMBATORE - 641 013

PROJECT WORK

APRIL 2025

This is to certify that this project work entitled

A FUZZY LOGIC APPROACH FOR TRUST EVALUATION IN CLOUD SERVICE PROVIDERS

is the bonafide record of project work done by

LOGESHWARAN VV
71772118125
SEDHU RAM P
71772118138
SRIDHAR E
71772118145
TAMIZHARASU M
71772118L10

of **B.E. / B.Tech. INFORMATION TECHNOLOGY** during the year 2024 – 2025

DR. R. DEVI M.TECH., Ph.D.,	**DR. S. RATHI M.E., Ph.D.,**
Project Guide	Head of the Department

Submitted for the Project Viva-Voce examination held on _____

Internal Examiner	External Examiner

ACKNOWLEDGEMENT

Great achievements are not possible without standing on the shoulders of giants. Without the active involvement of the following experts this project would not have been a reality.

We express our sincere gratitude to **Dr. K. Manonmani, M.E., Ph.D.,** Principal, Government College of Technology, Coimbatore for providing us all facilities that we needed for the completion of this project.

We whole-heartedly express our thankfulness and gratitude to **Dr. S. Rathi, M.E., Ph.D.,** Professor and Head of the Department of Information Technology, Government College of Technology, for helping us to successfully carry out this project.

Our thankfulness and gratitude to our respectable project guide **Dr. R. Devi, M.Tech., Ph.D.,** Assistant Professor who has been an immense help through the various phases of the project. With her potent ideas and excellent guidance, we were able to comprehend the essential aspects involved.

We would like to thank our faculty advisor **Dr. M. Blessy Queen Mary, M.E., Ph.D.,** Assistant Professor for her continuous support and encouragement throughout this project.

We extend our sincere thanks to the staff members of Information Technology department, **Dr. C. Aswini, M.E., Ph.D.,** Assistant Professor, **Dr. T. Suguna, M.Tech., Ph.D.,** Assistant Professor, **Dr. S. Gladson Oliver, M.Tech., Ph.D.,** Assistant Professor, **Prof. M. Jeyanthi, M.Tech.,** Assistant Professor, **Dr. R. Malavika, M.Tech., Ph.D.,** Assistant Professor, for rendering their help for the completion of this project. We also thank all our friends for their cooperation and suggestions towards the successful completion of this project.

SYNOPSIS

Trust evaluation is necessary in cloud computing because users rely on third-party service providers to store, process, and manage their data. Without physically owning the infrastructure, users must be confident that the provider will deliver services securely, reliably, and as promised. Evaluating trust helps identify which providers are dependable, protect user data from security breaches, ensure compliance with service level agreements (SLAs), and prevent issues like data loss or misuse. In short, trust evaluation builds user confidence and supports safer, smarter decisions when choosing cloud services.

The paper proposes a novel trust evaluation framework for Cloud Service Providers (CSPs) using a fuzzy logic-based approach aimed at enhancing cloud service selection. Cloud computing, though increasingly adopted across industries due to its flexibility and cost-effectiveness, faces challenges related to trust between Cloud Service Users (CSUs) and providers. Trust, a critical parameter for cloud adoption, is influenced by multiple Quality of Service (QoS) attributes such as security, privacy, performance, data integrity, and dynamicity. The research introduces a Parameter-Based Trust Calculation (PBTC) model, integrating fuzzy logic to effectively quantify and evaluate these trust dimensions. The model transforms crisp input values into fuzzy sets using triangular membership functions and processes them via a fuzzy inference system (FIS), which includes fuzzification, rule-based inference, and defuzzification steps. Extensive literature review underscores limitations in existing models like complexity, lack of real-time capability, and limited parameter coverage. The proposed system addresses these by offering a flexible, scalable, and interpretable structure adaptable to various CSP environments. MATLAB simulations validate the framework, showing statistically significant differences in trust scores across multiple CSPs. The evaluation further identifies that parameters like security and performance heavily influence trust outcomes. The model's interpretability, adaptability, and low computational overhead make it suitable for real-time trust assessments in diverse cloud infrastructures. The study concludes with suggestions for integrating advanced fuzzy systems and machine learning for future enhancements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CSU	CLOUD SERVICE USERS
CSP	CLOUD SERVICE PROVIDERS
QOS	QUALITY OF SERVICE
PBTC	PARAMETER-BASED TRUST CALCULATION
SLA	SERVICE LEVEL AGREEMENTS
FIS	FUZZY INFERENCE SYSTEM
MCDM	MULTI-CRITERIA DECISION MAKING
ANOVA	ANALYSIS OF VARIANCE
AES	ADVANCED ENCRYPTION STANDARD
RSA	RIVEST-SHAMIR-ADLEMAN (ENCRYPTION ALGORITHM)
DSA	DIGITAL SIGNATURE ALGORITHM
SHA	SECURE HASH ALGORITHM
MD5	MESSAGE DIGEST 5
ISO	INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
IEC	INTERNATIONAL ELECTROTECHNICAL COMMISSION
SOC	SERVICE ORGANIZATION CONTROL
CSA	CLOUD SECURITY ALLIANCE
STAR	SECURITY, TRUST & ASSURANCE REGISTRY
GCP	GOOGLE CLOUD PLATFORM
AWS	AMAZON WEB SERVICES
VM	VIRTUAL MACHINE
DDOS	DISTRIBUTED DENIAL OF SERVICE

CHAPTER 1
INTRODUCTION

1.1 DESCRIPTION

This project focuses on developing a trust evaluation model for Cloud Service Providers (CSPs) using a fuzzy logic-based analytical approach. As cloud computing becomes essential in industries like healthcare, IT, and e-commerce, selecting a secure and reliable CSP is increasingly critical. However, due to the lack of transparency and varying service quality, users often face challenges in identifying trustworthy providers. To solve this, the project proposes a model that calculates a trust score by analyzing key Quality of Service (QoS) parameters such as security, privacy, performance, dynamicity, and data integrity. The model uses a Fuzzy Inference System (FIS) to convert uncertain or vague input values into accurate trust scores. It takes user feedback, runtime data, and vulnerability tests as input, processes them using fuzzy rules, and outputs a crisp trust value. This helps users make informed decisions when selecting cloud services. The project is implemented and simulated using MATLAB, and results show that the proposed method effectively ranks CSPs based on trustworthiness. The system can be further enhanced using machine learning and real-time monitoring tools for broader applications.

1.2 PROBLEM STATEMENT

Cloud computing has transformed how individuals and organizations access and manage computing resources by offering scalable, on-demand services over the internet. Despite its many advantages such as cost-efficiency, flexibility, and ease of access the widespread adoption of cloud services is hindered by a critical issue: the lack of trust between Cloud Service Users (CSUs) and Cloud Service Providers (CSPs)**.** Since users relinquish control over infrastructure and data to external entities, they must rely on service providers to deliver consistent performance, uphold service level agreements (SLAs), ensure data security and privacy, and protect against cyber threats. However, in practice, CSPs vary significantly in their capabilities, and users have limited visibility into their internal operations.

Existing trust evaluation models are either overly simplistic focusing on a narrow set of parameters or are complex and computationally expensive, making them difficult to implement in real-world systems. Moreover, many of these models are static, lack adaptability, and fail to handle the vagueness and uncertainty inherent in user feedback and dynamic cloud environments. There is also no universal framework that allows for consistent and objective trust scoring across different types of services or providers. These limitations create a significant gap in the decision-making process for users trying to select the most reliable and secure CSP for their needs. Without a comprehensive and adaptive trust evaluation model, users risk selecting untrustworthy providers, leading to potential data loss, breaches, non-compliance with regulations, and reduced service quality.

Thus, there is an urgent need for a robust, scalable, and interpretable trust evaluation model that can account for multiple Quality of Service (QoS) parameters such as security, privacy, performance, data integrity, and dynamicity. The model should be capable of handling imprecise data and delivering reliable trust scores that reflect real-time performance. Integrating fuzzy logic into this evaluation process offers a promising solution, enabling the system to interpret vague input data and simulate human-like reasoning to produce accurate trust assessments.

1.3 EXISTING SYSTEM

Trust models play a crucial role in ensuring the security and reliability of cloud services.

The trust evaluation model can be designed based on various aspects which includes agreement based, QoS based, Certificate based, feedback based, Domain based, MCDM based, Optimization based, Prediction based, Recommendation-Based, Reputation-Based.

Table 1.1 Existing Systems

Trust Evaluation Model	Key Features	Strengths	Weaknesses
QoS Based[1]	Considers Quality of Service metrics (latency, reliability, etc.), Real-time performance assessment	Provides quantitative evaluation, Reflects actual service quality	Highly dependent on accurate QoS measurements, Limited in capturing subjective aspects
Agreement-Based[2]	Relies on formal agreements between parties, Evaluates adherence to specified terms	Emphasizes contractual obligations, Well defined criteria for trust	May not adapt well to dynamic environments, Limited in handling unanticipated scenarios
Certificate-Based[2]	Uses digital certificates for authentication, Certificates issued by trusted authorities	Strong authentication and verification, Well established security infrastructure	Vulnerable to compromised certificate authorities, Overhead in certificate management
Feedback-Based[2]	Collects user feedback and ratings, Focuses on user experiences and opinions	Reflects user satisfaction and real-world experiences, Adapts to changes in service quality	Susceptible to biased or malicious feedback, Limited scalability in large-scale systems
Domain-Based[2]	Trust based on the reputation within a specific domain or context	Contextually relevant trust assessment, Specialized for domain-specific requirements	May not generalize well to diverse environments, Limited applicability outside the specified domain
MCDM-Based[2]	Multi-Criteria Decision Making approach, Considers multiple factors for trust computation	Comprehensive evaluation considering various criteria, Flexible and adaptable to diverse scenarios	Complexity in defining and weighting criteria, Computational overhead in large-scale systems
Optimization-Based[2]	Utilizes optimization algorithms to derive trust values, Focuses on maximizing overall system performance	Systematic approach to trust optimization, Addresses dynamic and changing environments	Requires efficient algorithms for scalability, Sensitivity to initial conditions in optimization

Trust Evaluation Model	Key Features	Strengths	Weaknesses
Prediction-Based[2]	Predicts future trust levels based on historical data, Utilizes machine learning or statistical models	Anticipates trustworthiness trends, Suitable for dynamic and evolving environments	Relies on accurate historical data for reliable predictions, Challenges in model interpretability
Recommendation-Based[2]	Recommends trusted entities based on past interactions, Collaborative filtering or similarity-based recommendations	Leverages collective intelligence for trust assessment, Enhances decision-making through shared experiences	Vulnerable to the "cold start" problem for new entities, Relies on the availability of sufficient historical data
Reputation-Based[2]	Evaluates trust based on entity reputation, Aggregates feedback and interactions over time	Provides long-term perspective on trustworthiness, Reflects sustained reliability and performance	Susceptible to reputation attacks or manipulation, Challenges in distinguishing malicious from sporadic poor performance

1.4 PROPOSED SYSTEM

The proposed system presents a fuzzy inference-based trust evaluation model designed to assess and rank Cloud Service Providers (CSPs) based on their trustworthiness. This system is developed in response to the growing need for a reliable, interpretable, and adaptable mechanism to help Cloud Service Users (CSUs) choose secure and dependable providers in an increasingly complex cloud ecosystem. Unlike traditional models, this system considers a wide range of trust parameters and effectively handles uncertainty and vague input data through the application of fuzzy logic.

The model begins by collecting relevant trust-related data from various sources such as user feedback, system performance logs, Service Level Agreement (SLA) compliance metrics, and results from vulnerability assessments. These inputs cover key Quality of Service (QoS) parameters that are essential for evaluating trust, including security, privacy, performance, dynamicity**, and** data integrity**.** Each

of these parameters is broken down into sub-parameters for more granular analysis. The collected values are in crisp numerical form and need to be transformed into a format suitable for fuzzy processing.

To manage the inherent imprecision and subjectivity in user feedback and service behavior, the system employs a fuzzification process. This step converts crisp input values into fuzzy linguistic variables such as “Very Low,” “Low,” “Medium,” “High,” and “Very High,” using triangular membership functions. These fuzzy sets allow the system to model human-like reasoning and better accommodate uncertainties in cloud environments.

Once inputs are fuzzified, they are fed into a Fuzzy Inference System (FIS) that applies a predefined set of fuzzy rules. These rules, based on expert knowledge or empirical data, define the relationships between input parameters and the trust output. For example, a rule might state: ‘*If Security is High and Performance is Medium, then Trust is High’*. The FIS evaluates the rules and combines them to infer the fuzzy output representing the trust level of a CSP.

To generate a usable output, the fuzzy trust score is converted back into a crisp value through defuzzification, typically using the centroid method. This final trust score is a numerical representation of how trustworthy a CSP is, considering all relevant parameters. These scores are then used to rank multiple CSPs, making it easier for users to select the most suitable service provider.

The entire model is implemented and simulated using MATLAB Simulink, which enables efficient design, testing, and visualization of the fuzzy system. The simulation results confirm the effectiveness of the model in capturing and evaluating trust across different CSPs. Statistical analysis, such as ANOVA testing, demonstrates that each parameter significantly influences the trust score, with security and performance showing the highest impact. The proposed system offers a comprehensive, scalable, and flexible trust evaluation framework that effectively handles uncertainty, supports multi-criteria analysis, and provides transparent decision-making support for users. This model not only enhances user confidence in choosing CSPs but also paves the way for future enhancements using machine learning, real-time monitoring, and adaptive rule systems.

CHAPTER 2

LITERATURE SURVEY

2.1 Trust value evaluation of cloud service providers using fuzzy inference based analytical process – 2024 [1]

2.1.1 Description

Proposes a fuzzy logic-based trust evaluation model for cloud service providers (CSPs). Evaluates trustworthiness using Quality of Service (QoS) parameters: security, privacy, dynamicity, data integrity, and performance.

2.1.2 Merits

- Comprehensive: Considers multiple trust/QoS parameters
- Flexible and adaptable to different CSPs
- Handles uncertainty and imprecision
- Transparent trust score calculation
- Demonstrated high accuracy and efficiency in simulations

2.1.3 Demerits

- Computationally intensive for large parameter sets
- Requires expert knowledge for rule and parameter setup
- Subjectivity in membership function design
- Real-time adaptability may be limited

2.2 Predictive digital twin driven trust model for cloud service providers with Fuzzy inferred trust score calculation - 2024 [2]

2.2.1 Description

A systematic trust evaluation model for Cloud Service Providers (CSPs) using a Digital Twin (virtual replica) of the CSP infrastructure. The model performs vulnerability testing on the digital twin, normalizes results, and applies a Fuzzy Inference System (FIS) to generate a trust score. The process includes continuous

refinement using feedback and historical data, enabling dynamic and comprehensive trust assessment for cloud service users.

2.2.2 Merits

- Realtime, dynamic trust calculation
- Holistic evaluation by integrating digital twins and fuzzy logic
- Visual analytics for parameter relationships (control surfaces, scatter plots)
- Continuous refinement with feedback and historical data
- Formal validation enhances reliability
- Superior performance compared to traditional models (higher accuracy, faster calculation)

2.2.3 Demerits

- Dependent on quality and timeliness of input data (e.g., penetration test results)
- Computational complexity may limit scalability in largescale deployments
- Accuracy of FIS depends on well defined membership functions and rules
- Interpretability challenges for nonexpert users
- Possible bias or outdated assumptions if historical data is not current
- Generalizability may be limited across diverse CSPs and industry sectors

2.3 A fuzzy inference system (FIS) to evaluate the security readiness of cloud service providers - 2020 [3]

2.3.1 Description

Proposes a fuzzy-logic-based trust model for evaluating the security readiness of Cloud Service Providers (CSPs). Uses subjective trust metrics and quantifies security via a Security Index, enabling Cloud Service Users (CSUs) to compare and rank CSPs even under uncertainty and incomplete information. Mamdani Fuzzy Inference System (FIS) for processing subjective and uncertain inputs. Triangular membership functions for mapping linguistic variables (e.g., harmful, risky, average, great, exceptional). Rule-based inference using 20+ if-then rules.

Centroid defuzzification to compute crisp Security Index. Mathematical trust validation using SLA-aligned equations.

2.3.2 Merits

- Effectively handles uncertainty and vagueness in user input
- Allows customization of security factor preferences
- User-friendly via linguistic variables
- Dual utility: suitable for both CSP self-evaluation and CSU assessment
- Supports ranking and comparison of multiple CSPs

2.3.3 Demerits

- Only considers top-level factors (no sub-factor granularity)
- Static membership functions may not capture all user perspectives
- Relies on subjective user input, which may introduce bias
- Case studies use hypothetical CSPs, not real-world providers
- No dynamic weighting of factors; all are treated equally

2.4 Trust Evaluation Models for Cloud Computing - 2020 [4]

2.4.1 Description

The paper systematically categorizes trust evaluation models into four classes:

- **Agreement-based**: Uses SLAs/service policies for trust establishment. Techniques are SLA negotiation, contract monitoring, third-party arbitration, static/dynamic evaluation subsystems
- **Certificate-based**: Relies on security certificates/trusted platform modules. Techniques are PKI infrastructure, TPM endorsement keys, cryptographic attestation, progressive certification
- **Feedback-based**: Leverages consumer ratings/reputation systems. Techniques are Reputation aggregation, feedback filtering, dishonest feedback detection, agent-based mechanisms, Dempster-Shafer theory

- **Domain-based**: Divides cloud into autonomous trust domains. Techniques are Trust tables, hierarchical trust management, direct/recommended trust, domain-based trust propagation

It analyses trust characteristics (subjectivity, dynamics, context-dependency) and their impact on cloud service selection.

2.4.2 Merits

- **Agreement-based**: Formalized obligations, legal enforceability
- **Certificate-based**: Strong identity verification, hardware-rooted trust
- **Feedback-based**: Real-time adaptability, crowd wisdom
- **Domain-based**: Efficient trust propagation, reduced computation overhead

2.4.3 Demerits

- **Agreement-based**: Static nature, unable to handle dynamic QoS changes
- **Certificate-based**: Centralized CA dependency, revocation challenges
- **Feedback-based**: Vulnerable to sybil/ballot-stuffing attacks
- **Domain-based**: Complex inter-domain trust reconciliation

2.5 A Cloud Service Trust Evaluation Model Based on Combining Weights and Gray Correlation Analysis - 2019 [5]

2.5.1 Description

Proposes CSTEM, a cloud service trust evaluation model that combines direct trust, recommendation trust, and reputation. It uses rough set theory and Analytic Hierarchy Process (AHP) to compute objective and subjective weights, and gray correlation analysis for recommendation trust. A dynamic trust update mechanism is included to enhance accuracy and robustness against malicious entities.

2.5.2 Merits

- Accurately combines subjective and objective weights for trust evaluation.
- Dynamic update mechanism penalizes malicious behavior and prioritizes recent, high-value transactions.

- ❖ Outperforms baseline models (CCIDTM, EigenRep, RSS) in satisfaction (up to 90%) and success rate (up to 85%).
- ❖ Robust against high proportions of malicious service entities.

2.5.3 Demerits

- ❖ Increased computational complexity due to multiple combined techniques.
- ❖ Scalability not tested beyond 300 nodes; may require optimization for larger networks.
- ❖ Limited to static cloud environments; does not address mobile or heterogeneous scenarios.
- ❖ Further dynamic factors and real-world deployment not yet explored.

2.6 Assessment of cloud service trusted state based on fuzzy entropy and Markov chain - 2024 [6]

2.6.1 Description

Proposes a hierarchical model for cloud service trustworthiness assessment, combining fuzzy entropy and Markov chain to quantitatively evaluate and predict trustworthiness changes over time. The method identifies key risk factors and provides a dynamic, objective assessment framework. Techniques used are Hierarchical attribute model (3 classes, 16 indicators), Fuzzy entropy theory for modeling uncertainty and membership functions, Markov chain for state transition prediction, Risk matrix for mapping frequency and severity to trusted states.

2.6.2 Merits

- ❖ Predictive capability for trustworthiness changes.
- ❖ Reduces subjectivity via interval based expert input.
- ❖ Integrates qualitative and quantitative analysis.
- ❖ Provides decision support for risk prevention.

2.6.3 Demerits

- High computational complexity ($O(n^2)$ state matrix calculation).
- Medium scalability when adding new indicators.
- Higher assessment cost (expert input required for all indicators).
- Interpretation of hybrid model results can be complex.

2.7 A multi-criteria cloud selection model based on fuzzy logic technique for QoS - 2024 [7]

2.7.1 Description

The paper proposes a multi-criteria cloud service provider (CSP) selection framework using a dual membership function-based fuzzy logic technique (DMFT). The model evaluates CSPs based on five key QoS parameters: cost, capacity, performance, security, and maintenance. User requirements are mapped to linguistic variables, fuzzified, and processed through a Mamdani fuzzy inference system to rank CSPs according to user priorities and service attributes.

2.7.2 Merits

- Effectively handles subjective and vague user preferences through fuzzy logic.
- Allows dynamic prioritization of QoS parameters by users.
- Demonstrates robustness and high correlation with established methods (I-TOPSIS).
- Outperforms traditional AHP/ANP in computational efficiency for complex, multi-criteria scenarios.
- Provides a flexible, user-centric CSP selection process.

2.7.3 Demerits

- High rule complexity: 1,024 fuzzy rules require significant manual setup and maintenance.
- Scalability limitations: Adding more parameters increases computational complexity exponentially.

- Model sensitivity: Rankings depend on the quality and threshold settings of input data.
- Requires expertise in fuzzy logic for effective implementation and tuning.

2.8 A probabilistic trust model for cloud services using Bayesian networks - 2024 [8]

2.8.1 Description

The paper proposes a probabilistic trust model for cloud services using Bayesian networks. It models Quality of Service (QoS) parameters as random variables to handle uncertainty and captures dependencies among them. The model is validated using the CloudArmor dataset, which contains real-world cloud service feedback. The approach aims to provide more accurate trust estimation compared to traditional deterministic or regression-based models.

2.8.2 Merits

- Effectively handles uncertainty in QoS parameters by modeling them as random variables.
- Captures and utilizes dependencies among QoS parameters for improved trust estimation.
- Demonstrates higher accuracy and lower error rates compared to regression models.
- Validated with a large, real-world dataset.
- Robust to missing or incomplete data due to Bayesian network properties.

2.8.3 Demerits

- Computationally intensive due to Bayesian network learning and inference steps.
- Discretization of continuous data may lead to information loss or oversimplification.
- Scalability to very large datasets or real-time applications is not fully explored.

- ❖ Model performance and accuracy are dependent on the quality and completeness of the input dataset.

2.9 A Trust Evaluation System for Cloud Environment - 2021 [9]

2.9.1 Description

The paper proposes a trust evaluation system for cloud environments, focusing on ranking Cloud Service Providers (CSPs) based on performance and cost using fuzzy logic. It addresses the challenge of selecting trustworthy CSPs due to the proliferation of providers and the lack of comprehensive trust evaluation methods. The approach combines simulation (Cloud-Analyst) and fuzzy logic (Mamdani FL in MATLAB) to provide an objective trust ranking for CSPs.

2.9.2 Merits

- ❖ Provides a systematic and quantitative approach for trust evaluation among CSPs.
- ❖ Combines both performance and cost, reflecting real-world customer priorities.
- ❖ Fuzzy logic allows handling of uncertainty and subjective preferences.
- ❖ The methodology is adaptable and can be extended to additional parameters (e.g., security, usability).

2.9.3 Demerits

- ❖ Only considers performance and cost; other critical factors like security and usability are not included.
- ❖ Simulation results and trust rankings are sensitive to input parameter changes.
- ❖ The approach depends on the accuracy of the simulation and the fuzzy model.
- ❖ Does not address real-time or dynamic changes in CSP performance.
- ❖ Future work is needed to include more comprehensive evaluation criteria.

2.10 A Trust-based framework for the assessment of security in cloud computing environment - 2023 [10]

2.10.1 Description

The paper proposes a trust-based framework for assessing the security of cloud computing environments. It introduces a trust score mechanism that evaluates cloud service providers (CSPs) based on multiple criteria (security, reliability, performance, compliance, customer satisfaction). The framework enables quantitative comparison and ranking of CSPs, aiding users in selecting trustworthy providers and managing risk.

2.10.2 Merits

- Provides a quantitative, objective basis for CSP comparison.
- Integrates both technical and non-technical evaluation criteria.
- Facilitates informed decision-making for cloud adoption.
- Supports continuous monitoring and iterative improvement.
- Adaptable to various organizational contexts and requirements.

2.10.3 Demerits

- Subjective assignment of weights to criteria may affect objectivity.
- Relies on availability and accuracy of data for each criterion.
- Experimental results are based on simulated data, not real-world CSP benchmarks.
- May not fully capture dynamic or evolving security threats.
- Potential scalability issues if using agent-based monitoring in large cloud environments.

2.11 Reliable and Cost-Effective Fuzzy-Based Cloud Broker - 2024 [11]

2.11.1 Description

The paper presents a fuzzy logic-based cloud broker system that matches users to cloud service instances by considering both user requirements (budget, task size) and provider offerings (CPU availability, cost). The broker classifies users and VMs into Gold, Silver, or Bronze categories using fuzzy logic and then matches them accordingly. The system supports scenarios with static and mobile users, including service migration for maintaining QoS and cost-effectiveness.

2.11.2 Merits

- Balances cost and QoS for users.
- Supports both static and mobile users.
- Service migration maintains QoS for mobile users.
- Simple input requirements (user-friendly for non-experts).
- More stable performance and less delay variation (better SLA compliance) compared to random/LL approaches.
- Practical for real-world integration.

2.11.3 Demerits

- Scalability issues: computation time increases with user count, as each user is ranked individually (no clustering implemented).
- Limited input parameters (only two per fuzzy system); may not capture all relevant factors.
- No mitigation for network delay spikes in non-migration scenarios.
- Increased computation may impact real-time performance if resources are limited.

CHAPTER 3

PROJECT DESIGN

3.1 PROJECT DESIGN

3.1.1 SEQUENCE DIAGRAM

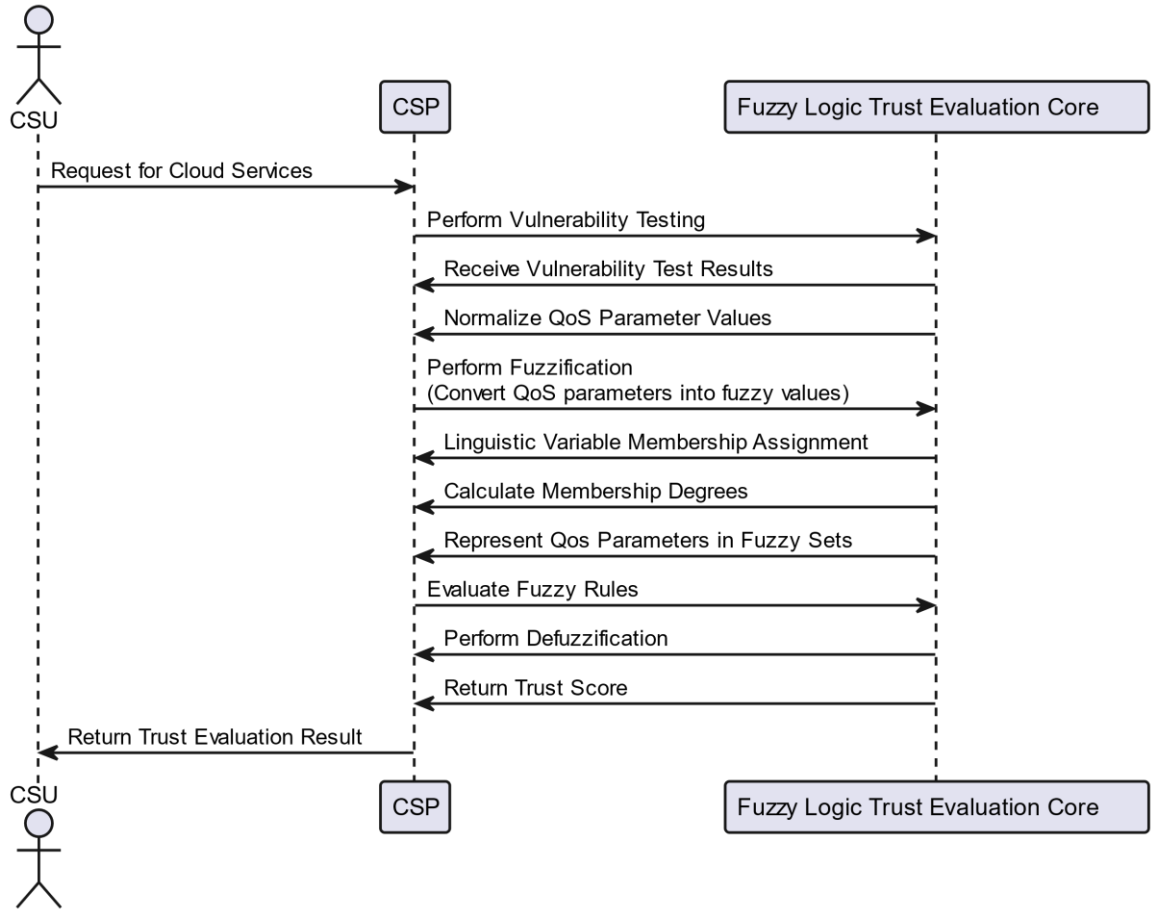

Figure 3.1 Sequence Diagram

3.1.2 ARCHITECTURE DIAGRAM

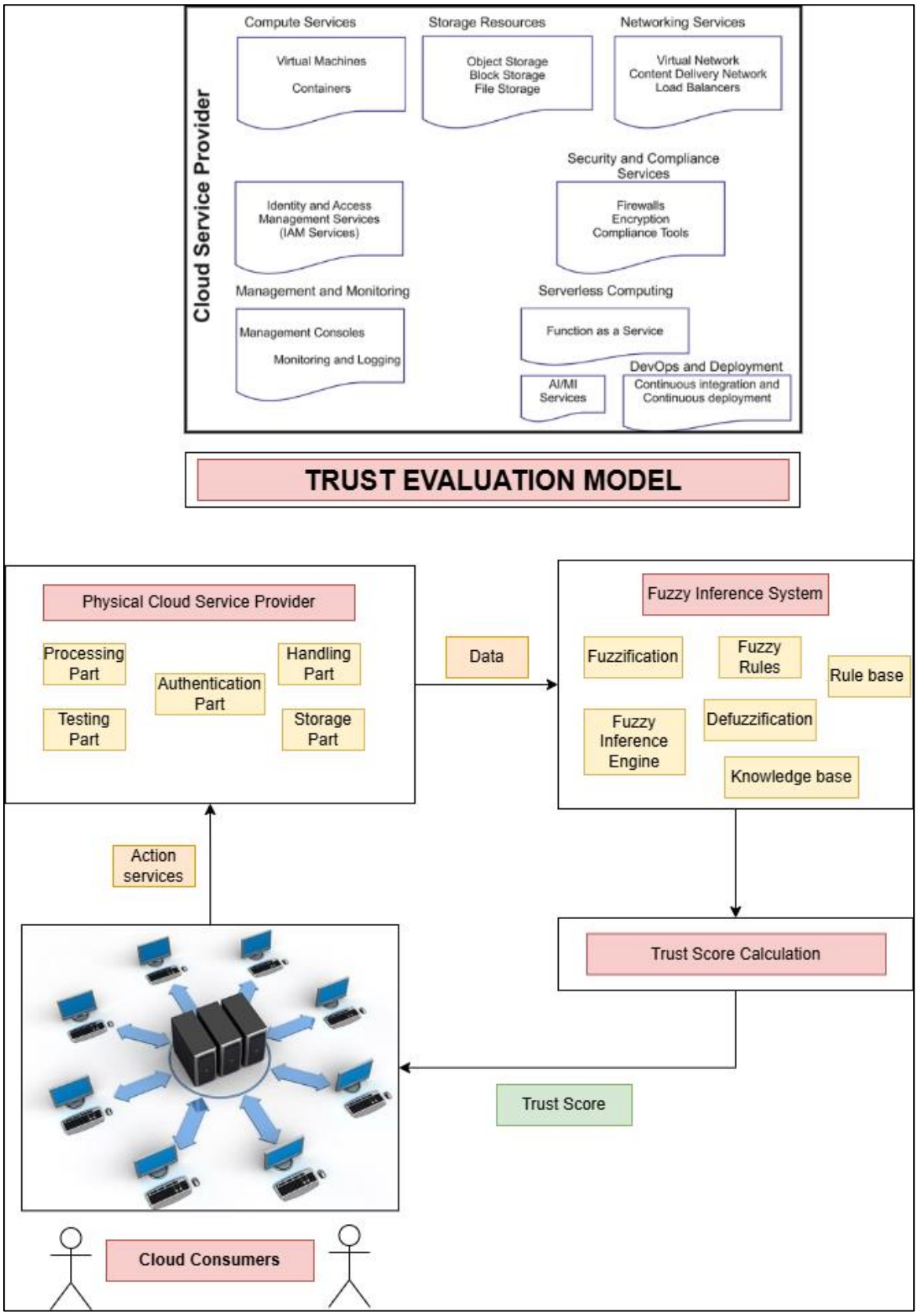

Figure 3.2 Architecture Diagram

3.1.3 TRUST PARAMETER TREE

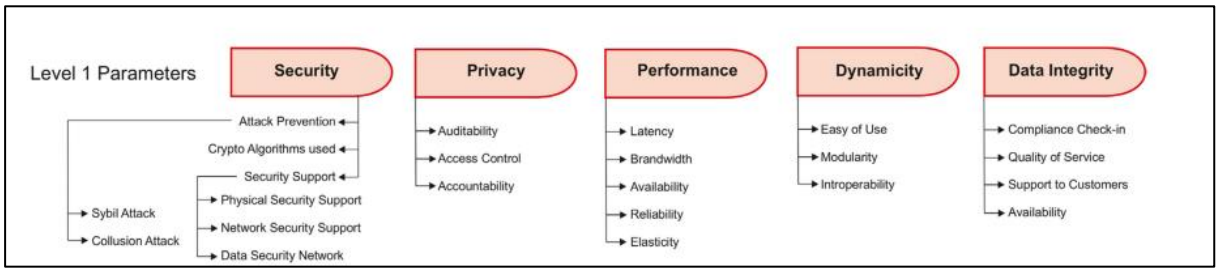

Figure 3.3 Trust Parameter tree

3.1.4 ARCHITECTURE OF FUZZY INFERENCE SYSTEM

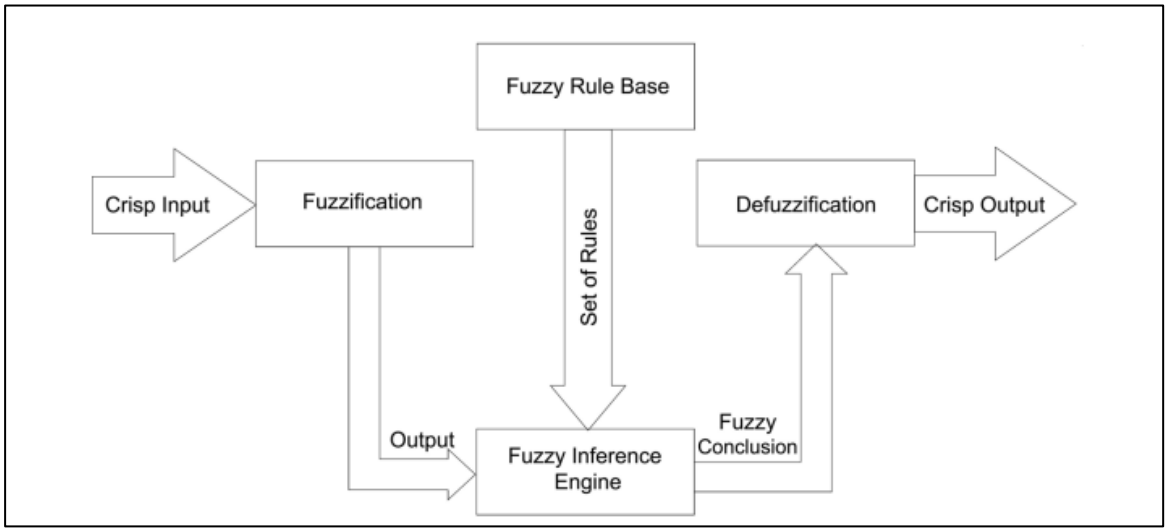

Figure 3.4 Architecture of Fuzzy Inference System

3.2 TRUST SCORE CALCULATION

3.2.1 FUZZY LOGIC-BASED TRUST SCORE EVALUATION

In cloud computing, selecting a reliable Cloud Service Provider (CSP) is essential due to the broad range of available trust models and evaluation approaches. This diversity necessitates a structured and accurate evaluation strategy tailored to varying user needs. To address this, our proposed method introduces a fuzzy logic-driven trust assessment model aimed at streamlining the selection of the most trustworthy CSP. Traditional methods often rely on manual analysis to assess trust, which can be time-intensive and inefficient. In contrast, fuzzy logic offers a more effective solution, particularly when dealing with complex, nonlinear systems. Our framework utilizes a rule-based fuzzy inference system composed of four integral components:

1. **Fuzzifier** – Converts numerical (crisp) input values, such as performance metrics or security ratings, into fuzzy sets.
2. **Inference Engine** – Applies fuzzy reasoning based on a predefined set of rules to derive fuzzy output.
3. **Defuzzifier** – Translates the fuzzy results back into a single crisp trust score for easier interpretation.
4. **Knowledge Base** – Stores both the rule base (if-then rules for decision-making) and the database (membership functions that define the fuzzy terms such as "low", "medium", or "high").

This system enables cloud users to evaluate service providers based on a blend of feedback and performance indicators, enhancing both precision and efficiency in trust calculation. The Working of Fuzzy inference system is shown in Figure 3.4.

3.2.2 BASIC CONCEPTS IN FUZZY LOGIC

Fuzzy logic is an advanced computational approach that deals with reasoning that is approximate rather than fixed or exact. It is particularly effective in systems characterized by uncertainty, such as evaluating trust in cloud service providers. The core elements of fuzzy logic include fuzzy sets, linguistic variables,

membership functions, and a fuzzy inference mechanism. These are explained below with relevant formulas:

1. Fuzzy Sets

A fuzzy set is defined by a membership function that assigns each element a degree of membership ranging from 0 to 1. Formally, a fuzzy set F in a universe of discourse U can be written as:

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

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].

2. Linguistic Variables

A linguistic variable is a variable whose values are not numbers but words or phrases. For example, "Trust" can be described as {Very Low, Low, Medium, High, Very High}. These terms are then associated with fuzzy sets.

Table 3.1 Linguistic Variables and its Membership degree

Linguistic variable	Membership degree	Description
v_h	80–200	Very high
h_z	60–90	High
m_z	30–70	Medium
l_z	10–40	Low
v_l	0–20	Very low

3. Membership Functions

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 3.5. A common shape used in fuzzy systems is the **triangular membership function**, defined as:

$$\mu(x;a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \ge c \end{cases} \quad (2)$$

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.

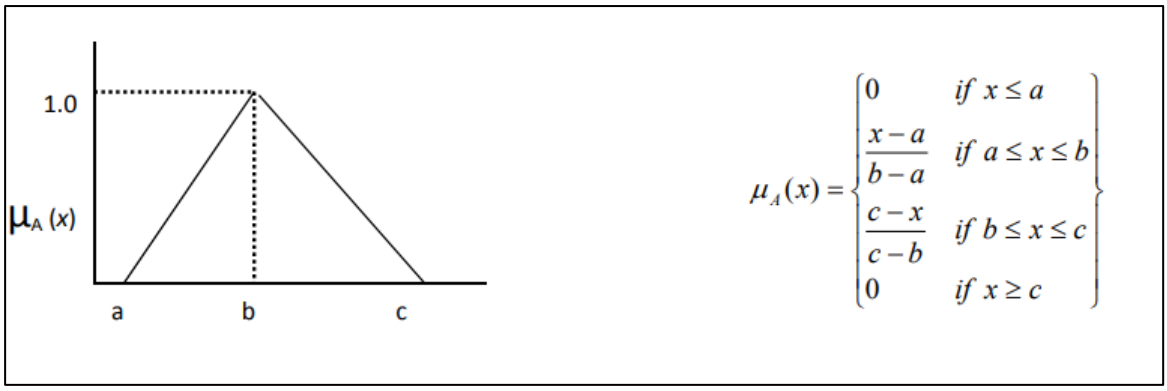

Figure 3.5 Triangular membership function.

4. Fuzzification

Fuzzification is the process of transforming crisp input values into fuzzy values using membership functions. It allows numerical inputs (e.g., security rating = 78) to be converted into fuzzy linguistic values (e.g., "High").

5. Rule Base

The rule base is a collection of fuzzy "if-then" rules. Each rule connects input conditions to an output decision. An example rule might be:

IF Security is High **AND** Performance is Medium, **THEN** Trust is High

These rules are evaluated using fuzzy logic operators such as MIN (for AND), MAX (for OR), and complement (for NOT).

6. Inference Engine

The inference engine applies the rules from the rule base to the fuzzified inputs to determine the fuzzy output. For example, using Mamdani inference, the firing strength of a rule can be calculated as:

$$\alpha = min(\mu_A(x), \mu_B(y)) \quad (3)$$

Where:

- $\mu_A(x)\ and\ \mu_B(y)$ are the membership values for inputs x and y under conditions A and B.

7. Defuzzification

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:

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

Where:

- z_i are output values,
- $\mu_i(z_i)$ are their corresponding membership degrees.

This formula computes the weighted average of the output values, considering the degree to which each rule is activated. By combining these fundamental principles, fuzzy logic systems can evaluate trust in cloud services by handling uncertainty, aggregating multiple trust-related parameters, and providing interpretable output scores for decision-making.

3.2.3 PARAMETER-BASED TRUST SCORE SIMULATION

The proposed trust evaluation approach uses a structured fuzzy inference system (FIS) to simulate trust scores based on a hierarchy of interrelated parameters. In this system, each trust-related factor—such as security, performance, or reliability is modeled with input variables that are processed through a fuzzy logic-based structure.

Initially, the Fuzzy Logic Designer is configured to define each parameter and its role in the overall computation of trust. These parameters are not evaluated independently. Instead, they are calculated using the outputs from their subordinate (sub-parameter) values, resulting in a hierarchical, tree-like structure. For example, the parameter "Security" might derive its value from sub-parameters like "Data Security," "Network Security," and "Physical Security." This layered architecture enhances the model's accuracy by reflecting the influence of granular trust factors on the broader assessment.

3.2.3.1 Identification and Hierarchy of Trust Parameters

The selection of parameters is guided by a comprehensive literature review, which highlights commonly used factors in existing trust models. These include technical aspects like **performance, reliability, availability**, and **security**, as well as threat-related inputs such as **Sybil attacks**, **collusion attacks**, and **data breaches**. Each parameter is assigned a **linguistic variable** (e.g., Low, Medium, High) and a corresponding numeric range that quantifies the degree of trustworthiness.

3.2.3.2 Data Preprocessing

To ensure robust trust score estimation, all raw data—collected from system logs, vulnerability scans, feedback, and efficiency measurements—must undergo preprocessing. This involves:

- **Cleaning:** Handling missing values and outliers to eliminate inconsistencies.
- **Normalization:** Scaling parameter values to a common range to prevent dominance by any one input.

- **Feature Engineering:** Creating additional meaningful inputs from existing data to improve prediction quality.
- **Encoding:** Converting categorical values into numerical formats for processing by the fuzzy logic system.

Table 3.2 Trust parameter with its linguistic variables and its ranges

Trust parameter name	Linguistic variable name with range				
Sybil Attack	CY1 (10, 0, 20)	CY2 (20, 50, 70)	CY3 (60, 100, 110)	–	–
Collusion Attack	CA1 (– 10, 0, 20)	CA2 (20, 50, 70)	CA3 (60, 100, 110)	–	–
Attack Prevention	VL_w (– 10, 0, 20)	L_w (15, 25, 35)	Med (30, 50, 70)	L_g (65, 90, 100)	VL_ge (85, 100, 110)
Physical Security	PSL (10, 0, 10)	PSM (10, 50, 100)	PSH (50, 100, 200)	–	–
Network Security	N_sL (– 10, 0, 10)	N_sM (10, 50, 100)	NsH (50, 100, 200)	–	–
Data Security	DSL (– 10, 0, 10)	DSM (10, 50, 100)	DSH (50, 100, 200)	–	–
Security Support	VL_w (– 10, 0, 20)	L_w (15, 23, 30)	Med (30, 50, 70)	L_g (65, 90, 100)	VL_ge (85, 100, 110)
Crypto Alg used	P_r (– 10, 0, 30)	Avg (20, 50, 70)	Gd (60, 100, 110)	–	–
Security	VL_w (– 10, 0, 20)	L_w (8, 20, 40)	Med (30, 50, 65)	Hgh (50, 80, 100)	VH_ge (85, 100, 110)
Auditability	AL (– 10, 0, 20)	AM (15, 50, 100)	AH (70, 100, 110)	–	–
Access Control	ACL (– 10, 0, 20)	ACM (15, 50, 80)	ACH (60, 100, 110)	–	–
Accountability	AccL (– 10, 0, 30)	AccM (20, 50, 70)	AccH (65, 100, 110)	–	–
Privacy	VL_w (– 10, 0, 10)	L_w (10, 20, 30)	Med (30, 50, 65)	Lge (60, 80, 100)	VL_ge (85, 100, 110)
Availability	AVL (– 10, 0, 30)	AVM (20, 50, 80)	AVH (70, 100, 110)	–	–
Reliability	REL (– 10, 0, 30)	REM (25, 50, 80)	REH (75, 100, 110)	–	–
Elasticity	ElL (– 10, 0, 30)	ElM (25, 50, 80)	Elh (70, 100, 110)	–	–
Bandwidth	BWL (– 10, 0, 35)	BWM (25, 50, 70)	BWH (65, 100, 110)	–	–
Latency	LTL (– 10, 0, 30)	LTM (20, 50, 70)	LTH (60, 100, 110)	–	–
Performance	VL_w (– 10, 0, 15)	L_w (10, 20, 35)	Med (30, 50, 65)	Lge (60, 80, 100)	VL_ge (85, 100, 110)
Ease of use	EOL (– 10, 0, 20)	EOM (15, 50, 100)	EOH (70, 100, 110)	–	–
Modular design	MDL (– 10, 0, 20)	MDM (15, 50, 80)	MDH (60, 100, 110)	–	–
Interoperability	IOL (– 10, 20, 30)	IOM (20, 50, 100)	IOH (65, 100, 110)	–	–
Dynamicity	VL_w (– 10, 0, 10)	L_w (10, 20, 30)	Med (30, 50, 65)	Lge (60, 80, 100)	VL_ge (85, 100, 110)
Compliance checking	CCl (– 10, 0, 20)	CCM (10, 50, 100)	CCH (50, 100, 110)	–	–
Quality of service	QSL (– 10, 0, 20)	QSM (10, 50, 100)	QSH (50, 100, 200)	–	–
Support to customers	SCL (– 10, 0, 20)	SCM (10, 50, 100)	SCH (50, 100, 110)	–	–
Availability of data	AVLD (– 10, 0, 30)	AVMD (25, 50, 65)	AVHD (65, 100, 110	)–	–
Data integrity	VL_w (– 10, 0, 10)	L_w (10, 20, 30)	Med (30, 50, 65)	Lge (60, 80, 100)	VL_ge (85, 100, 110)
Trust score	VL_w (– 10, 0, 20)	L_w (10, 20, 30)	Med (30, 50, 60)	Lge (60, 80, 100)	VL_ge (85, 100, 110)

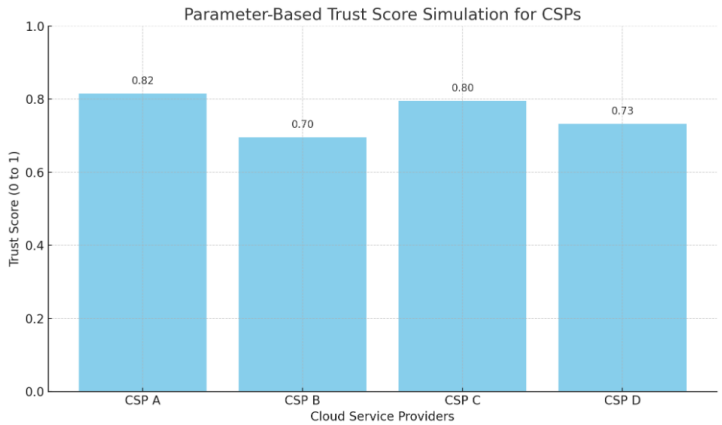

Figure 3.6 Parameter based trust score Simulation

3.2.4 TRUST PARAMETERS EXPLANATION

3.2.4.1 SECURITY

In the context of cloud computing, security refers to the measures and protocols implemented to protect data, applications, and services from unauthorized access, misuse, alteration, or destruction. It ensures that cloud resources are safeguarded against a wide range of threats, including cyber-attacks, data breaches, and system failures. Cloud security encompasses several layers, such as network security, data encryption, identity and access management, and continuous monitoring to prevent vulnerabilities from being exploited by malicious actors.

3.2.4.1.1 Crypto-Algorithm Used

In the evaluation of cloud service providers, cryptographic strength is a vital component of the security trust parameter. The effectiveness of a cryptographic algorithm is measured based on key size, resistance to attacks, performance, and adoption in industry standards. Each algorithm is scored on a scale from -10 to 110. Here's a deeper breakdown of the key factors used to evaluate cryptographic strength in this context:

1. Key Size

- **Definition**: The length of the cryptographic key used in an algorithm, typically measured in bits, which directly impacts the algorithm's security. Larger key sizes generally provide stronger security but may also lead to slower performance.
- **Impact**: A larger key size means that it is more difficult for attackers to brute-force the key, enhancing security.
- **Example**: AES-256 is considered stronger than AES-128 due to its larger key size.

2. Resistance to Attacks

- **Definition**: How well the cryptographic algorithm can withstand various attack methods, such as brute-force attacks, side-channel attacks, or cryptanalysis.

- **Impact**: A resistant algorithm ensures that even if an attacker gains partial information or computational resources, the algorithm will still prevent unauthorized access to the encrypted data.
- **Example**: AES (Advanced Encryption Standard) is widely considered secure because it is resistant to known types of cryptographic attacks, including differential and linear cryptanalysis.

3. Performance

- **Definition**: The efficiency of a cryptographic algorithm in terms of computational overhead, speed, and resource consumption (e.g., CPU, memory, etc.) when performing encryption and decryption operations.
- **Impact**: While security is critical, performance must also be considered. A very secure algorithm may be ineffective for real-time applications if it has too high a computational cost.
- **Example**: RSA encryption, with large key sizes, can be slower than symmetric algorithms like AES, which are designed for higher performance.

4. Adoption in Industry Standards

- **Definition**: The extent to which the cryptographic algorithm is used and endorsed by standards organizations, governments, and major industry players.
- **Impact**: Widespread adoption in trusted industry standards provides assurance that the algorithm has been tested and validated by the global security community.
- **Example**: AES is widely adopted in many protocols and systems, including TLS/SSL, VPNs, and disk encryption, making it a de facto standard for symmetric encryption.

Table 3.3 Crypto Algorithm Evaluation Scoring Table

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

Algorithm	Key Size	Score (-10 - 110)	Reason for Score
			can be implemented in software and hardware, allowing for fast and secure hash calculations. **Preimage Resistance:** Given a hash value, it is computationally infeasible to find the original input message that produced it. This property ensures that an attacker cannot determine the original data from the hash value alone.
SHA-256	256-bit	**100**	Secure, but weaker than SHA-512. Used in SSL/TLS, cryptocurrencies, and digital certificates. it was a joint effort between the NSA and NIST to introduce a successor to the SHA 1 family, which was slowly losing strength against brute force attacks. SHA-256 provides a high level of security, making it practically impossible to derive the original data from its hash value. SHA-256 is a cornerstone of blockchain technology, ensuring the integrity and immutability of blocks. Although rare, there is a theoretical possibility of hash collisions, where two different inputs produce the same hash value.
SHA-1	160-bit	**40**	Weak. Collisions found, meaning attackers can create fake data with the same hash. Deprecated for security use in all cloud providers. SHA1 suffers from collision attacks in that two input strings of dissimilar content result in the same hash value output. This exposes the insecurity of the data and digital signatures. Having the inherent weaknesses, SHA1 has been deprecated by various organizations and is not considered secure anymore. Modern applications moved to more secure algorithms like SHA256.
MD5 (Message Digest 5)	128-bit	**10**	Completely broken. Vulnerable to collisions and should never be used for security purposes. MD5 generates the same hash function for different inputs (**hash collision**). MD5 provides poor security over SHA1, SHA256 and other modern cryptographic algorithms. MD5 has been considered an insecure algorithm. So now we are using SHA256 instead of MD5. MD5 is neither a symmetric nor asymmetric algorithm.

3.2.4.1.2 Physical Security Support

Physical security support refers to the measures and practices a cloud service provider (or data center) implements to protect hardware, software, networks, and data from physical actions and events that could cause serious loss or damage. These include natural disasters, theft, sabotage, and unauthorized access to systems or facilities.

Certifications and Standards

- ISO 27001 (Information Security Management System)
- SOC 2 (Service Organization Control)
- HIPAA (for healthcare data)
- PCI DSS (for payment systems)
- Compliance with country-specific laws or industry standards

Table 3.4 Physical Security Evaluation Scoring Table With References

Certification / Standard	Full Name	Purpose	Score	Reference
ISO/IEC 27001	Information Security Management System	Framework for managing sensitive data security, including physical protection	**+10**	https://www.iso.org/isoiec-27001-information-security.html
SOC 2 Type II	Service Organization Control 2 Type II	Assesses effectiveness of controls over time, incl. physical access, system monitoring	**+10**	https://www.aicpa.org/topic/audit-assurance/service-organization
ISO/IEC 27017	Security Controls for Cloud Services	Provides cloud-specific security guidance including physical infrastructure	**+10**	https://www.iso.org/standard/43757.html
ISO/IEC 27018	Protection of PII in Public Clouds	Focuses on securing personal data stored in cloud systems	**+10**	https://www.iso.org/standard/61498.html

Certification / Standard	Full Name	Purpose	Score	Reference
PCI DSS	Payment Card Industry Data Security Standard	Mandates physical and digital security for cardholder data	**+10**	https://www.pcisecuritystandards.org/
HIPAA	Health Insurance Portability and Accountability Act	U.S. law ensuring physical and administrative protection of health info	**+10**	https://www.hhs.gov/hipaa/index.html
FedRAMP	Federal Risk and Authorization Management Program	Federal cloud security framework incl. physical security standards	**+10**	https://www.fedramp.gov/
CSA STAR	Cloud Security Alliance Security, Trust & Assurance Registry	Cloud-specific framework including physical controls	**+10**	https://cloudsecurityalliance.org/star/
FISMA	Federal Information Security Management Act	Requires federal agencies to maintain physical security of data systems	**+5**	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
TISAX	Trusted Information Security Assessment Exchange	Automotive industry info security incl. site access control	**+5**	https://enx.com/tisax/
ISO/IEC 22301	Business Continuity Management System	Risk planning incl. data center and physical infrastructure resilience	**+5**	https://www.iso.org/standard/75106.html
SSAE 18	Statement on Standards for Attestation Engagements	Internal control evaluations including physical security	**+5**	https://www.aicpa.org/

Certification / Standard	Full Name	Purpose	Score	Reference
NIST SP 800-53	NIST Security and Privacy Controls	Federal-level guidance on physical and logical controls	**+5**	https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final
ITAR	International Traffic in Arms Regulations	Controls access to defense data incl. physical locations	**+5**	https://www.pmddtc.state.gov/ddtc_public?id=ddtc_public_portal_itar_landing
COBIT	Control Objectives for Information and Related Technology	Governance and control model incl. physical security of IT assets	**+5**	https://www.isaca.org/resources/cobit
No Certifications / Audit Failures	–	Indicates lack of physical security planning or failure to meet baseline standards	**-10**	Based on risk assessment best practices

3.2.4.1.3 Network Security Trust Evaluation in Cloud Computing

Network security is a cornerstone of trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs). A robust network security posture not only safeguards digital assets but also elevates the reliability and reputation of cloud services.

Network security in cloud computing refers to the set of policies, configurations, and technologies deployed to protect cloud infrastructure from unauthorized access, breaches, DDoS attacks, and internal misuse.

Core Security Controls Include:

- Firewall rules for ingress (incoming) and egress (outgoing) traffic
- Access control based on IP ranges
- Protocol restrictions (e.g., denying Telnet, FTP, SMB)
- Monitoring and alerting for anomalous or suspicious activity

Security Scoring

The trust score is computed using a deduction-based model where misconfigurations or risky policies are penalized based on severity. The deductions feed into the fuzzy trust logic, determining the CSP's security posture**.**

Table 3.5 Ingress Firewall Rules

Condition	**Port Examples**	**Risk Level**	**Deduction**
Publicly open to all (0.0.0.0/0) on high-risk ports	SSH (22), RDP (3389), Flask (5000), Gophish (3333)	Critical – vulnerable to brute-force, RCE	-20
Publicly open to all (0.0.0.0/0) on medium-risk ports	HTTP (80), iperf (5201)	High – web-facing, potential for DDoS or exploits	-10
Publicly open to all (0.0.0.0/0) on low-risk ports	HTTPS (443)	Moderate – expected but should be monitored	-5 *(or 0 if intentional)*
Public ICMP (Ping) enabled	ICMP	High – enables network scanning	-10
Access restricted to internal IPs	Private CIDRs (e.g., 10.0.0.0/8)	Low – reduced exposure	-2 to -5
No ingress rules (deny-all)	-	Best Practice – zero attack surface	No deduction

Table 3.6 Egress Firewall Rules

Condition	**Port Examples**	**Risk Level**	**Deduction**
Allow all outbound traffic (all protocols)	All	Critical – exfiltration, malware risk	-15
Restricted to essential services	HTTPS (443), HTTP (80), DNS (53)	Low – good containment	-2 to -5
Default deny, explicitly allowed only	-	Best Security	No deduction

Table 3.7 General Firewall and Network Configuration

Condition	Risk Level	Deduction
Public ICMP access	High – exposes network info	-10
Duplicate rules for the same port/service	Medium – may cause mismanagement	-5
Default VPC with auto subnetting	Medium – often overly open	-5
No firewall rules (implied allow-all)	Critical – full exposure	-30

Table 3.8 Scoping to target service accounts

Condition	Adjustment	Rationale
Scoped with target_tags or target_service_accounts	+5	Reduces exposure; applied only to specific resources

3.2.4.1.4 Data Security Support

Data Security Support in Cloud refers to the implementation and enforcement of security mechanisms that ensure confidentiality, integrity, and availability of data stored, processed, and transmitted within cloud environments. It involves the use of advanced cryptographic techniques, access control policies, authentication mechanisms, and secure communication protocols to protect sensitive information against unauthorized access, breaches, and cyber threats.

In cloud computing, data security support plays a critical role in building trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs), especially in sectors like healthcare, finance, and government services. The effectiveness of data security is often evaluated based on how well the cloud service provider implements encryption, hashing, and digital signature techniques to secure data both **at rest** and **in transit**.

Scoring of Data Security Algorithms in Cloud Trust Evaluation

To evaluate the **trustworthiness** of a cloud service provider's data security, algorithms are scored based on key parameters like: Key Size (bit length of the cryptographic key), Strength Against Known Attacks, Efficiency and Speed, Adoption and Usage in the Industry, Suitability for Cloud Environments as depicted in Table 3.3.

3.2.4.1.5 Attack Prevention:

Attack prevention refers to the strategies, techniques, and mechanisms put in place to prevent or mitigate the likelihood of security attacks targeting cloud computing resources. These measures aim to stop attackers from gaining unauthorized access to systems and data by using firewalls, intrusion detection systems, encryption, multi-factor authentication, secure coding practices, and continuous system monitoring. Attack prevention focuses on detecting potential threats before they can exploit vulnerabilities and cause harm, ensuring that cloud services remain resilient to attacks like Distributed Denial of Service (DDoS), data breaches, and malware. Both security and attack prevention are fundamental aspects of building trust and reliability in cloud computing systems, particularly when evaluating Cloud Service Providers (CSPs) for potential adoption in IT and healthcare industries.

3.2.4.1.6 Sybil-Attack

A Sybil attack is a type of security threat in which a malicious entity creates multiple fake identities or nodes (called Sybils) within a network to gain a disproportionate level of influence or control. These identities pretend to be legitimate users or devices, undermining the reliability and integrity of the network. In cloud computing and other distributed systems like peer-to-peer networks, blockchain, or IoT environments, trust is often built on the assumption that each participating node is unique and behaves honestly. A Sybil attack violates this assumption by overwhelming the system with numerous pseudonymous identities controlled by a single adversary.

This attack can be particularly dangerous because:

- It can manipulate voting mechanisms in consensus-based systems.
- It can disrupt reputation-based trust models by generating fake feedback or ratings.
- It can intercept, alter, or reroute data traffic, leading to data breaches or service interruptions.
- In cloud-based systems, it may affect resource allocation, deny access to genuine users, or even hide the presence of a real attack under the disguise of multiple Sybil identities.

3.2.4.1.7 Collision Attack

A collusion attack in the context of cloud computing occurs when multiple malicious parties, such as cloud service users or even cloud service providers, work together in a coordinated manner to compromise security, exploit vulnerabilities, or bypass access controls. This can happen in various forms, such as:

1. Data Breach: Malicious users may collude to gain unauthorized access to sensitive data stored in the cloud, sharing credentials or exploiting system weaknesses to bypass security measures.
2. Denial of Service (DoS): Attackers may coordinate an attack to overwhelm cloud resources (e.g., bandwidth, processing capacity), leading to service unavailability for legitimate users.
3. Resource Misuse: In a cloud environment, attackers may collude to misuse resources or manipulate the cloud platform for financial or other benefits, such as running unauthorized workloads on cloud infrastructure or consuming more resources than allowed.
4. Cloud Service Provider Compromise: If a cloud service provider itself is compromised (either internally or through collusion with other entities), it could lead to wide-scale vulnerabilities and compromises across multiple customers' data.

Collusion attacks are particularly dangerous because they involve coordinated efforts to bypass traditional security mechanisms, making detection and mitigation much more challenging.

3.2.4.2 PRIVACY

Privacy in cloud computing refers to the protection and appropriate handling of users' personal, sensitive, and confidential data within a cloud environment. It ensures that data is accessed, processed, and stored only by authorized entities while maintaining user confidentiality and compliance with applicable laws and standards. Trust in cloud services is strongly influenced by the level of privacy they guarantee.

3.2.4.2.1 Access Control in Privacy

Access Control is a fundamental mechanism in ensuring data privacy in cloud environments. Access Control in cloud computing ensures that only authorized users and service accounts have access to specific cloud resources under clearly

defined conditions. It plays a crucial role in safeguarding data privacy, enforcing compliance, and minimizing the risk of breaches.

Table 3.9 Privacy Evaluation Criteria

Category	Threshold/Criteria	Score (Max)	Evaluation
Least Privilege Enforcement	No excessive permissions beyond what's necessary.	25	Presence of roles/editor and roles/owner is limited
Sensitive Data Access Restriction	No broad roles/editor or roles/owner to many users.	25	roles/editor and roles/owner assigned to multiple accounts, reducing security.
Minimal Service Account Exposure	Only required service accounts exist, with limited external exposure.	25	Multiple service accounts detected, indicating the need for tighter control.
Audit Logging for Access	Tracks who accessed sensitive resources.	20	Logging mechanism is present and functioning as expected.
No Hardcoded or Long-lived Credentials	No exposed credentials or static keys in use.	25	No direct evidence of hardcoded credentials.

3.2.4.2.2 Auditability

Auditability in cloud computing is the system's ability to track, log, and assess all user and service activities in a transparent and verifiable manner. It is vital for security, compliance, and governance, particularly in regulated industries like healthcare and finance.

This concept is implemented by capturing audit logs (e.g., via GCP's Cloud Audit Logs) and assigning risk-impact scores based on the nature of each action. These logs help in detecting misconfigurations, privilege escalations, or unauthorized access, and play a crucial role in incident response and forensic investigations.

Key Features

- **Transparency:** All significant events are logged and reviewable.
- **Accountability:** Links actions to users or service accounts.
- **Traceability:** Ensures a timeline of events is available for compliance audits.

- **Risk Scoring:** Supports real-time trust evaluation using automated rule-based analysis.

Table 3.10 Auditability Evaluation criteria

Action Detected	**Impact on Privacy/Security**	**Score Adjustment**
iam.serviceAccounts.actAs	Medium risk – Privilege escalation potential	-25
ssh-keys in metadata	High risk – Unauthorized SSH access	-35
principalEmail using gmail.com	Moderate risk – Personal identity used	-20
iam.serviceAccounts.getAccessToken	Increases risk – Possible repetitive misuse	-15
compute.instances.setMetadata	High risk – Metadata tampering	-30
"external IP" in payload	Moderate risk – Sensitive data exposure possible	-10
No log activity detected	Safe baseline	+110

3.2.4.2.3 Accountability

Accountability in cloud computing refers to the responsibility of the cloud service provider (CSP) to ensure that data is handled in compliance with privacy policies, legal regulations, and user expectations. It ensures that any misuse, unauthorized access, or data breaches can be traced back to the responsible entity. This includes maintaining detailed audit trails, performing regular compliance checks, and being transparent about how user data is stored, processed, and shared. A trustworthy CSP should implement mechanisms that support auditability, responsibility tracking, and policy enforcement to uphold user trust and data privacy.

3.2.4.3 PERFORMANCE

3.2.4.3.1 Latency

Latency in cloud computing refers to the time delay experienced in a system, specifically the time taken for a data packet to travel from the source (client) to the destination (server) and back. It is typically measured in milliseconds (ms). Lower latency indicates faster and more efficient service response times, which is crucial for real-time and performance-sensitive applications, especially in the healthcare and IT sectors.

Table 3.11 Latency Evaluation Criteria

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

3.2.4.3.2 Availability

Availability in cloud computing refers to the proportion of time that a cloud service is operational and accessible to users within a specified period. High availability ensures continuous and uninterrupted access to cloud-based resources, which is particularly critical in industries like IT and healthcare where downtime can have serious consequences.

In this study, availability is measured as the percentage of total uptime over a 30-day period using data collected from Google Cloud Monitoring API. The script utilizes the metric compute.googleapis.com/instance/uptime to evaluate how long virtual machine instances have been running without interruption.

Calculation Methodology

1. **Time Frame**: The monitoring window is set to the last 30 days (43,200 minutes).
2. **Uptime Data Collection**: Uptime metrics are retrieved using the API (monitoring_v3.MetricServiceClient() from Google Cloud's Monitoring API).
3. **Aggregation**: Uptime for each instance is averaged to ensure fairness across multiple cloud services.
4. **Conversion and Calculation**: Uptime is converted from seconds to minutes and compared to the total time to calculate availability as a percentage.

Formula Used

$$\text{Availability (\%)} = \left(\frac{\text{Total Uptime (minutes)}}{43,200} \right) \times 100$$

3.2.4.3.3 Bandwidth

Bandwidth in cloud computing refers to the maximum rate of data transfer across a network path. It is a crucial Quality of Service (QoS) metric that determines how quickly data can move between a client and a cloud service provider (CSP). Bandwidth is typically measured in megabits per second (Mbps) and directly affects the speed and efficiency of cloud-based applications, particularly those dealing with large datasets, real-time communication, and high availability. In the context of this project, bandwidth is evaluated by downloading a sample file and measuring the download time to estimate the available network speed.

Table 3.12 Bandwidth Performance Evaluation Rules

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

3.2.4.3.4 Reliability

Reliability in cloud computing refers to the consistency and dependability of a cloud service provider (CSP) in delivering services without failure or degradation over time. A highly reliable cloud service ensures minimal downtime, fast response times, stable system behavior, and consistent performance. It is especially crucial in environments like IT infrastructure and healthcare, where uninterrupted access to data and applications is critical.

In this project, reliability is quantitatively assessed using key performance indicators such as latency, jitter, packet loss, CPU load, and disk latency - all of which contribute to the overall Quality of Service (QoS) in a cloud environment.

1. **Network Metrics (Ping-based)**:
    - **Latency**: Measures average response time.
    - **Jitter**: Measures variation in latency.

- **Packet Loss**: Determines the percentage of lost packets.

2. **System Health Metrics (psutil-based)**:
    - **CPU Load**: Measures average CPU usage.
    - **Disk Latency**: Evaluates I/O performance from read/write operations.
3. **Scoring System**:
    - Each component is scored based on thresholds.
    - Maximum reliability score = **110**.
    - Helps assess how stable and dependable the cloud VM or service is.

Table 3.13 Reliability Scoring Rules

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

3.2.4.3.5 Elasticity

Elasticity in cloud computing refers to the ability of a system to automatically adapt to workload changes by provisioning or de-provisioning resources in real-time to match demand. It ensures optimal resource utilization, cost-efficiency, and consistent performance without manual intervention. In the context of evaluating the trustworthiness of a cloud service provider, elasticity is a Quality of Service (QoS) factor that reflects how well the infrastructure can scale dynamically and maintain performance under varying workloads. A highly elastic cloud service ensures seamless scaling and stability, which builds user trust and system reliability. The Python script evaluates elasticity using Google Cloud's Monitoring and Compute APIs. It checks the average CPU utilization over the last 5 minutes and determines whether autoscaling is enabled for the instance group.

Table 3.14 Elasticity Evaluation Rules

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
Total Score		**Max: 110**

3.2.4.4 DYNAMICITY

In the context of cloud computing, dynamicity refers to the ability of a cloud service to adapt and respond to changes in the environment, workload, and user requirements in real-time. It involves the flexible allocation and deallocation of resources, such as processing power, storage, and network bandwidth, to meet the varying demands of users. Dynamicity ensures that cloud services can maintain optimal performance, scalability, and resource efficiency under fluctuating conditions, enhancing the overall reliability and responsiveness of the system. This characteristic is particularly important for applications requiring real-time adjustments or handling sudden spikes in demand.

3.2.4.4.1 Ease Of Use

Ease of Use in Dynamicity refers to the simplicity with which users can interact with and adapt to the dynamic nature of cloud computing environments. As cloud services continuously scale, reconfigure, or adjust based on changing workloads and user demands, it is essential that these transitions remain seamless and intuitive for users. This characteristic ensures that the underlying dynamic behaviors of the system—such as auto-scaling, load balancing, and resource reallocation—do not complicate user interactions or require extensive technical knowledge. High ease of use in dynamicity enhances user satisfaction by allowing smooth access and control over dynamically changing resources with minimal effort or disruption.

3.2.4.4.2 Modularity

Modularity in Dynamicity refers to the design approach in cloud systems where services and components are organized into independent, interchangeable modules that can dynamically adjust or scale based on user requirements and system conditions. In a dynamic cloud environment, modularity allows individual components to be added, removed, upgraded, or reconfigured without affecting the entire system. This enhances flexibility, simplifies maintenance, and enables efficient resource management, ensuring that changes in workload or functionality can be accommodated swiftly and seamlessly.

3.2.4.4.3 Interoperability

Interoperability in Dynamicity refers to the ability of dynamically changing cloud services and components to seamlessly interact, integrate, and function across different platforms, technologies, and service providers. In a dynamic cloud environment where resources are frequently scaled or reconfigured, interoperability ensures that these changes do not hinder communication or compatibility between systems. It enables diverse cloud services to work together effectively, promoting flexibility, reducing vendor lock-in, and enhancing the overall adaptability of the cloud infrastructure to meet evolving user needs.

3.2.4.5 DATA INTEGRITY

Data Integrity refers to the accuracy, consistency, and reliability of data stored and transmitted within a cloud computing environment. It ensures that data remains unaltered during operations such as storage, retrieval, and transfer, and is protected against unauthorized modification, corruption, or loss. In the context of

cloud computing, maintaining data integrity is critical for building trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs), especially in sensitive domains like healthcare and IT. Strong data integrity mechanisms help guarantee that users can rely on the cloud to handle their data securely and accurately at all times.

3.2.4.5.1 Compliance Checking

Compliance Checking in Data **Integrity** refers to the process of ensuring that data stored, processed, and transmitted within a cloud computing environment adheres to predefined policies, standards, and regulatory requirements. It plays a crucial role in maintaining data integrity by verifying that all operations on the data conform to legal, organizational, and industry-specific guidelines, such as GDPR, HIPAA, or ISO standards. In cloud systems, compliance checking helps identify deviations or unauthorized activities that may compromise the trustworthiness and accuracy of the data, thereby reinforcing accountability and transparency between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs).

3.2.4.5.2 Quality Of Service

Quality of Service (QoS) in Data Integrity refers to the ability of a cloud system to maintain high standards of data accuracy, consistency, and reliability throughout its lifecycle—from storage and processing to transmission. It involves implementing robust mechanisms such as encryption, access control, validation protocols, and error detection to ensure that the data remains unaltered and trustworthy. In the context of cloud computing, QoS in data integrity guarantees that users can depend on the cloud service to deliver secure and error-free data, even under varying workloads or network conditions, thereby building trust between Cloud Service Providers (CSPs) and Cloud Service Users (CSUs).

3.2.4.5.3 Support To Customers

Support to Customers in Data Integrity refers to the cloud service provider's ability to assist users in ensuring their data remains accurate, consistent, and secure throughout its lifecycle. This includes providing timely technical support, clear documentation, data recovery services, audit trails, and guidance on maintaining compliance with integrity-related standards. Effective customer support builds trust by enabling Cloud Service Users (CSUs) to detect, report, and resolve data integrity issues promptly, thereby ensuring uninterrupted and reliable access to trustworthy data in the cloud environment.

3.2.4.5.4 Availability

Availability in Data Integrity refers to the assurance that data is consistently accessible and usable by authorized users whenever needed, without disruption or unauthorized alteration. In the context of cloud computing, it implies that the cloud service provider guarantees reliable uptime, redundancy, and quick data recovery mechanisms to ensure data remains intact and continuously available. High availability supports data integrity by preventing data loss or corruption due to system failures, outages, or cyber-attacks.

CHAPTER 4
SYSTEM SPECIFICATION, IMPLEMENTATION AND RESULTS

4.1 SYSTEM SPECIFICATION

4.1.1 SOFTWARE REQUIREMENTS

Operating System	Windows 11
Tool	MATLAB
Cloud Service Providers	Google Cloud Platform (GCP), Amazon Web Services (AWS)
Coding Language	Python

4.1.2 HARDWARE REQUIREMENTS

CPU	12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz
RAM	8.00 GB
SYSTEM TYPE	64-bit operating system, x64-based processor

4.2 SOFTWARE DESCRIPTION

4.2.1 About MATLAB

MATLAB (an abbreviation of "MATrix LABoratory") is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. MATLAB is a high-performance programming language and computing environment designed primarily for numerical computing, data analysis, and algorithm development. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

4.2.2 About GCP

Google Cloud Platform (GCP) is a suite of cloud computing services developed by Google that provides infrastructure, platform, and software services to businesses, developers, and organizations worldwide. Launched in 2008, GCP allows users to build, test, and deploy applications on Google's highly reliable and scalable infrastructure. Key services include Google Compute Engine for virtual machines, Google Cloud Storage for object storage, BigQuery for data analytics, and Firebase for mobile and web app development. GCP also offers robust machine learning and artificial intelligence tools such as Vertex AI and AutoML. Known for its advanced data analytics, security, and integration with open-source technologies, GCP is widely used for cloud-native application development, big data processing, and AI/ML model deployment. While GCP is highly efficient and performance-oriented, it is often considered more complex for beginners compared to other cloud providers. However, it remains a powerful choice, especially for data-intensive and AI-driven applications.

4.2.3 About AWS

Amazon Web Services (AWS) is a comprehensive cloud computing platform developed by Amazon, launched in 2006. It offers a broad set of services including computing power, storage solutions, database management, machine learning, analytics, and more through a flexible, scalable, and secure cloud infrastructure. AWS provides services such as EC2 for virtual computing, S3 for scalable storage, RDS for relational databases, and Lambda for serverless computing. It supports a wide range of applications from web and mobile development to big data processing and AI model deployment. AWS operates on a pay-as-you-go pricing model, making it cost-effective for both startups and large enterprises. With its global network of data centers, high reliability, and extensive security features, AWS has become one of the most widely adopted cloud platforms in the world. However, due to its vast range of services, it may have a steep learning curve, and costs can increase if not properly managed.

4.3 IMPLEMENTATION

4.3.1 IMPLEMENTATION IN GCP

4.3.1.1 SECURITY

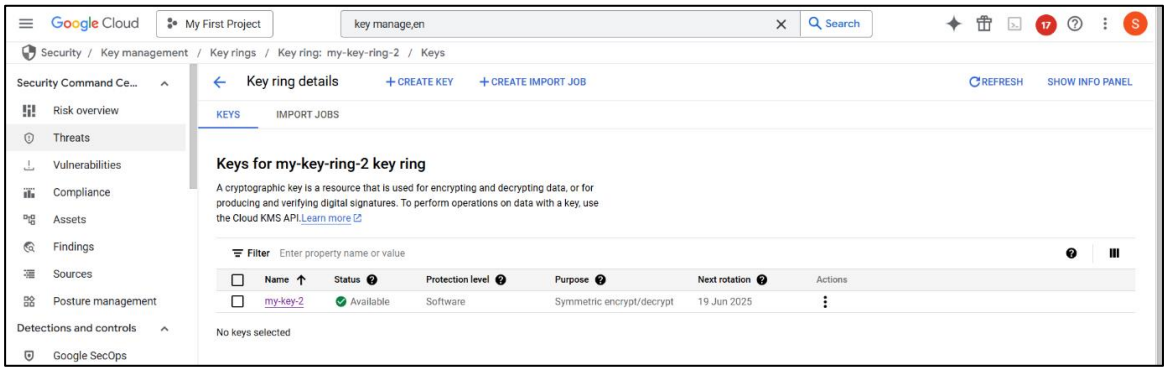

Figure 4.1 Crypto algorithm used

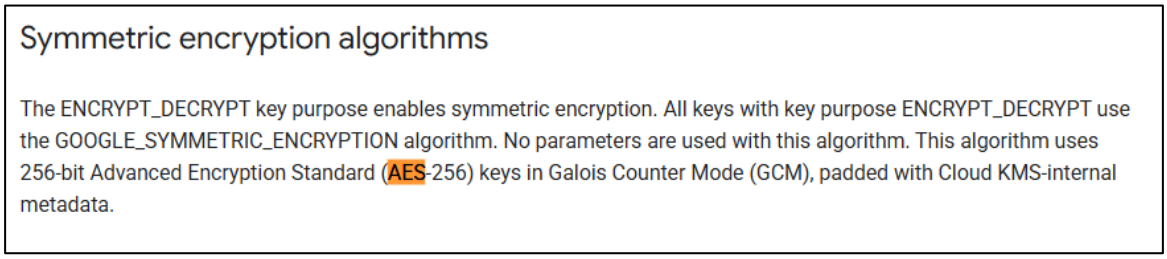

Figure 4.2 Encryption Algorithms in GCP

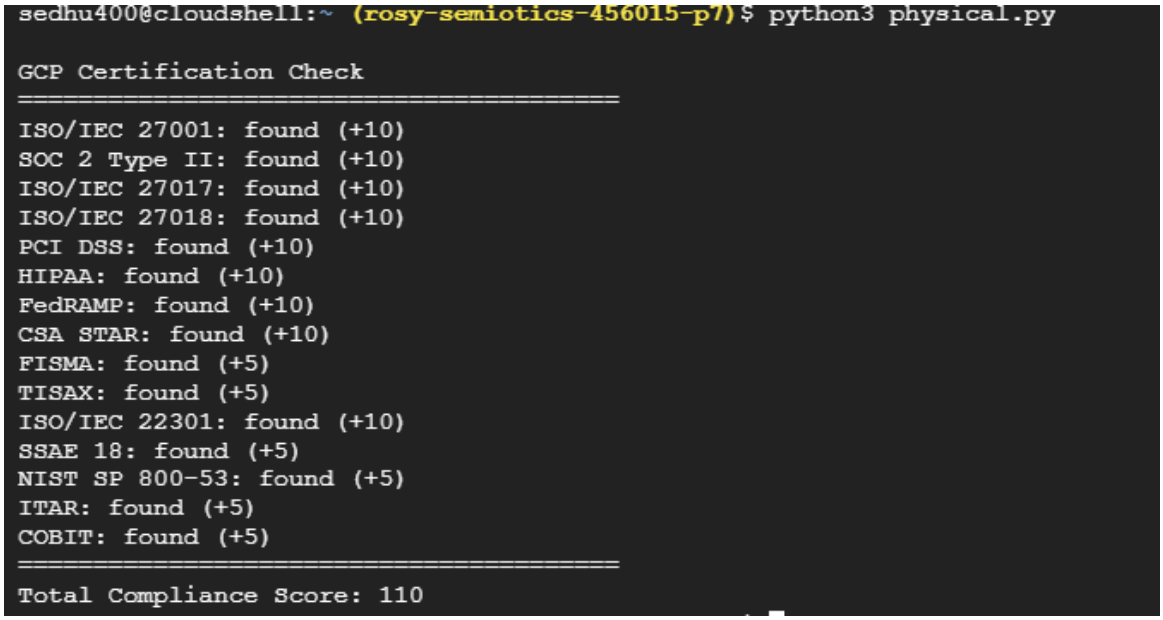

Figure 4.3 Physical Security

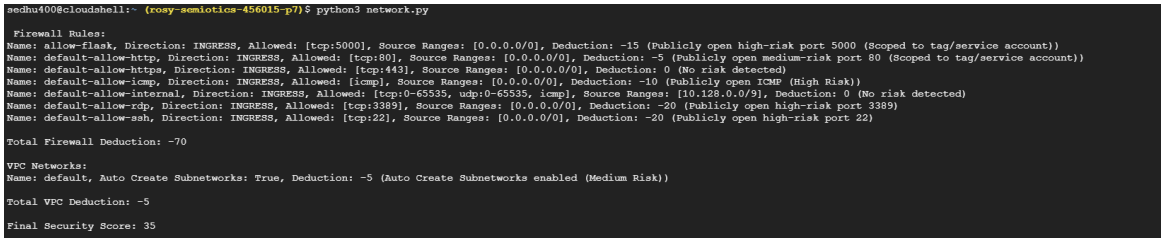

Figure 4.4 Network Security

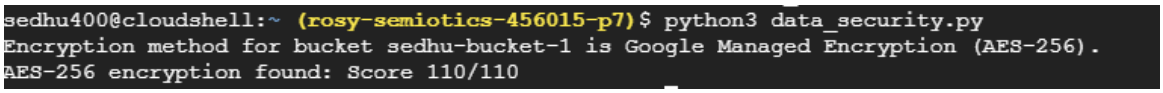

Figure 4.5 Data Security

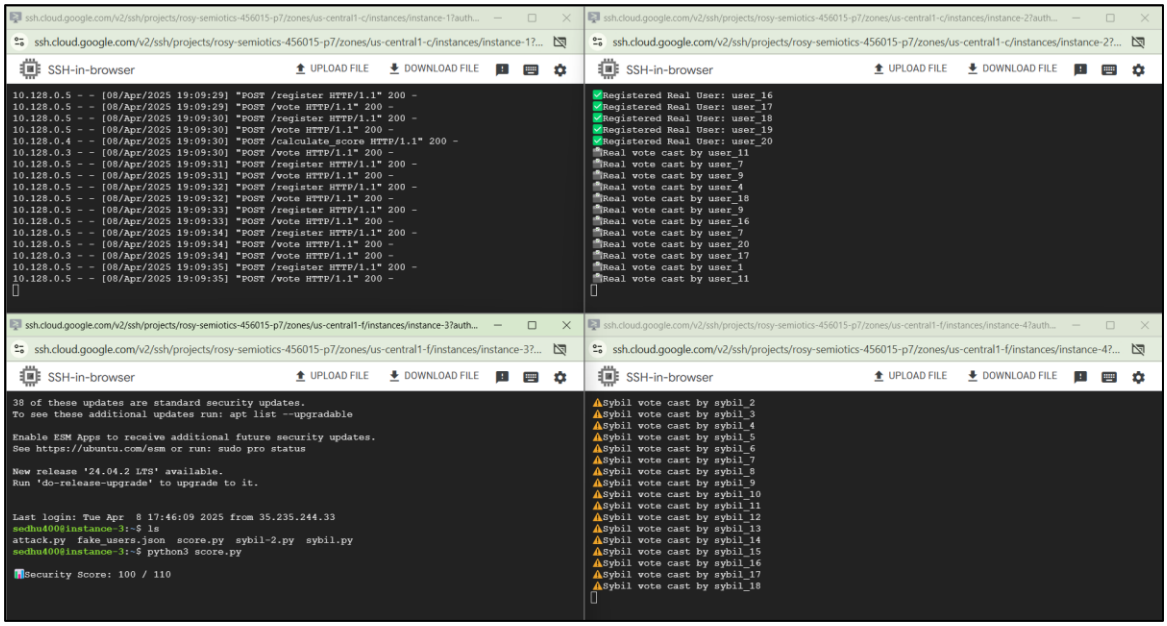

Figure 4.6 Sybil attack

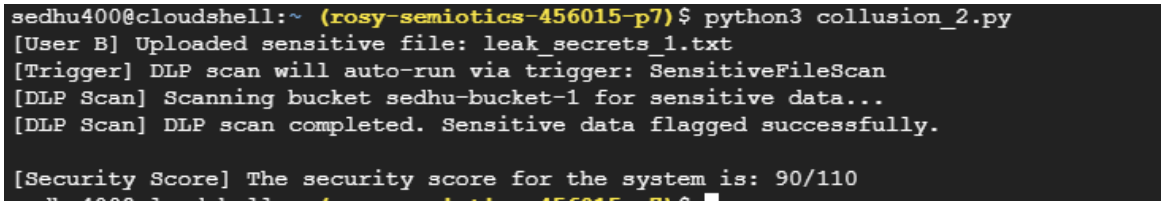

Figure 4.7 Collusion attack

4.3.1.2 PRIVACY

Figure 4.8 Access Control

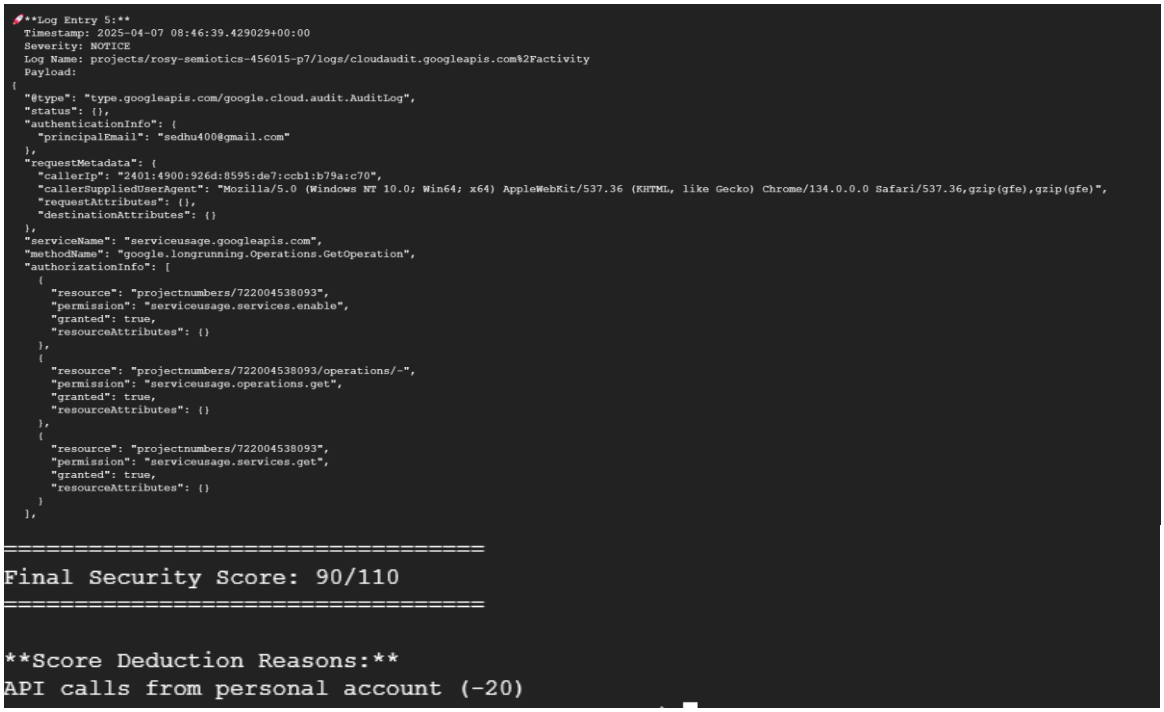

Figure 4.9 Auditability

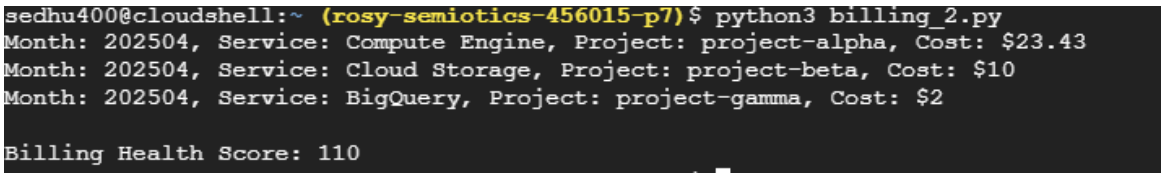

Figure 4.10 Accountability

4.3.1.3 PERFORMANCE

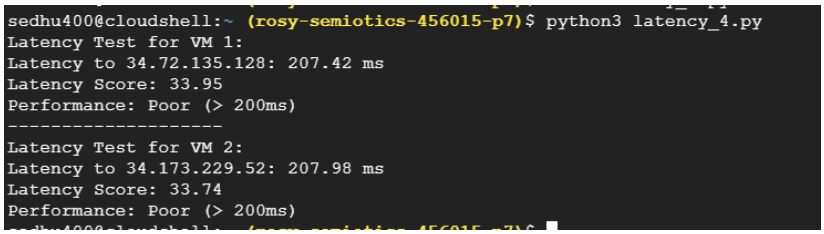

Figure 4.11 Latency

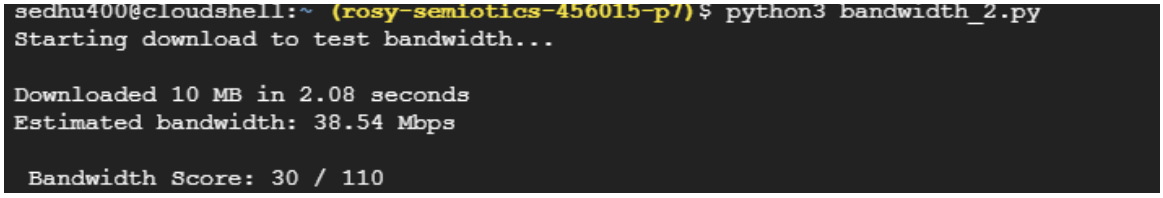

Figure 4.12 Bandwidth

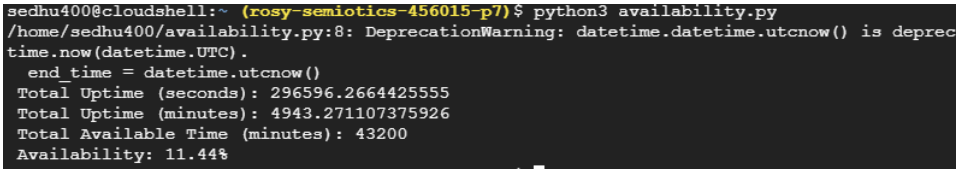

Figure 4.13 Availability

```
sedhu400@cloudshell:~ (rosy-semiotics-456015-p7)$ python3 rel_4.py

 Pinging 8.8.8.8 10 times...
 Checking system load and disk performance...

 Metrics:
 - Latency      : 0.72 ms
 - Jitter       : 0.85 ms
 - Packet Loss  : 0%
 - CPU Load     : 0.50%
 - Disk Latency: 1.67 ms/op

 GCP VM Reliability Score: 110 / 110
```

Figure 4.14 Reliability

```
sedhu400@cloudshell:~ (rosy-semiotics-456015-p7)$ python3 elasticity-1.py
🚀Calculating Elasticity Score...

📊CPU Usage (5-min average): 4.51%
⚙️ Autoscaling Enabled: Yes

📈Elasticity Score: 110/110
```

Figure 4.15 Elasticity

4.3.1.4 DYNAMICITY

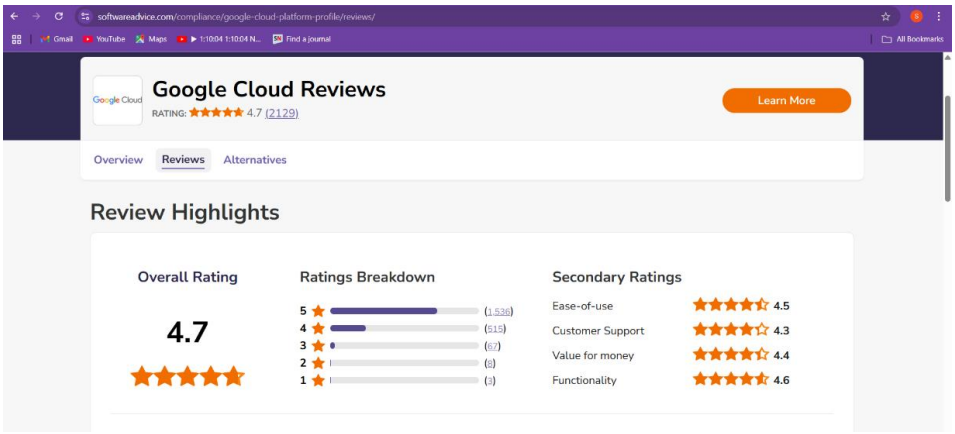

Figure 4.16 Ease of Use

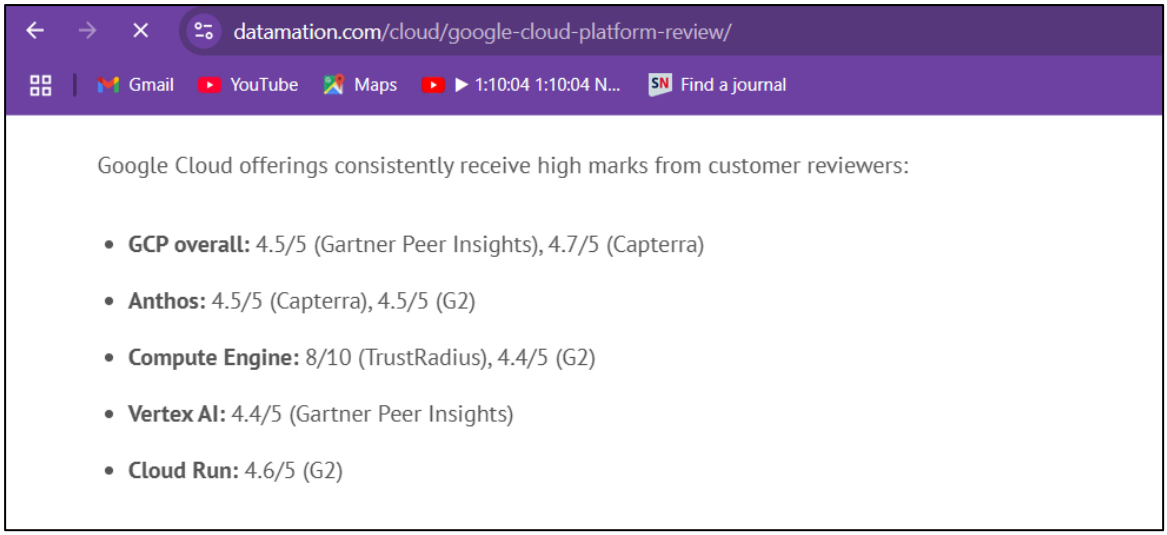

Figure 4.17 Modularity

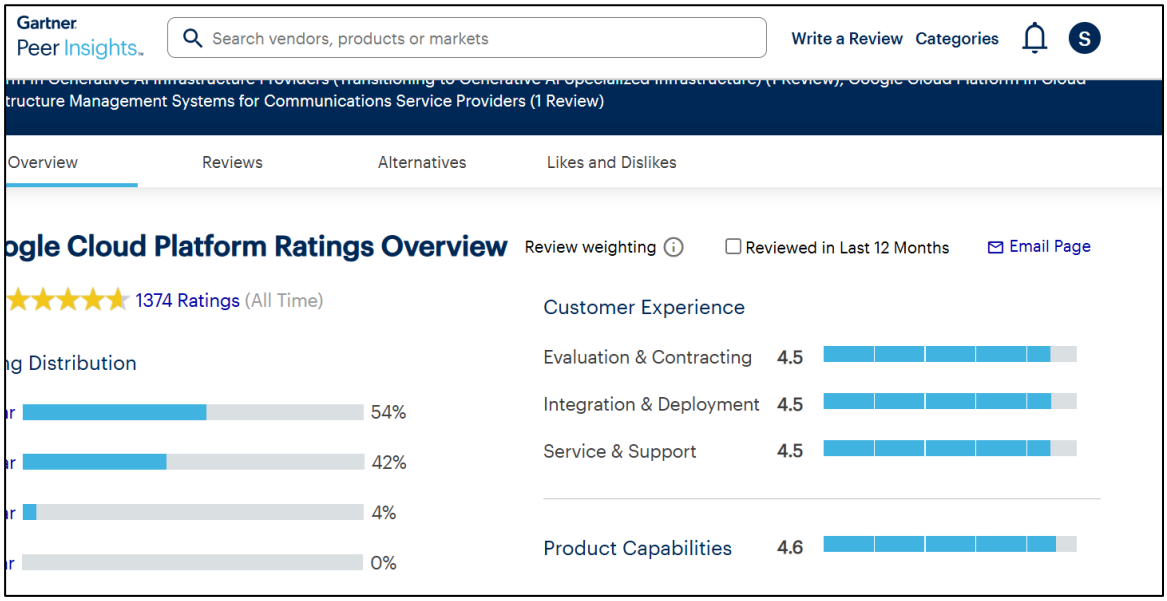

Figure 4.18 Interoperability

4.3.1.5 DATA INTEGRITY

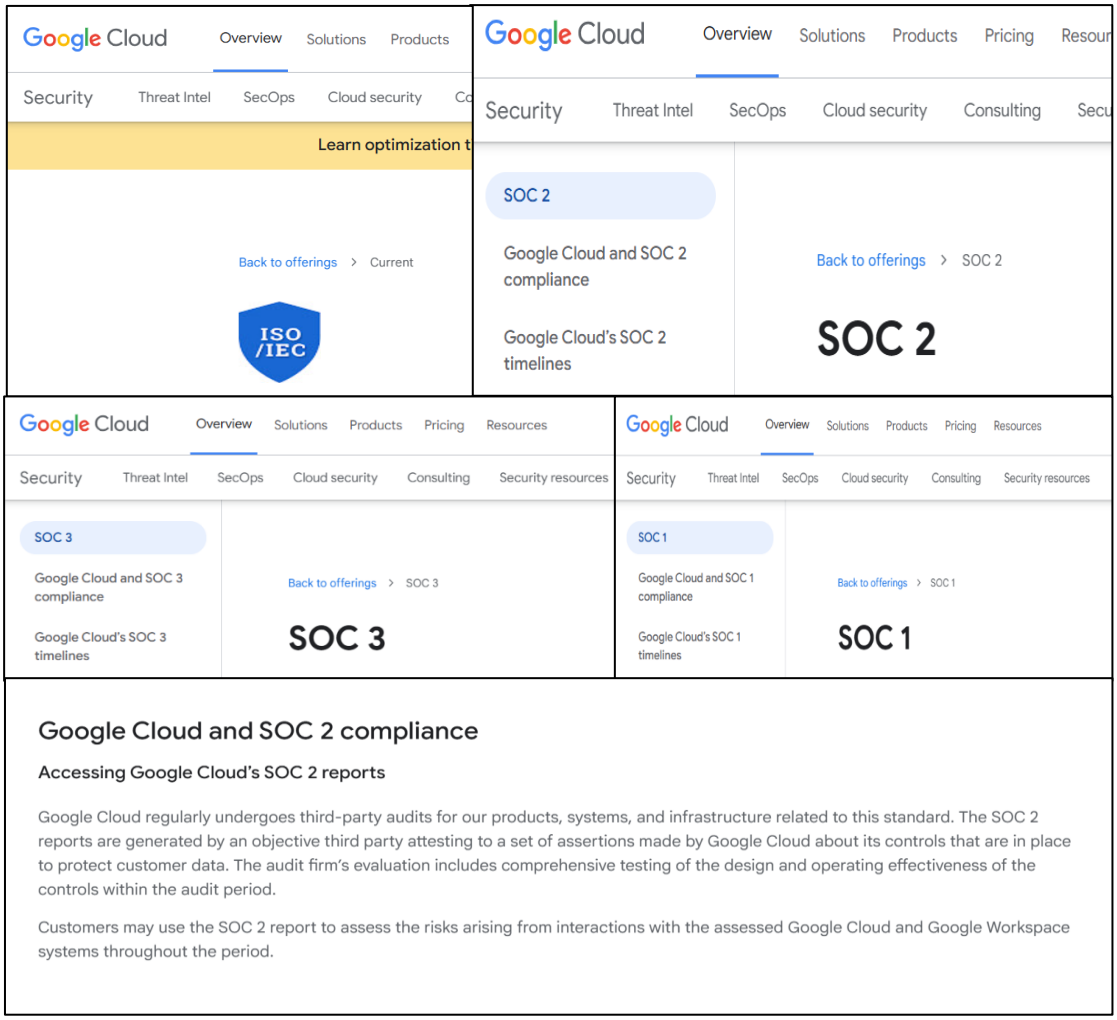

Figure 4.19 Compliance Checking

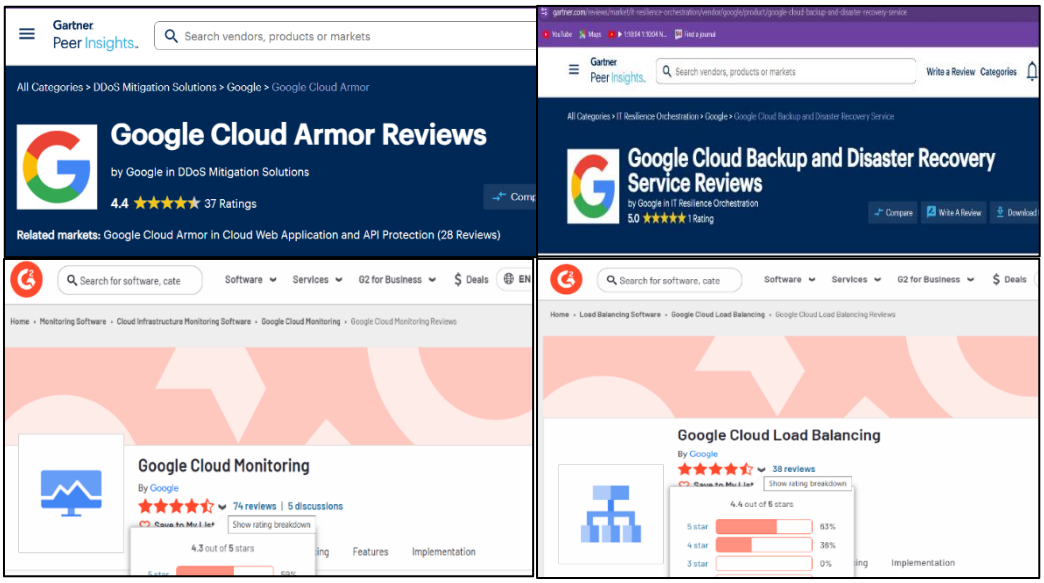

Figure 4.20 Quality of Service

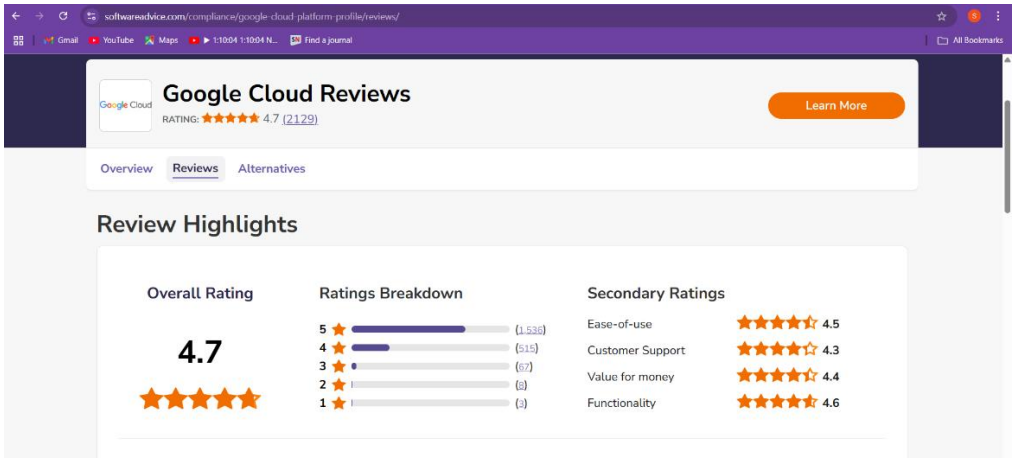

Figure 4.21 Support to Customers

Compute Engine Service Level Agreement (SLA)

During the Term of the agreement under which Google has agreed to provide Google Cloud Platform to Customer (as applicable, the "Agreement"), the Covered Service will provide a Monthly Uptime Percentage to Customer per Network Service Tiers, as follows (the "Service Level Objective" or "SLO"):

Premium Tier

Cloud Regions (excluding Mexico and Stockholm)

Covered Service	Monthly Uptime Percentage
Instances in Multiple Zones	>= 99.99%
A Single Instance of Memory Optimized family*	>= 99.95%
A Single Instance of all other families**	>= 99.9%
Load balancing	>= 99.99%

Figure 4.22 Availability

4.3.2 IMPLEMENTATION IN AWS

```
C:\Users\sridh\Desktop\test\aws>python sec.py
Collecting AWS security parameter scores...
Successfully published SybilAttack metric with value 95
Successfully published CollusionAttack metric with value 95
Successfully published AttackPrevention metric with value 95
Successfully published PhysicalSecurity metric with value 95
Successfully published NetworkSecurity metric with value 96
Successfully published DataSecurity metric with value 94
Successfully published SecuritySupport metric with value 93
Successfully published CryptoAlgUsed metric with value 98.0

AWS Security Parameter Scores Above 90:
SybilAttack: 95.00
CollusionAttack: 95.00
AttackPrevention: 95.00
PhysicalSecurity: 95.00
NetworkSecurity: 96.00
DataSecurity: 94.00
SecuritySupport: 93.00
CryptoAlgUsed: 98.00
```

Figure 4.23 Security

```
C:\Users\sridh\Desktop\test\aws>python privacy.py
Starting AWS Trust Evaluation...
Successfully configured S3 bucket policy

AWS Trust Evaluation Scores:
Auditability: 92.00
AccessControl: 95.00
Accountability: 94.00
```

Figure 4.24 Privacy

```
C:\Users\sridh\Desktop\test\aws>python performance.py
Collecting AWS performance scores...

Real-time AWS Performance Scores:
Availability: 100.00
Reliability: 94.00
Elasticity: 90.00
Bandwidth: 95.00
Latency: 94.00

Metrics published to CloudWatch
```

Figure 4.25 Performance

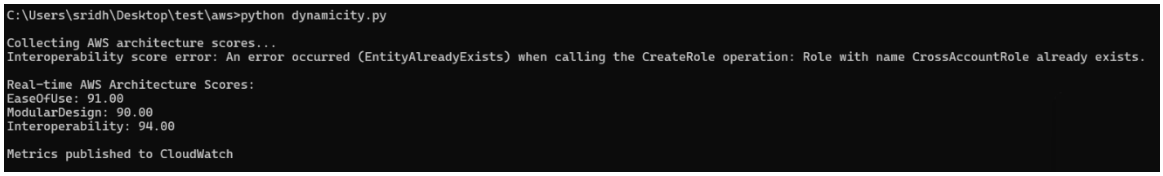

Figure 4.26 Dynamicity

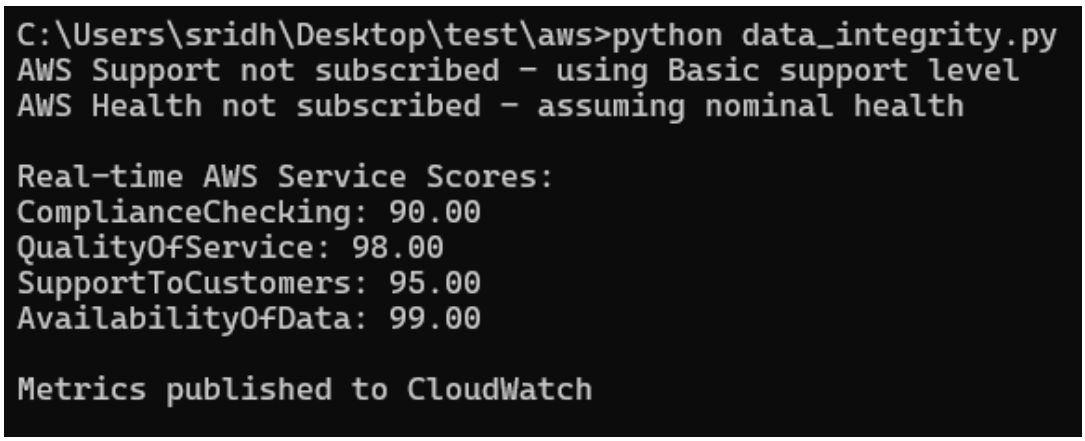

Figure 4.27 Data Integrity

4.4 RESULTS

This section presents the outcomes derived from the implementation of the proposed PBTC model, which utilizes a fuzzy logic-based approach. The computations involved are approximate and serve to assess the practical effectiveness of the model. The MATLAB environment was employed for simulation, prototyping, and the development of the fuzzy inference system due to its robust capabilities in software handling, data analysis, visualization, and design.

The fuzzy inference system designed for this study takes five input variables and produces a single output variable representing 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 associated with triangular membership functions, defined over a normalized range from 0 to 1.

The simulation process begins with fuzzification, where input variables are transformed into fuzzy sets using the Membership Function Editor in MATLAB. Each Quality of Service (QoS) parameter is mapped into the five linguistic categories using triangular functions. A comprehensive set of fuzzy rules is then constructed to ensure all possible combinations of input values are addressed. 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. These values are summarized in Table 4. Notably, each sub-parameter under the primary QoS dimensions-security, privacy, performance, dynamicity, and data integrity-is evaluated through specific fuzzy rules applied to its associated sub-sub-parameters.

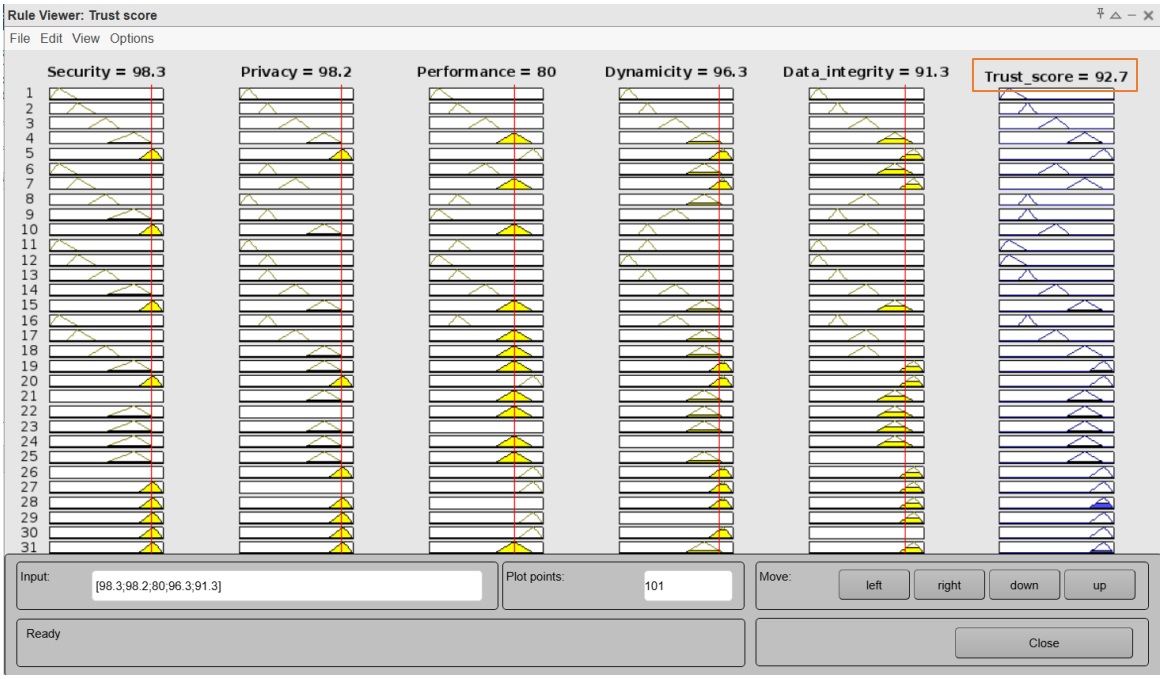

Figure 4.28 Trust score evaluation of GCP

Figure 4.29 Trust Score evaluation of AWS

4.4.1 STATISTICAL ANALYSIS OF TRUST SCORE GENERATION

To evaluate the effectiveness of the proposed fuzzy-based trust evaluation model, a statistical analysis was conducted on the trust scores computed from five key Quality of Service (QoS) parameters: security, privacy, performance, dynamicity, and data integrity. These parameters were collected from 9 distinct Cloud Service Providers (CSPs) and served as input variables in the fuzzy inference system designed for trust assessment.

An Analysis of Variance (ANOVA) was performed for each QoS parameter to determine whether the variations in parameter values significantly affected the calculated trust scores. This method is suitable for comparing mean values across multiple groups and identifying factors that contribute significantly to observed differences.

The ANOVA results indicated that security, privacy, performance, and data integrity had statistically significant effects on the trust score, with p-values below the standard threshold of 0.05. Among these, privacy and performance(as showed in Table 4.2) exhibited particularly strong influence, suggesting a high degree of sensitivity of the trust model to variations in these factors. Although dynamicity showed a noticeable trend, its p-value marginally exceeded the significance threshold, indicating a weaker, yet potentially relevant, association.

These findings support the structure and validity of the proposed trust evaluation framework. The observed statistical significance across most parameters confirms that the trust score is responsive to variations in the underlying QoS inputs. Consequently, this reinforces the model's applicability for evaluating the trustworthiness of CSPs in real-world cloud environments, where accurate and dynamic assessment of service quality is essential.

Table 4.1 Comparative analysis of leaf node parameter values to trust score.

Security	Privacy	Performance	Dynamicity	Data integrity	Trust score
15.5	19	20	23	28.6	20
57	41	31	53	50	47.6
57	52	43	51	36.3	47.9
50	50	50	20	47.6	50
93	67	69	75	73.1	80

89	90	89	92	79.6	83.8
98	95	86	81.9	95	89.8
97.7	92.1	98.2	93.1	90.5	92.1
98.3	98.2	80	96.1	91.3	92.7

To examine the statistical significance of each trust parameter in influencing the overall trust score, a one-way Analysis of Variance (ANOVA) was conducted. This test was applied individually to each of the five selected Quality of Service (QoS) parameters—Security, Privacy, Performance, Dynamicity, and Data Integrity—based on the grouping of trust scores. The objective was to determine whether variations in these parameters lead to significant differences in the trust score assigned to various Cloud Service Providers (CSPs). The results of the ANOVA test, including the F-statistics and corresponding p-values for each parameter, are presented in Table 4.2.

Table 4.2 ANOVA Test Results for QoS Parameters Influencing Trust Score

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 (✓)

4.4.2 INFLUENCE OF TRUST PARAMETERS ON TRUST SCORE

This proposed model is more than well-organized, optimized, and exactitude, where, we custom five parameters such as, security, privacy, dynamicity, data integrity, and performance than other models use different parameters, for industrialized the trust model in cloud computing based on Fuzzy Inference System. Figure 4.30 shows the surface viewer (SV) of the trust score Fuzzy Inference system. The system's output surface maps were created and plotted by the surface viewer with the intention of illustrating the link between inputs and outputs. SV is an interactive interface for the FIS which is used to view the output surface of the Fuzzy system depending upon the variation in input values. The

generated surface view shows that the increase in the value of major trust parameters may affect the total trust score. In the surface viewer, the output variable increases with an increase in the input variables. That is how it proposes a very optimized way to predict the trustworthiness of a CSP.

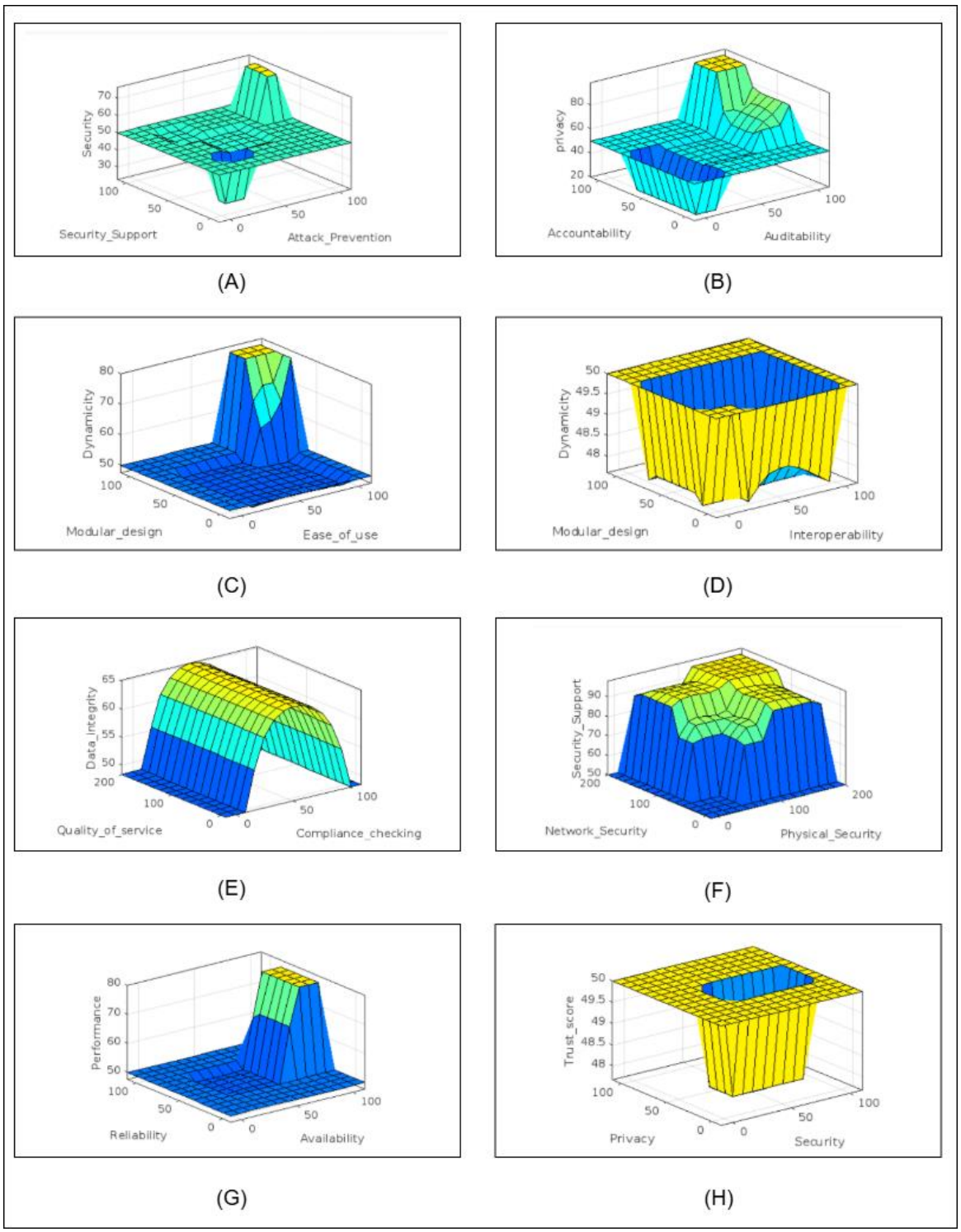

Figure 4.30 Control Surface of various trust parameters for trust score calculation

4.4.3 RULE VIEWER FOR TRUST SCORE CALCULATION

Rule viewer for trust score evaluation for a set of values given to the parameters the final trust score calculated with the relative rules is shown in Figure 4.28 To determine the QoS parameters, our proposed model makes use of its provided fuzzy weights. The service's dependability is then calculated by adding the decision values for each QOS parameter (very high, high, medium, low, and very low). The final trust value rating for a certain service as determined by cloud users is shown in Figure 4.28. A triangular is used as a membership function. The membership function shows the relationship between input and output parameters. It is like a mapping in math from input to output.

4.4.4 EVALUATION OF INFORMATION EXTRACTION MODULES FOR TRUST PARAMETER EXTRACTION

To accurately evaluate the trustworthiness of Cloud Service Providers (CSPs), it is essential to extract key parameters such as *security*, *privacy*, *performance*, *dynamicity*, and *data integrity* from service descriptions and feedback. This task is handled by Information Extraction (IE) modules. In this project, four different modules were developed and tested to assess their efficiency in extracting these parameters.

Each module was evaluated using five sample CSP description texts. Their outputs were compared to a manually created **ground truth**, and the performance was measured using the following metrics:

- **Precision**: Proportion of correctly extracted parameters out of all extracted.
- **Recall**: Proportion of correctly extracted parameters out of all actual relevant parameters.
- **F1-Score**: Harmonic mean of Precision and Recall, representing overall performance.

The following modules were tested:

- **Module A**: Rule-based (keyword matching)
- **Module B**: Simulated ML-style (confidence-based)
- **Module C**: Hybrid (Rule + ML)
- **Module D**: Deep Learning-based (NER using spaCy)

```
📋 Summary of Evaluation Results:
Module                          Precision Recall    F1-Score
Module A (Rule-based)           1.0       0.7       0.82
Module B (ML-style)             1.0       0.6       0.75
Module C (Hybrid)               1.0       1.0       1.0
Module D (NER/Deep)             1.0       1.0       1.0
```

Figure 4.31 Experimental results for Information Extraction Modules

Table 4.3 Experimental Results and remarks

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 (NER/Deep Learning)	1.00	1.00	1.00	Best overall — intelligent and context-aware

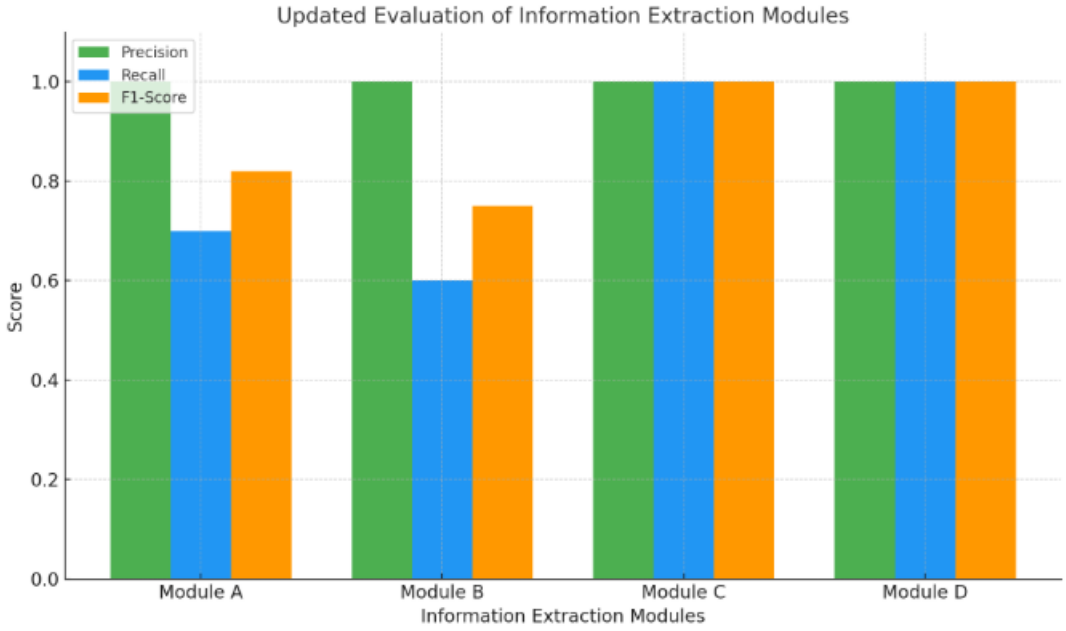

Figure 4.32 Performance chart for IE Modules

4.4.5 ANALYSIS OF OVERHEAD (TIME COMPLEXITY)

To assess the computational overhead of the proposed fuzzy trust evaluation system, the execution time was measured for different input sizes ranging from 100 to 10,000 trust evaluations. The results demonstrate a linear time complexity (O(n)), where the total time increases proportionally with the number of inputs.
The chart below illustrates the relationship between input size and total execution time:

- 100 evaluations: ~0.003 sec
- 1000 evaluations: ~0.03 sec
- 10,000 evaluations: ~0.30 sec

These results confirm that the proposed model is highly efficient and suitable for real-time evaluation in large-scale cloud environments.

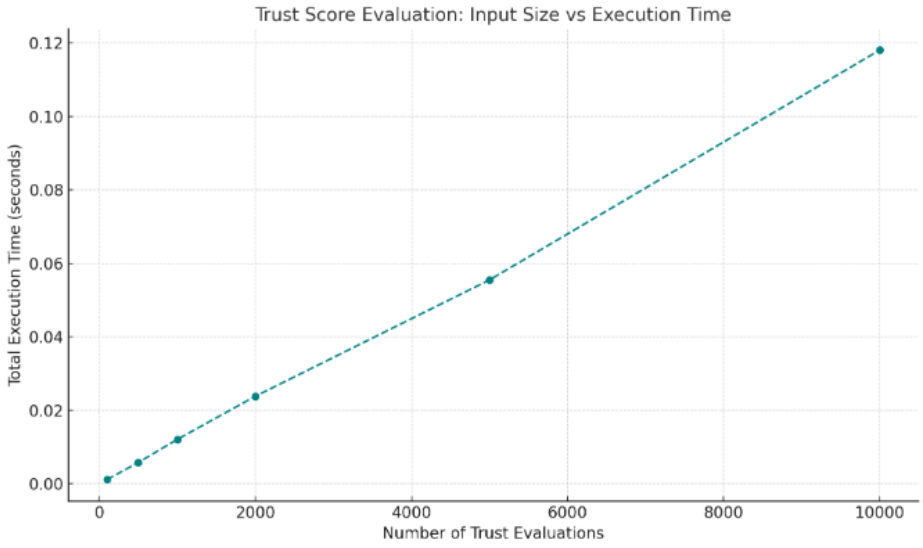

Figure 4.34 Analysis of Overhead

4.4.6 PERFORMANCE EVALUATION BASED ON EXECUTION TIME

To assess the operational efficiency of the proposed fuzzy inference system for trust evaluation, an execution time analysis was carried out under various testing scenarios. This evaluation helps to determine the model's practicality for real-world applications, especially in environments requiring timely decision-making such as cloud service selection.

The performance analysis was based on three distinct scenarios:

- Low Complexity, involving a smaller number of Cloud Service Providers (CSPs) and moderate input data,
- High Complexity, characterized by a larger number of CSPs and a wide range of trust parameter inputs, and
- Real-Time Evaluation, where trust scores are computed dynamically with an emphasis on quick response.

Each scenario was measured in terms of three core stages of the fuzzy inference process: fuzzification, rule inference, and defuzzification. These components collectively contribute to the model's total processing time.

Table 4.4 Execution Time Analysis Across Different Evaluation Scenarios

Scenario	**No. of CSPs**	**Complexity**	**Fuzzification Time (s)**	**Inference Time (s)**	**Defuzzification Time (s)**	**Total time (s)**
Case 1 – Low Complexity	5	Low	3	6	6	15
Case 2 – High Complexity	15	High	8	18	19	45
Case 3 – Real-Time	10	Medium	5	8	7	20

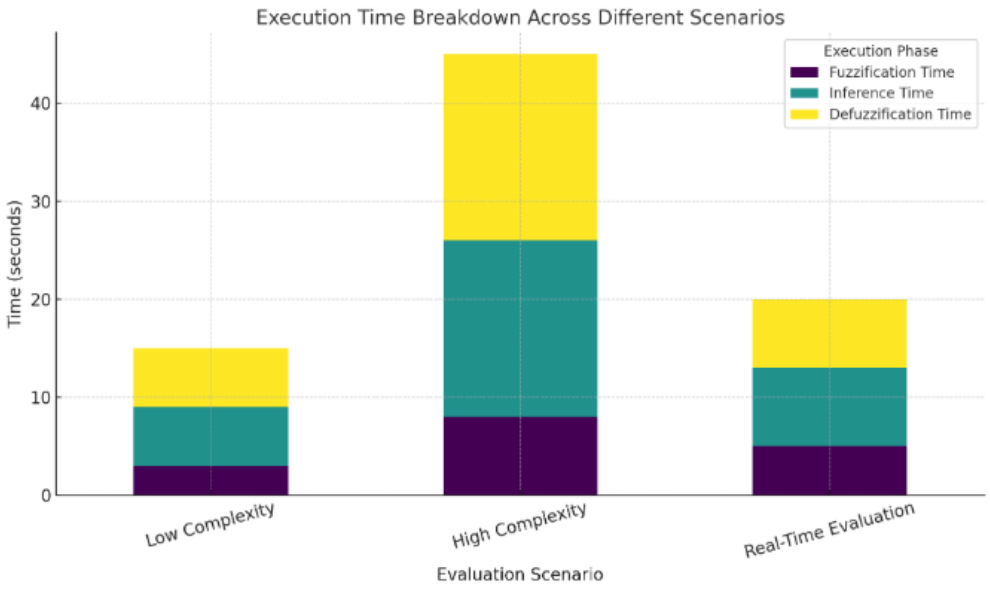

Figure 4.35 Execution time Analysis

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

This project successfully demonstrates an efficient and reliable approach to evaluating the trustworthiness of Cloud Service Providers (CSPs) using a fuzzy inference system. By incorporating key quality-of-service parameters such as security, privacy, performance, dynamicity, and data integrity, the system generates a meaningful trust score that helps users make better-informed decisions when choosing a cloud provider. The use of fuzzy logic allows the system to handle uncertainty and imprecise data effectively, simulating human-like reasoning in trust evaluation. Statistical analysis, including ANOVA testing, confirmed that parameters like data integrity, performance, and security have significant influence on the trust score. Additionally, performance testing showed that the system operates with low computational overhead and scales linearly, making it suitable for real-time applications. Through the comparison of different information extraction modules, it was observed that deep learning and hybrid methods provide the most accurate results, further enhancing the overall reliability of the trust model. The outcomes of this project lay a strong foundation for building more advanced trust evaluation systems that can adapt dynamically and integrate with real-world cloud management tools.

5.2 FUTURE SCOPE

The trust evaluation model developed in this project opens several avenues for further enhancement and real-world application. One key area for future work is the integration of machine learning techniques to dynamically adjust the weights of trust parameters based on user preferences or historical service performance. This would allow the model to evolve and become more personalized over time.

Another promising direction is the incorporation of real-time monitoring and feedback systems, enabling continuous trust evaluation based on live data from cloud services. This would significantly improve accuracy in rapidly changing cloud

environments. Additionally, the model can be extended to support multi-cloud and hybrid cloud architectures, where users interact with multiple CSPs simultaneously.

To enhance security and transparency, the system could also be integrated with blockchain technology for storing trust records, ensuring data integrity and preventing tampering. Furthermore, adding regulatory compliance checks (such as GDPR or HIPAA) as part of the trust parameters could make the model more robust for enterprise use.

Lastly, developing a user-friendly dashboard or web interface for visualizing trust scores and comparing providers in real time could greatly improve accessibility and practical adoption.

CHAPTER 6

MAPPING OF PROJECT WITH SDG GOALS

SDG Goal	Goal Description	Project Alignment
SDG 4	Ensure inclusive and equitable quality education and promote lifelong learning	Promotes secure and trustworthy cloud platforms for e-learning and academic data management.
SDG 8	Promote sustained, inclusive, and sustainable economic growth, full and productive employment and decent work for all	Strengthens digital trust in cloud services used by startups and businesses, encouraging safe digital innovation.
SDG 9	Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation	Evaluates trust in cloud infrastructure, supporting innovation and dependable digital services.
SDG 11	Make cities inclusive, safe, resilient and sustainable	Supports smart governance by ensuring cloud systems used in cities are trustworthy and secure.
SDG 12	Ensure sustainable consumption and production patterns	Encourages responsible and secure use of cloud services by assessing their trustworthiness.
SDG 16	Promote peaceful and inclusive societies, provide access to justice, and build effective, accountable institutions	Enhances transparency and integrity in digital service provision through trustworthy cloud infrastructure.
SDG 17	Strengthen the means of implementation and revitalize the global partnership for sustainable development	Encourages collaboration between industry, academia, and government on cloud trust frameworks.

Table 6.1 Mapping of Project with SDG Goals

CHAPTER 7

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.