University of Miskolc

FACULTY OF MECHANICAL ENGINEERING AND

INFORMATICS

Adaptive Fuzzy Logic Models for Efficient Cloud Service Management and SLA Optimization

PhD DISSERTATION

AUTHOR:

Ihab Razzaq Sekhi
MSc in Information Science

József Hatvany Doctoral School of

Information Science, Engineering and Technology

HEAD OF DOCTORAL SCHOOL

Prof. Dr. László Kovács

ACADEMIC SUPERVISORS

Prof. Dr. Szilvester Kovacs
Dr. Károly Nehéz

Miskolc
2025

Table of Contents II

1. Introduction

Cloud computing is a transformative technology enabling flexible, scalable access to computing resources, including applications, servers, storage, and networks, without initial investments. Known collectively as "XaaS," cloud services significantly enhance productivity, collaboration, accessibility, and security through internet-based delivery [1]. Service Level Agreements (SLAs) are essential in defining service expectations between cloud providers and users, ensuring accountability and trust through clear guarantees on availability, uptime, and downtime. SLAs are fundamental in enhancing Quality of Service (QoS) and building customer confidence in cloud environments [2][3]. Effective performance evaluation in cloud computing requires clearly defined objectives and metrics, such as Round-Trip Time (RTT) and response time, to address complex interactions within cloud infrastructure [4]. Cloud resource management involves brokers, SLA allocators, virtual machines (VMs), and physical machines (PMs). Brokers facilitate dynamic resource allocation, improving efficiency and quality of service by optimizing resource utilization and balancing workloads across geographically distributed data centers [5][6]. Fuzzy logic, a mathematical framework handling uncertainty through approximate reasoning, enhances decision-making in cloud computing. By simulating human cognitive processes, fuzzy logic provides flexible and precise classification, enabling effective handling of imprecise and ambiguous information [7][8]. This booklet introduces innovative fuzzy logic-based approaches for enhanced SLA management, VM allocation, and decision-making in cloud computing:

- **Estimating Cloud Computing RTT Using Fuzzy Logic:** Offers precise network latency evaluation for Amazon cloud environments using fuzzy logic categorization of inter-region distances.
- **Selecting SLA Guarantees Based on QoS Availability:** Develops intelligent fuzzy-based models categorizing SLAs into nine levels (90%–99.999%) to better align provider offerings with user-specific needs.
- **Optimized Fuzzy Logic for Decision-Making:** Introduces optimized fuzzy systems using flexible mathematical modeling to enhance precision, reduce costs, and improve scheduling and classification accuracy compared to traditional methods.
- **Efficient Broker-Driven Request Packet Size Approach:** Employs fuzzy logic for dynamic VM allocation based on request sizes, significantly enhancing system performance and reducing costs, demonstrated through simulations on Google Cloud [9].
- **Intelligent Validation Cloud Broker System (IVCBS):** Implements a dynamic allocation algorithm based on trapezoidal fuzzy membership functions, tested across global AWS centers, achieving optimized response time, reduced costs, and enhanced energy efficiency.
- **Reliable and Cost-Effective Fuzzy-based Cloud Broker:** Evaluates user-service compatibility, including mobile scenarios, through fuzzy logic to optimize cloud service selections using the Edge CloudSim simulator within Mobile Edge Computing (MEC) environments involving major providers like AWS, Google, and Azure [10].

2. Literature Review

Estimating Round-Trip Time (RTT) in cloud environments is challenging due to dynamic infrastructure factors. Geographical distances between globally distributed data centers significantly elevate latency, impacting real-time applications and necessitating optimized routing strategies [11]. Additionally, network congestion in shared, multi-tenant environments complicates RTT estimation by causing packet reordering, unnecessary retransmissions, and reduced TCP performance, emphasizing the importance of effective congestion management for stable network operations. RTT remains a vital metric for assessing network performance and Quality of Service (QoS) across diverse systems, including IoT and cellular networks, where it supports delay diagnosis and congestion control optimization [12]. Furthermore, RTT-based assessments improve reliability and reduce environmental impacts, particularly within IoT settings [13]. Service Level Agreements (SLAs) are crucial for managing cloud computing resources effectively. Patel et al. [14] proposed a WSLA-based architecture for automating cloud SLA management, incorporating third-party security enhancements. Alhamad et al. [15] outlined key SLA criteria across different cloud models (IaaS, PaaS, SaaS), emphasizing performance factors such as boot time and response times. Qiu et al. [16] analyzed multiple SLAs, identifying gaps in customer data security, privacy, and insufficient clarity regarding availability commitments and penalties. Baset [17] further dissected SLAs from several providers to clarify availability obligations, while Godhrawala and Sridaran [18] leveraged machine learning techniques to enhance QoS management. Akbari-Moghanjoughi et al. [19] advocated comprehensive methodologies, and Saqib et al. [20] proposed adaptive machine learning-based traffic classification to sustain SLA compliance. Fuzzy logic has become instrumental for decision-making under uncertainty, offering flexibility and approximate reasoning. However, challenges persist, including the complexity of fuzzy rule formulation and computational inefficiencies. Building on foundational concepts introduced by Zadeh [21], recent studies integrated fuzzy logic with machine learning, aiming to improve diagnostic accuracy despite inherent subjective biases and complexity [22]. Adaptive fuzzy systems often experience stability issues, leading to inconsistent decision-making [23]. Traditional approaches like the Mamdani fuzzy inference model, although foundational, exhibit limited robustness under varying conditions. Hybrid methods combining fuzzy logic and genetic algorithms also face convergence and efficiency problems [24]. Consequently, there is an urgent need for optimized methodologies to enhance fuzzy logic's practical applicability and robustness [25]. Cloud computing's rapid adoption highlights significant challenges in performance evaluation and security management [26]. Recent studies explored various aspects, including SLA methodologies [2], roles and functions of cloud brokers, and algorithms like COTD for cost-effective service delivery. Researchers have emphasized security and compliance challenges in multi-cloud environments, advocating advanced encryption and identity management strategies [27]. Interoperability among different cloud services remains an ongoing challenge, necessitating frameworks for federated cloud integration and agent-based resource management. Studies employing Cloud Analyst simulations examined load balancing algorithms and service broker policies to optimize QoS [28]. Advanced VM allocation strategies have increasingly focused on optimizing resource utilization and performance, addressing overlooked issues such as variable packet sizes. Innovations include integrating neural-fuzzy systems with Ant Colony Optimization (ACO) techniques [29], broker-driven VM allocation strategies [30], fuzzy logic controllers, and reinforcement learning schedulers. Machine learning-based VM migration, intelligent multi-agent systems , AI-driven resource management methods, and hybrid heuristic algorithms further enhance resource efficiency. These developments represent significant advancements, outlining promising future directions for resource management in cloud computing [31]. Cloud brokerage services have attracted extensive research attention, primarily focusing on QoS enhancement, broker profitability, or balancing provider and customer interests. Approaches such as game theory [32], reinforcement learning, weighted algorithms, ontology, AHP combined with TOPSIS, and fuzzy logic each offer distinct advantages but also face significant challenges, including prolonged negotiation, complex user interactions, and extended learning periods [33]. Addressing these limitations, this study integrates fuzzy logic and modified

TOPSIS to efficiently match users with appropriate service instances from major CSPs, simplifying complexity and enhancing the balance between user and provider interests.

3. Cloud Computing

Cloud computing has evolved into a central paradigm in modern computing, reshaping how individuals and organizations access and manage technological resources. Chapter 2 offers a comprehensive examination of cloud computing, detailing service models, deployment strategies, and core characteristics, providing a valuable lens through which to analyze its increasing relevance and transformative impact [34]. At the heart of cloud computing are its service models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)—each providing varying levels of abstraction and management responsibility between cloud service providers (CSPs) and consumers [35]. IaaS offers virtualized hardware resources such as compute instances, storage, and networking, enabling users to deploy and manage their operating systems and applications while avoiding the complexity of physical infrastructure maintenance. Prominent examples of IaaS include AWS EC2, Google Cloud Platform (GCP), and Microsoft Azure Virtual Machines, offering benefits such as elasticity, scalability, and pay-per-use cost structures [36]. PaaS abstracts away infrastructure management, offering platforms where developers can deploy, test, and manage applications without dealing with server or storage configurations. PaaS includes tools for coding, database management, middleware, and application hosting, streamlining the development process and accelerating time-to-market. Examples include Google App Engine, Heroku, and Azure App Service [37]. SaaS, in contrast, delivers complete applications over the internet, with CSPs managing all underlying infrastructure, updates, and maintenance. SaaS products—such as Dropbox, Slack, Zoom, and Google Workspace—offer users simplified access to powerful software solutions through subscription-based pricing, reducing costs and operational burdens [34]. Beyond service models, deployment strategies shape how cloud services are architected and delivered. The public cloud model provides services from third-party vendors, shared among multiple tenants, offering cost efficiency and high scalability. However, it raises concerns around data privacy and security, requiring strong access controls and encryption mechanisms. In contrast, private clouds are dedicated to single organizations, offering greater control, security, and compliance but typically requiring higher investments and operational expertise [38]. Hybrid clouds combine public and private environments, allowing organizations to keep sensitive workloads in private infrastructure while leveraging public clouds for scalability and cost savings. This integration demands orchestration tools and APIs to ensure seamless operations [39]. Community clouds, meanwhile, serve specific groups of organizations sharing similar objectives, such as regulatory compliance or industry-specific requirements. They blend benefits of private clouds with cost savings achieved through shared resources, offering a middle ground between exclusivity and affordability [40]. Central to understanding cloud computing are its defining characteristics as outlined by the National Institute of Standards and Technology (NIST): on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [39]. These properties enable users to dynamically provision computing resources, access services from diverse devices, and benefit from granular billing based on actual consumption, ensuring cost efficiency and operational agility. Overall, Chapter 2 highlights how cloud computing delivers a flexible and scalable architecture for modern IT services, empowering organizations to innovate while optimizing costs and resources. The layered service models and deployment options allow organizations to select solutions that balance security, compliance, performance, and financial considerations, positioning cloud computing as a pivotal enabler of digital transformation.

4. Adoption and Implementation of Cloud Platforms

Chapter 3 provides a detailed exploration of the drivers, benefits, and complexities involved in adopting and implementing cloud platforms. Organizations are increasingly moving to cloud solutions driven by the need for agility, scalability, cost optimization, and technological innovation [41]. Major Cloud Service Providers (CSPs) like AWS, Google Cloud, and Azure offer structured adoption frameworks, including tools, templates, and readiness assessments to guide enterprises through migration and modernization efforts [42]. Key adoption drivers include enhancing business agility, enabling rapid deployment and scalability to support dynamic market conditions. Cloud services allow global infrastructure deployment within minutes, significantly shortening innovation cycles compared to traditional IT setups [41]. Business adaptability is further supported by flexible and scalable cloud resources, enabling organizations to adjust quickly to changing demands and leverage emerging technologies such as AI and analytics for strategic decision-making [43]. Ensuring business continuity is a vital consideration in cloud adoption. Cloud platforms incorporate redundancy and disaster recovery measures to maintain operations during disruptions. Local redundancy within a single data center protects against localized issues, while geographical redundancy disperses data across distant sites to safeguard against regional failures—features critical for disaster recovery and maintaining high availability. CSPs typically guarantee high uptime, such as 99.99% (≈52 minutes of downtime annually), though certain industries, like healthcare, demand stricter standards approaching 99.999% availability. Cloud providers also ensure exceptional data durability through extensive replication, achieving up to eleven nines (99.99999999%) durability, minimizing data loss risks and preserving organizational trust [44]. Security is another fundamental pillar of cloud adoption, addressed through a shared responsibility model where CSPs secure infrastructure while customers manage access and data protection. Advanced encryption, dedicated security teams, and multi-level protections ensure robust defense against vulnerabilities, supporting compliance with regulatory standards [45]. Economically, cloud adoption transitions costs from significant upfront capital expenditures to flexible operational expenses through pay-as-you-go models, potentially reducing IT spending by over 50%. CSPs leverage economies of scale to lower service costs, enabling businesses to avoid overprovisioning and achieve more efficient resource use [46]. Virtualization underpins the scalability and efficiency of cloud platforms, allowing multiple Virtual Machines (VMs) to run on a single physical server, optimizing hardware utilization and reducing costs. Hypervisors, either Type 1 (bare metal) or Type 2 (hosted), facilitate VM creation and resource allocation, supporting efficient multi-tenant operations. Type 1 hypervisors, such as VMware ESXi and Hyper-V, are preferred for enterprise use due to their performance advantages, while Type 2 hypervisors like VirtualBox are suitable for smaller deployments [47]. Networking architecture in cloud environments is crucial for connecting distributed resources and ensuring performance. Data Center Networks (DCNs) use hierarchical architectures, incorporating access, aggregation, and core layers to manage internal traffic and connect to external networks. Meanwhile, Data Center Interconnect Networks (DCINs) bridge geographically separated data centers, supporting seamless service delivery, disaster recovery, and workload migration [48]. Innovations in optical networking enhance throughput but add management complexity. CSPs form a competitive ecosystem offering diverse services, including IaaS, PaaS, and SaaS [49]. AWS, Google Cloud, and Azure dominate the market, each providing extensive service portfolios tailored to various business needs, from compute resources to AI tools [50]. SLAs formalize service expectations, covering performance, availability, and security, ensuring accountability and legal protection for cloud consumers [51]. Overall, Chapter 3 underscores that successful cloud adoption requires strategic planning, technical readiness, and careful consideration of operational, security, and financial factors. Cloud platforms offer transformative potential, but organizations must align technology adoption with business objectives and risk management to fully capitalize on cloud benefits [52].

5. RTT Estimation and Optimization in Cloud Computing

This study focuses on the estimation and optimization of Round-Trip Time (RTT) in cloud computing, addressing critical challenges posed by the dynamic and geographically distributed nature of cloud infrastructures. RTT, a fundamental metric representing the time for a signal to travel from a source to a destination and back, directly influences latency and impacts the Quality of Service (QoS) for time-sensitive applications such as telemedicine, online gaming, and robot-assisted surgery. Traditional cloud infrastructures often struggle to meet stringent latency requirements for such applications, necessitating innovative approaches to RTT estimation and management [53]. Geographical distance significantly impacts RTT, as data transmission delays increase with physical separation between cloud data centers and end-users [11]. Private network backbones and direct peering have been shown to mitigate such latencies, improving performance across regions [54]. For example, analysis of the Tahoe Least-Authority File System (Tahoe-LAFS) highlighted variations in performance between community network clouds and commercial platforms like Microsoft Azure, driven largely by differences in network homogeneity and routing efficiencies [55]. Studies further reveal that queue-based dynamic resource allocation outperforms spatial resource partitioning in reducing latency and improving performance, indicating the importance of intelligent resource management for low-latency operations [56]. Network congestion compounds RTT estimation challenges in multi-tenant environments. Congestion triggers packet reordering, causing Transmission Control Protocol (TCP) to interpret these events as packet loss, leading to spurious retransmissions and throttling of transmission rates [12]. The development of systems such as Bolt has emerged as a response to these issues, leveraging techniques like Sub-RTT Control (SRC), Proactive Ramp-Up (PRU), and Supply Matching (SM) to reduce latency and maximize network utilization, even at high line rates up to 400Gbps [57]. Addressing RTT estimation, this chapter introduces a fuzzy logic-based model integrating triangular membership functions to analyze inputs such as geographical distance and network congestion. By categorizing these variables into linguistic terms—e.g., small, medium, and long distances, and Light, Average, and Peak congestion—the model produces nuanced RTT predictions across nine output categories. Such an approach overcomes the binary limitations of traditional methods, offering flexibility and precision in cloud performance evaluation [58]. Empirical testing on AWS infrastructure demonstrated the superiority of fuzzy logic based RTT estimates compared to standard online measurement tools, yielding lower and more accurate RTT values [59]. Moreover, the fuzzy logic framework facilitates intelligent system behavior analysis and decision-making under uncertainty. Unlike conventional binary systems, fuzzy logic captures partial truth values, enabling adaptive responses to complex network conditions and supporting tasks like delay prediction and cloud resource allocation. This capability is vital for managing uncertainty in dynamic cloud environments and maintaining SLA compliance. Beyond RTT estimation, intelligent systems incorporating fuzzy logic are instrumental in optimizing cloud operations, enhancing reliability, and ensuring efficient workload management [60]. Studies highlight that fuzzy reasoning models excel in processing imprecise data, improving adaptability, and supporting predictive analytics, all crucial for real-time, scalable cloud services [61]. In summary, Chapter 4 underscores the critical role of advanced methodologies like fuzzy logic in RTT estimation and optimization within cloud computing. By effectively modeling variables such as geographical distance and congestion, the proposed approach advances cloud performance analysis and SLA management, ensuring improved user experiences and reliable service delivery in increasingly demanding digital ecosystems.

5.1 Experimental Methodology for RTT Measurement and Analysis Using Fuzzy Logic

Accurately measuring Round-Trip Time (RTT) is critical for assessing network performance, especially in cloud computing contexts where latency significantly affects Quality of Service (QoS) and application responsiveness. Chapter 4 presents an experimental methodology for RTT analysis, combining traditional measurement techniques and advanced modeling approaches to enhance precision in cloud environments like AWS. Among the primary techniques utilized is the Ping Test, a widely accepted method for quickly evaluating network latency by sending packets from a user's device to a remote server and measuring the response time. The RTT captured through ping reflects latency influenced by factors such as network congestion and the physical distance between nodes. A stable network is indicated by a consistent, horizontal trend in ping test results, while fluctuating RTT values can signal network congestion or instability. In this study, ping testing was crucial for verifying connectivity between senders and AWS endpoints, offering practical insights into real-world network performance [62]. Complementing ping tests, mathematical modeling techniques are employed to derive RTT more precisely. These models capture network transactions—including client-server interactions via TCP and UDP—and analyze the timing of read and write operations. Precision appliances such as the Exinda device intercept and timestamp network packets with nanosecond accuracy, enabling fine-grained RTT calculation [63]. This approach segments RTT into server-side and client-side components, enhancing measurement granularity. As data accumulates, these models continuously refine RTT estimates by averaging observations, thereby improving accuracy over time [64]. Beyond measurement techniques, the study integrates geospatial analysis to understand how geographical distance influences RTT. Data was systematically extracted for 28 AWS regions worldwide, treating these as endpoints for RTT evaluation. Utilizing the AWS latency testing platform, researchers measured network latency from the sender's location in Kut, Muhafazat Wasit, Iraq. Geographic coordinates were determined for all endpoints, and the Haversine formula was applied to calculate great-circle distances between the sender and AWS data centers. This geospatial approach quantifies how physical distance contributes to RTT, providing a precise assessment of latency across different continents and enabling detailed connectivity analyses [2]. Collectively, the integration of empirical ping tests, high-resolution network monitoring devices, and geospatial analysis provides a robust experimental framework for evaluating RTT in cloud environments. This methodology offers critical insights into latency patterns, enabling more accurate cloud service evaluations and enhancing resource allocation strategies for improving QoS.

5.1.1 Fuzzy Logic Framework

The proposed model employs several triangular membership functions. [65], formulated in Equation (5.1), to convert crisp values into fuzzy sets. The MATLAB Fuzzy Logic Designer tool was utilized to develop the model, as depicted in Figure 5.1, the model integrates two input parameters, distance, and network congestion. The model utilizes three triangular membership functions for each input parameter.

$$Triangular\ membership\ function\ (d: l, m, n) \quad = \quad \begin{cases} 0, & d < l \\ \frac{d-l}{m-l}, & l \le d \le m \\ \frac{n-d}{n-m}, & m < d \le n \\ 0, & d > n \end{cases} \tag{5.1}$$

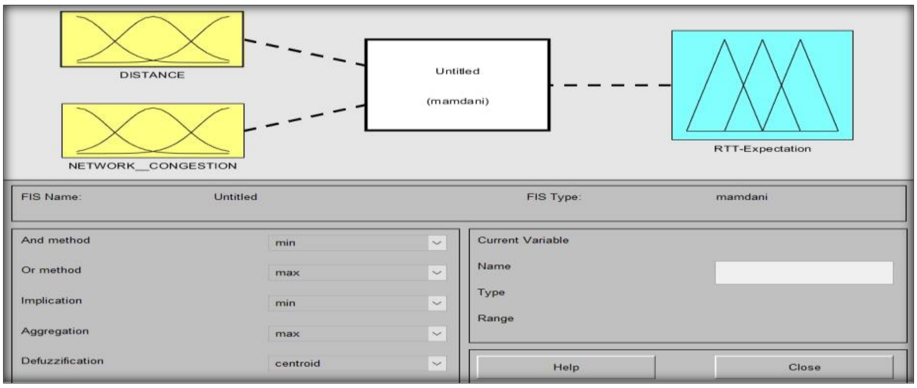

Figure 5.1. Proposed model design.

i. Input Variables Definition

- Distance:

  Small: [0, 862.94, 4516]; Medium: [2689, 8170, 11824]; Long: [9997, 15478, 15478.65]

Network Congestion was similarly divided into three fuzzy sets. Light, Average, and Peak reflecting latency characteristics were measured at different times of the day and under varying network load conditions.

- Network Congestion:

  Light: [0, 3, 6]; Average: [3, 6, 8]; Peak: [7, 14, 23.59].

ii. Output Variables Definition

The expected (RTT-Expectation) output is defined as follows:
RTT1: [0, 0, 25]; RTT2: [10, 50, 75]; RTT3: [50, 100, 125]; RTT4: [100, 150, 175]; RTT5: [150, 175, 200]; RTT6: [175, 200, 250]; RTT7: [200, 250, 325]; RTT8: [250, 325, 350]; RTT9: [325, 430, 500].

The output variable, Expected RTT, Table 5.1, was defined using nine triangular membership functions labeled RTT1 through RTT9, each corresponding to specific ranges of RTT delays identified in our measurements.

Table 5.1 Expected RTT.

Distance / Network congestion	Small	Medium	Long
	RTT Expectation		
Light	RTT1	RTT4	RTT7
Average	RTT2	RTT5	RTT8
Peak	RTT3	RTT6	RTT9

5.1.2 Fuzzy Logic-Based RTT Estimation Model

The study introduces a fuzzy logic-based model for estimating Round-Trip Time (RTT) in cloud networks, designed to manage uncertainties inherent in network latency. The model comprises four key components: fuzzification, inference engine, knowledge base, and defuzzification. Fuzzification converts crisp numerical inputs, like geographical distance (km) and network congestion (ms), into fuzzy sets using triangular membership functions, enabling nuanced handling of variability in network conditions. The inference engine applies nine “if-then" rules stored in the knowledge base, processing these fuzzy inputs to produce fuzzy outputs. These outputs are then converted back into precise numerical RTT estimates through the centroid defuzzification method (COG), ensuring realistic, weighted predictions aligned with actual network behaviors. This methodology delivers superior adaptability and precision over traditional deterministic techniques, effectively capturing the impacts of distance and congestion on RTT. Empirical testing demonstrates that RTT remains low at short distances and light congestion (e.g., RTT1 ≈ 25 ms). However, as distance or congestion increases, RTT rises progressively, reaching values like RTT9 ≈ 500 ms under long-distance and peak congestion conditions. The proposed fuzzy model provides critical benefits: it enables informed decisions regarding data center selection to minimize latency, anticipates congestion for proactive resource management, and supports adaptive routing and bandwidth allocation for optimized QoS and SLA compliance. Additionally, the model can act as an early warning system, detecting rising RTT as an indicator of potential network strain, prompting preventive actions such as rerouting traffic or upgrading infrastructure. This fuzzy logic approach ensures precise, real-time RTT predictions, empowering network operators to maintain service reliability and performance across dynamic cloud environments.

5.1.3 Evaluation and Analysis of the Proposed Model for RTT Estimation

The proposed RTT estimation model was validated through simulations reflecting real-world conditions between Kut, Iraq, and 28 AWS regions. Using ping tests and precise distance calculations, the model captured RTT variations across different congestion levels. Results showed low RTT during off-peak hours and gradual increases during business hours and peak usage. Unlike static RTT values from providers like AWS, the model provides dynamic, detailed estimations. RTT1–RTT3 represent optimal performance, while RTT9 signals severe degradation. This dynamic approach empowers users to choose regions with the best latency, enhancing network performance and service quality Table 5.2.

Table 5.2 Comparison of the Proposed Model Results with AWS Round-Trip Time (RTT) Measurements.

NO	Computed Distance Between the Sender and Receiver(km)	Amazon (RTT) (ms) During Daytime	Estimated Latency Values in the Proposed RTT Classifications During Daytime Hours(ms)		
			Light congestion	Average congestion	Peak congestion
1	862.94	62	9	45	92
2	1234.23	50	9	45	92
3	3089.72	361	30	65	110
4	3428.79	88	50	86	128
5	3525.01	100	57	92	134
6	3601.23	102	62	97	138
7	3607.54	113	62	97	139
8	4009.87	115	93	127	166
9	4202.65	112	110	144	181
10	4238.49	127	113	147	184
11	4682.33	138	142	175	208
12	5981.25	388	142	175	208
13	6012.87	212	142	175	208
14	6789.34	347	142	175	208

15	7056.22	339	142	175	208
16	7289.64	369	142	175	208
17	7435.78	414	142	175	208
18	7832.90	426	142	142	208
19	8053.21	374	142	142	208
20	8923.45	181	142	142	208
21	10023.67	172	143	143	210
22	10289.47	198	155	155	232
23	12345.89	279	258	258	418
24	12678.56	242	258	258	418
25	13756.90	390	258	258	418
26	14321.76	427	258	258	418
27	14989.34	266	258	258	418
28	15478.65	300	258	258	418

5.2 Summary of an Innovative Fuzzy Logic-Based Model for RTT Assessment in AWS Cloud Services and SLA Optimization

This research presents a fuzzy logic-based model for estimating RTT in AWS cloud services, integrating factors like geographical distance and network congestion for precise, adaptive predictions. Unlike static AWS tools, this model offers dynamic RTT assessments, empowering users to choose optimal cloud regions for SLA compliance and QoS, especially in latency-sensitive applications. Triangular membership functions categorize RTT into performance levels, capturing variability across network conditions. Comparative analysis shows superior accuracy over AWS's static reporting. The approach enhances cloud service selection, network monitoring, and resource allocation, supporting reliable, real-time cloud operations and future research into broader performance metrics.

6. Quality of Service (QoS) Availability Assessment for Optimal SLA Selection

Cloud computing has fundamentally transformed IT infrastructure, enabling real-time, on-demand access to computing resources like applications, servers, and networks without significant upfront investment. It provides scalability and flexibility through service models such as SaaS, PaaS, and IaaS, earning user trust for its cost efficiency and reliability [66]. However, widespread adoption has brought challenges, notably in ensuring data privacy, system security, and transparent guarantees of service performance defined in Service Level Agreements (SLAs) [67]. QoS metrics—including throughput, Round Trip Time (RTT), jitter, and packet loss—are crucial indicators of service availability, yet often remain obscured in complex SLA documents, complicating users' ability to make informed decisions [68]. Establishing clear, measurable guarantees in cloud SLAs requires thorough assessment and transparent communication between providers and customers. This necessity aligns with the shared responsibility model, which delineates duties between cloud providers and users for securing and managing resources across IaaS, PaaS, and SaaS environments [69]. Traditional SLA selection mechanisms typically focus on formal, quantifiable service attributes but overlook users' subjective preferences and qualitative considerations [70]. Many platforms restrict customers to pre-configured service packages, failing to clarify the guarantees behind each package. This gap has prompted research into methodologies that integrate users' subjective opinions into SLA decisions, capturing factors like personal performance expectations or specific operational needs. In response, a fuzzy logic-based model has emerged to classify SLAs into nine availability levels ranging from 90% to 99.999%, reflecting diverse user requirements and budgets. This model integrates both computing metrics (vCPU, RAM, storage) and networking metrics (BW, jitter, RTT, packet loss), enabling nuanced SLA evaluations tailored to individual applications—from basic office use to high-demand scenarios like gaming or scientific computing [71]. Existing literature has significantly explored SLA methodologies. Patel et al. [14] proposed a WSLA-based architecture with trusted third-party components to automate SLA management and enhance security. Alhamad et al. [15] outlined critical SLA design factors across IaaS, PaaS, and SaaS, emphasizing attributes like boot times and response times. Qiu et al. [16] analyzed 29 SLAs, identifying widespread gaps in customer protections, notably around data privacy, backup policies, and concrete penalty structures, despite general guarantees of availability. They underscored the need for clearer, enforceable SLAs to foster trust. Baset [17] provided a framework for dissecting SLAs into components for clearer provider comparisons, focusing on compute and storage services in IaaS and PaaS contexts. Similarly, Godhrawala and Sridaran [18] advanced SLA methodologies by integrating ML-based A priori algorithms into service-oriented architectures, connecting QoS metrics and improving resource optimization. Akbari-Moghanjoughi et al. [19] conducted comprehensive reviews on SLA deployment, emphasizing the necessity of linking Service Level Objectives (SLOs) to specific service domains for more meaningful guarantees. Meanwhile, Saqib et al. [20] proposed adaptive machine learning methods to classify network traffic dynamically, aiming to reduce SLA violations and optimize resource allocation, ultimately enhancing SLA integrity and cost-efficiency. Together, these studies and the newly proposed fuzzy logic model mark significant progress toward more transparent, user-centered SLA selection in cloud computing. They highlight the importance of both quantitative metrics and subjective user inputs in aligning service offerings with practical operational demands, ensuring reliability and user satisfaction in modern digital infrastructures.

6.1 QoS Availability and SLA Assessment Framework in Cloud Computing

The assessment of Quality of Service (QoS) availability is crucial for effective Service Level Agreement (SLA) management in cloud computing environments. This chapter introduces structured methodologies for calculating both computing and networking availability metrics, forming the basis for a fuzzy logic-driven SLA selection framework designed to align cloud services with user requirements and expectations.

6.1.1 Calculation of QoS Computing Availability Metrics

QoS computing availability is computed by aggregating the individual availability percentages for vCPU, RAM, and Storage using a weighted average, as follows:

$$A_{Computing}=(W_{vCPU}\times A_{vCPU})+(W_{RAM}\times A_{RAM})+(W_{Storage}\times A_{Storage}) \tag{6.1}$$

- A_{vCPU}, A_{RAM}, $A_{Storage}$ is represent the individual availability percentages.
- W_{vCPU}, WRAM, $W_{Storage}$ is represent the relative weights assigned to these metrics.

If explicit weights are not provided, equal weighting (1/3 each) is assumed, thus simplifying the formula to:

$$A_{Computing}=A_{vCPU}+A_{RAM}+A_{Storage}/3 \tag{6.2}$$

6.1.2 Calculation of QoS Networking Availability Metrics

Similarly, QoS networking availability aggregates four network metrics: BW, Round Trip Time (RTT), Jitter, and Packet Loss. The weighted average aggregation formula is:

$A_{Networking}=(W_{BW}\times A_{BW})+(W_{RTT}\times A_{RTT})+(W_{Jitter}\times A_{Jitter})+(W_{PacketLoss}\times A_{PacketLoss})$ (6.3)

- A_{BW}, A_{RTT}, A_{Jitter}, $A_{PacketLoss}$ is represent individual network metric availabilities.
- W_{BW}, W_{RTT}, W_{Jitter}, $W_{PacketLoss}$ is represent metric weights.

If explicit weights are not provided, equal weighting (1/4 each) simplifies this equation to:

$$A_{Networking}=A_{BW}+A_{RTT}+A_{Jitter}+A_{PacketLoss}/4 \quad (6.4)$$

These equations provide a structured, transparent, and reproducible approach to calculating the fuzzy inputs clearly from the individual QoS metrics.

6.2 Proposed SLA Assessment Framework

Building on these metrics, a fuzzy logic-based framework is proposed for optimal SLA selection (see Figure 6.1). This system processes QoS availability metrics as input variables, integrating both technical performance indicators and user preferences into a unified decision model. The framework categorizes SLAs into tiers:

- Normal SLA (90–92%)
- Bronze SLA (93–95%)
- Silver SLA (96–97%)
- Gold SLA (98–99.999%)

This tiered approach ensures a structured differentiation of service levels, enabling users to select SLAs tailored to their operational needs and budget constraints. Both computing and networking availability metrics are evaluated within a defined universe of discourse spanning 90% to 100%. To achieve granularity, a non-linear sequence of availability values is generated using the formula:

$$A_n = 90+(n-1)\times(0.09999-(n-1)\times 0.00001) \quad (6.5)$$

for n ranging from 1 to 101, progressively converging toward 99.999% availability. This mathematical precision ensures a nuanced classification of QoS levels for SLA optimization.

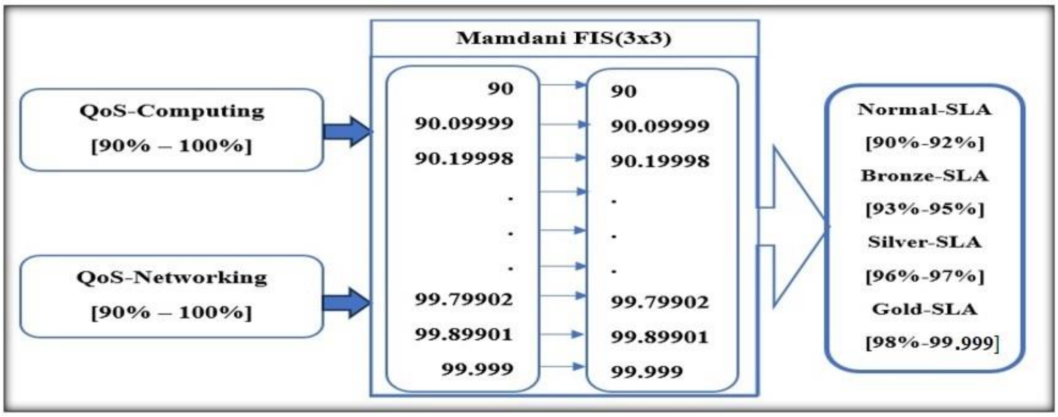

Figure 6.1 Proposed SLA guarantee model.

6.2.1 Fuzzy Logic-Based Methodology for QoS Evaluation and SLA Classification

The fuzzy logic-based methodology presented for Quality of Service (QoS) evaluation and SLA classification introduces a sophisticated framework that blends mathematical precision with domain expertise, enabling nuanced decision-making in cloud service management.

6.2.1.1 Key Input Parameters and Fuzzification

At the core of the proposed system is fuzzification, which translates crisp numerical inputs into fuzzy linguistic variables to handle uncertainty inherent in network performance assessments . The model utilizes two key input variables:

- **QoS-Computing Availability** — covering vCPU, RAM, and storage availability, defined over a universe of discourse from 90% to 100% and modeled using triangular membership functions:
  - Light Availability: [90, 90, 95]
  - Middle Availability: [90, 95, 100]
  - High Availability: [95, 99.999, 100]
- **QoS-Networking Availability** — representing metrics like bandwidth, RTT, jitter, and packet loss, also defined over 90% to 100% with similar triangular membership functions:
  - Low Availability: [90, 90, 95]
  - Average Availability: [90, 95, 100]
  - Top Availability: [95, 99.999, 100]

This dual-input structure ensures the system captures both computational and networking performance factors for SLA assessment.

6.2.1.2 Fuzzy Inference System and Defuzzification

The model implements a Mamdani-type Fuzzy Inference System (FIS), utilizing a rule base that comprehensively links the two inputs through 3×3 = 9 inference rules in Table 6.1. The rules are structured in a matrix to determine SLA classification based on combinations of computing and networking availability. For example, high computing and top networking availability map to the Gold SLA tier, while lower combinations fall into Bronze or Normal SLAs.

Table 6.1 Fuzzy rule base.

QoS- Computing / QoS-Network	Light	Middle	High
	(SLA) Guarantees		
Low	Normal-SLA1	Bronze-SLA1	Silver-SLA1
Average	Normal-SLA2	Bronze-SLA2	Silver-SLA2
Top	Normal-SLA3	Bronze-SLA3	Gold-SLA

6.2.1.4 Validation Process

Validation was rigorous, combining:

- Expert reviews to refine rules and membership functions.
- Simulation-based testing in MATLAB, exploring boundary conditions and ensuring stable system behavior across all inputs.
- Benchmarking against published SLA policies from major CSPs, confirming practical alignment.

Iterative adjustments were made based on simulation results and feedback, ensuring robustness and accuracy in the final system. This integrated methodology provides a precise, adaptive, and context-aware tool for SLA classification, enhancing decision-making and resource allocation in modern cloud environments.

6.3 Experimental Evaluation

The proposed fuzzy logic-based model for SLA classification was thoroughly tested in MATLAB to assess its effectiveness in evaluating cloud services based on Quality of Service (QoS) metrics. The model integrates key computing parameters—including vCPU, RAM, and storage—with networking metrics like bandwidth, delay, jitter, and packet loss, translating them into SLA classifications using a Mamdani-type fuzzy inference system. Inputs are fuzzified over a continuous domain from 90% to 100% using triangular membership functions, enabling nuanced analysis of availability conditions. The system categorizes services into SLA tiers—Normal, Bronze, Silver, and Gold—each further divided into sublevels (e.g., Normal-SLA 1 to 3). As availability conditions improve, services are progressively assigned higher SLA levels, reflecting increased reliability and performance. This dynamic framework allows cloud users to match service choices to performance needs and budget constraints. Figure 6.1 provides outputs demonstrating how the continuous fuzzy mapping function transitions through SLA levels as input metrics rise. The model’s experimental results confirm that it offers precise, adaptable SLA predictions, supporting informed decision-making for diverse cloud applications and ensuring alignment between user requirements and service guarantees.

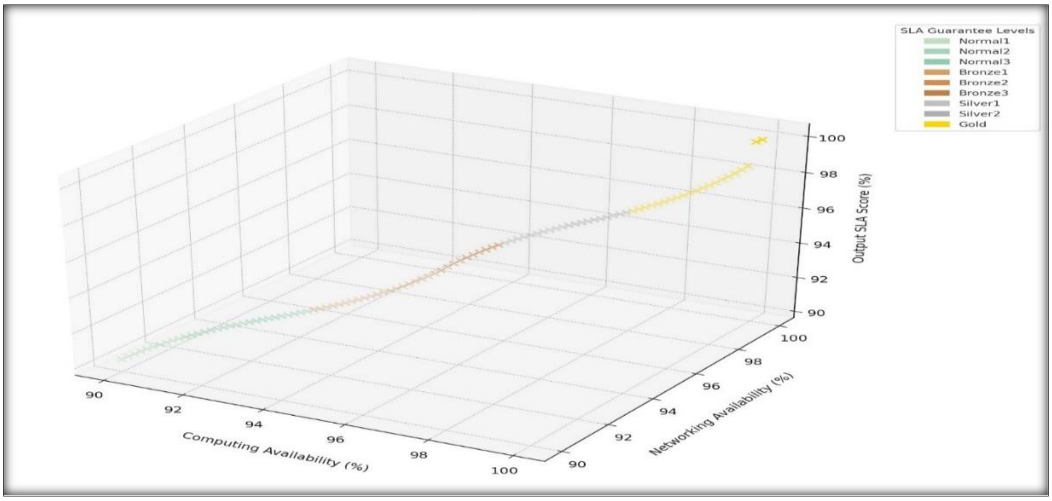

Figure 6.1 Results of the proposed model.

6.4 Summary of SLA Selection Model

The proposed fuzzy logic-based SLA selection model offers a dynamic approach for aligning user preferences with optimal SLA classifications in real-time. Unlike static, provider-defined SLA frameworks, it accommodates uncertainties in computing and networking performance, providing flexibility and personalized service decisions. The model introduces a systematic method for calculating availability ratios, enabling Cloud Service Providers (CSPs) to deliver tiered offerings tailored to diverse user needs. A significant innovation is the model's ability to generate continuous SLA mappings, moving beyond fixed definitions to reflect real-world variability in service performance. Experimental validation in MATLAB involved simulating over 100 paired computing and networking QoS inputs, producing granular availability scores (e.g., 90.333%, 95.999%, 99.999%) that map precisely to industry-standard SLA categories like Normal, Bronze, Silver, and Gold, as shown in Figure 6.1. These results align closely with published SLA commitments from providers such as AWS EC2, confirming the model's practical relevance and classification accuracy. The dynamic mapping ensures that users receive SLA guidance matched to current network and computing conditions, supporting informed decision-making and compliance. Future work focuses on refining fuzzy logic models with optimization techniques, aiming to enhance SLA management and efficient cloud service allocation in complex, dynamic environments.

7. Enhanced Decision-Making in Uncertain Domains

This study presents an innovative mathematical methodology for enhancing decision-making in uncertain environments by introducing three specialized algorithms that mathematically define membership functions for fuzzy logic systems. Unlike traditional fuzzy approaches that rely on heuristic tuning or specialized software like MATLAB's Fuzzy Toolbox, this chapter proposes precise analytical algorithms for computing membership degrees for triangular, trapezoidal, and Gaussian membership functions, thus increasing computational efficiency and independence from external tools. The novelty lies in replacing heuristic adjustments with structured mathematical optimization, enabling precise and dynamic classification of crisp input values into fuzzy sets, and improving accuracy and adaptability in practical applications. The chapter begins by reviewing challenges faced by fuzzy logic systems, including complexities in rule formulation and inefficiencies in computation. While fuzzy logic has been widely integrated into control systems, robotics, and AI to handle imprecision and ambiguity, traditional systems remain heavily dependent on subjective parameter tuning, software dependencies, and lack robust optimization techniques [72]. Previous efforts, such as combining fuzzy logic with machine learning or genetic algorithms, have attempted to overcome these challenges but often suffer from issues like convergence instability, complexity, and computational overhead [73][74]. To address these shortcomings, the proposed approach defines explicit mathematical methods for constructing membership functions, eliminating reliance on manual or software-driven parameter adjustments. These methods ensure faster, more precise classification of inputs, offering high adaptability for various AI applications without requiring specialized tool environments. The algorithms were implemented and tested in MATLAB on a dataset of 10,000 task-size entries, successfully categorizing them into small, medium, or large classes. Results demonstrated equivalence or superior precision compared to Mamdani fuzzy systems, validating the proposed approach's efficiency and accuracy in practical scenarios. The chapter also emphasizes the broader implications of these mathematical methods for AI, particularly in environments where traditional fuzzy logic implementations are impractical due to software dependencies or computational limitations. By systematically computing membership degrees through deterministic calculations rather than heuristic tuning, the methodology enables more scalable, efficient, and tool-independent decision-making processes. This development significantly advances the practical applicability of fuzzy logic, bridging theoretical advancements with real-world implementation and providing a robust alternative to traditional fuzzy systems in complex, uncertain domains [75]. Overall, Chapter 6 contributes a groundbreaking method for optimizing fuzzy logic systems, offering a mathematically rigorous, efficient, and adaptable framework for decision-making across diverse AI and engineering applications.

7.1 Background of Fuzzy Logic System

The outlines the fundamental principles and architecture of fuzzy logic systems, emphasizing their capability to handle uncertainty and partial truths—a stark contrast to binary logic's rigid true/false dichotomy. Unlike crisp sets where membership is strictly 0 or 1, fuzzy sets allow values between 0 and 1, enabling nuanced reasoning similar to human decision-making, often termed "computing with words" [76][77]. Figure 7.1 illustrates the typical architecture of a fuzzy logic system, comprising several core components. First, **crisp input processing** transforms precise inputs into either binary (crisp) values or fuzzy sets. Crisp sets use an indicator function assigning 1 for membership and 0 otherwise. Next, **fuzzification** converts crisp inputs into fuzzy sets using linguistic terms and membership functions, forming the basis for flexible reasoning [78]. The **inference engine** then applies a set of IF-THEN rules from a **fuzzy rule base**, matching fuzzified inputs to conditions and aggregating conclusions to produce fuzzy outputs [79]. Finally, **defuzzification** converts fuzzy outputs into actionable crisp values, ensuring practical application. The centroid method, as described in Equation 6.2, calculates a precise value by averaging outputs weighted by their membership degrees [80].

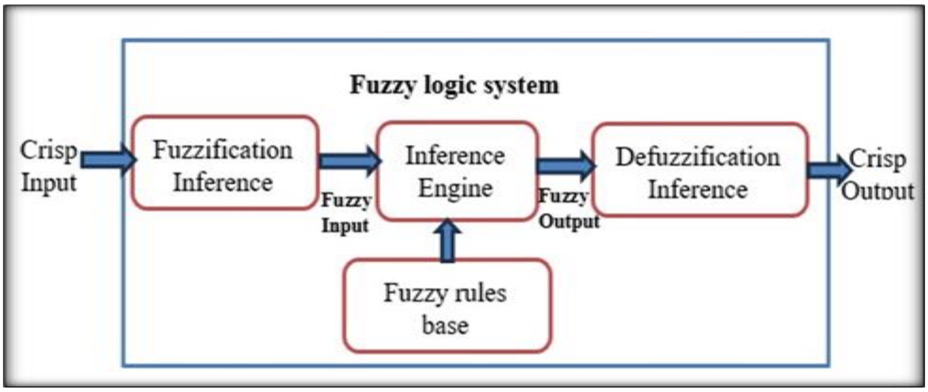

Figure 7.1 Architecture of a fuzzy logic system.

Crucial to this entire process are **membership functions**, which determine how strongly an input belongs to a fuzzy set, providing a smooth mapping from inputs to truth values in [0,1] [81]. Three widely used functions include:

- **Triangular membership functions**, defined by parameters {a, b, c}, forming a simple linear slope.
- **Trapezoidal membership functions**, defined by {a, b, c, d}, allowing flat-topped regions for broader intervals of full membership [Equation 7.1].

$$\mu_{F(x)} = \begin{cases} 0; \; x \leq a \\ \frac{x-a}{b-a} \; ; a < x < b \\ 1; b \leq x \leq c \\ \frac{d-x}{d-c}; \; c < x < d \\ 0; x \geq d \end{cases} \tag{7.1}$$

- **Gaussian membership functions**, producing smooth bell-shaped curves for gradual transitions in membership, expressed mathematically by an exponential function centered around a mean value with a specified width [Equation 7.2].

$$\mu A(x) = e^{-(\frac{x-c}{\sigma})^2} \tag{7.2}$$

These components collectively empower fuzzy logic systems to handle ambiguous, complex real-world problems across diverse applications, such as AI, control systems, and decision-making environments.

7.2 Methodology

This study introduces a precise, mathematically grounded methodology for enhanced decision-making in uncertain domains, replacing traditional heuristic fuzzy logic implementations with systematic calculations. The foundation of the approach remains the Mamdani fuzzy inference system (FIS), a Max-Min method that transforms linguistic rules into actionable outputs by evaluating the degrees of membership of crisp input values across multiple fuzzy sets [23]. This mechanism ensures a nuanced understanding of how each input contributes to decision-making.

The proposed methodology consists of three specialized algorithms designed to calculate membership functions analytically, improving precision and computational efficiency.

- **Algorithms 1 & 2** utilize geometric principles to define **triangular and trapezoidal membership functions.** Both rely on the mathematics of linear functions. For triangular and trapezoidal shapes, the method applies line equations such as the slope-intercept form [Equation 7.3] to construct membership function boundaries. Absolute value functions [Equations 7.4] are employed to manage slope variations and symmetrical properties.

$$y\text{-}y1\text{=}m(x\text{-}x1) \tag{7.3}$$

$$f(x) = \begin{cases} x, \; if \;\; x > 0 \\ 0, \; if \;\; x = 0 \\ -x, \; if \; x < 0 \end{cases} \tag{7.4}$$

- **Algorithm 3** focuses on **Gaussian membership functions**, grounded in probability theory. It uses the Gaussian probability density function (PDF) [Equation 6.11], parameterized by the mean (m) and standard deviation (σ). Variance calculations [Equation 7.5] ensure accurate modeling of uncertainty, making Gaussian functions suitable for representing smoothly varying memberships in fuzzy sets [82][83].

$$fX^{(x)} = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (7.5)$$

Together, these algorithms systematically categorize input variables within a defined universe of discourse, transforming crisp values into fuzzy membership degrees with mathematical precision. The process significantly reduces reliance on external toolboxes (e.g., MATLAB's fuzzy toolbox), thereby enhancing portability and computational speed. The second and third algorithms operate within the same input domain in the proposed system but differ in how they structure and apply conditions to define and classify the five membership functions used in this study. This methodology establishes a robust framework for determining membership degrees, directly enabling precise classification of inputs under uncertainty, and enhancing decision-making processes across diverse AI and control applications.

Algorithm 1: Input Partitioning and Membership Classification as similar work as Triangular MF
//Membership degrees are calculated for each input value V_i with respect to membership functions defined over the universe of discourse. *//The parameters PV (parameter values) is defining the shape, boundaries, or centers of the membership functions—not the input data itself.* • *Input: V, a set of crisp input values for which the degree of membership will be calculated.* *//Parameters: Definitions of membership functions (PVs specifying boundaries, centers, or slopes for triangular MFs.* *n: The number of fuzzy partitions (i.e., number of membership functions) into which the universe of discourse is divided.* • *Output:* *A matrix of membership degrees $\mu(v_i)$ for each v_i across all defined membership functions.* *Procedure:* *1. Initialization:* • *Max(V) $\leftarrow$ max(Vi) // Calculate the maximum value of sets V in the universe discourse.* *2. Parameter Value Calculation:* • *$PV_1 \leftarrow (Max(V)/n)$ // Determine the first parameter value.* • *$PV_n \leftarrow n \times PV_1$ // Compute the last parameter value.* *3. 3. Iterate Over Each Input Value V_i in the Set of Parameter Values:* *for each $V_i \in V$:* • *Case 1:if $V_i \geq 0$ and $V_i \leq PV_1$* *$MF_1 \leftarrow (\frac{-V_i}{PV2})+1$; Output $\leftarrow$ (MF_1, Degree (V_i))* *//Compute Membership Function 1.* *Output $\leftarrow$ ($MF_2, MF_3, \ldots, MF_{m-1}$, Degree($V_i$)) // Determining the degree of element in the remaining MF domain.* • *Case 2: if $V_i \geq PV_1$ and $V_i \leq PV_2$* *$MF_1 \leftarrow (\frac{-V_i}{PV2})+1$; Output $\leftarrow$ (MF_1, Degree (V_i))* *// Compute the degree of element affiliated with both domains MF_1 and Subsequent it, as MF_2.* *$\alpha \leftarrow (V_i - PV_2)$ // Define the alpha variable.*

$MF_2 \leftarrow (\frac{-1}{PV2-PV1}) \times (|\alpha| + 1)$

// Compute the degree of element affiliated with both domains MF_2 and previous it, as MF_1.

Output $\leftarrow$ *(MF_3, MF_4, …,MF_{m-1}, Degree(V_i))*

//Determining the element's degree of membership across the remaining membership functions.

- *Case 3: if $V_i \geq PVn - 1$ and $V_i \leq PV_n$*

$MF_m \leftarrow ((\frac{1}{PVn-PVn-1}) \times (Vi - Pn - 1)$; *Output* $\leftarrow$ *(MF_m, Degree (V_i))*

// Calculate Membership Function m.

Output $\leftarrow$ *(MF_1,MF_2, …,MF_{m-1}, Degree(V_i))*

//Determining the element's degree of membership across the remaining membership functions.

4.End of Algorithm 1

7.3 Experimental Results and Analysis

This study presents the experimental validation of the proposed fuzzy logic methodology, using a large dataset of over 10,000 user tasks extracted from the Parallel Workloads Archive [1]. This dataset comprises job-level usage data from supercomputers, clusters, and grid systems, including diverse task sizes ranging from 0 to 67,170 bytes. These tasks, categorized as “small,” “medium,” and “big,” represent unstructured and varying user demands, providing a realistic basis for testing fuzzy classification approaches. The data were used in their raw format without preprocessing, ensuring authenticity for experimental analysis. MATLAB® (R2018b) was employed for computational modeling, leveraging its strong capabilities for mathematical analysis, data handling, and visualization. The first part of the experimental analysis focuses on applying the proposed triangular membership function, implemented through the first algorithm. This method determines the degree of membership of task sizes across the defined universe of discourse, effectively classifying inputs based on fuzzy logic principles. Results are visually presented in Figures 7.2, which demonstrate triangular membership functions.

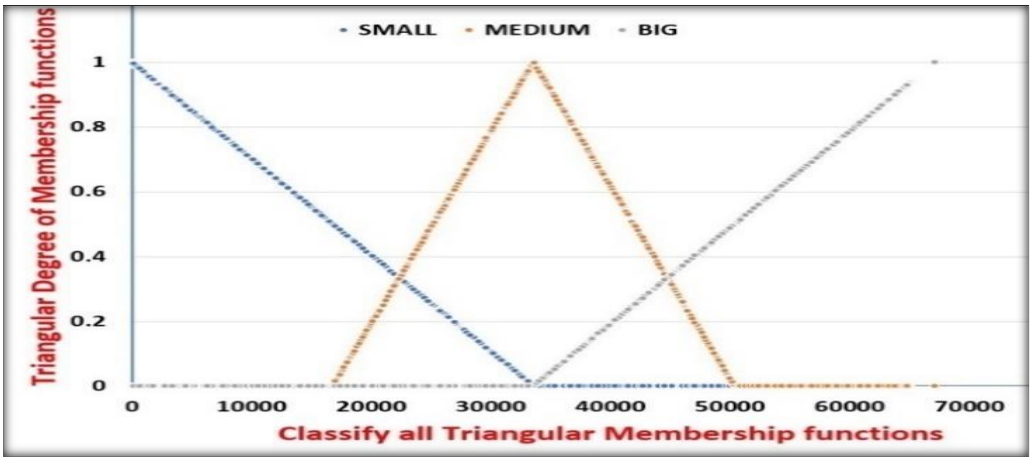

Figure 7.2 Classify all Triangular MF.

The second experimental segment utilizes the trapezoidal membership function, implemented through the second algorithm, to classify task sizes. This approach assigns membership values based on trapezoidal curves, enabling smooth transitions between classes while maintaining precise delineation of task categories. Figures 7.3 display the classification outcomes for combined trapezoidal functions. underscoring the proposed method's ability to deliver precise, interpretable classifications while effectively modeling uncertainty.

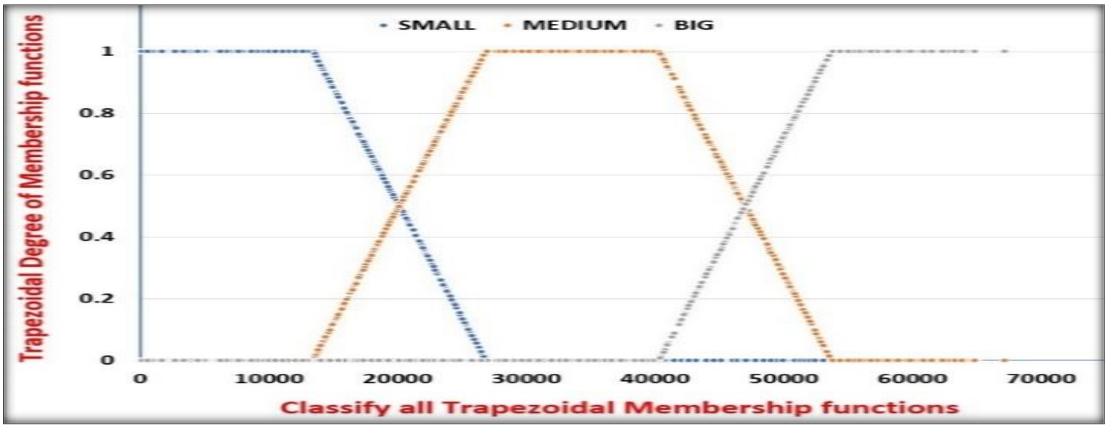

Figure 7.3 Classify all Trapezoidal MF.

In the final experimental section, the Gaussian membership function is applied using the third algorithm. Recognized for its smooth, continuous curves, the Gaussian function offers high precision in membership degree assignment, crucial for nuanced classification of overlapping task sizes. Figures 7.4 Gaussian membership function classifications, this demonstrates how the proposed model integrates fuzzy inference with probabilistic modeling for more refined and consistent outcomes.

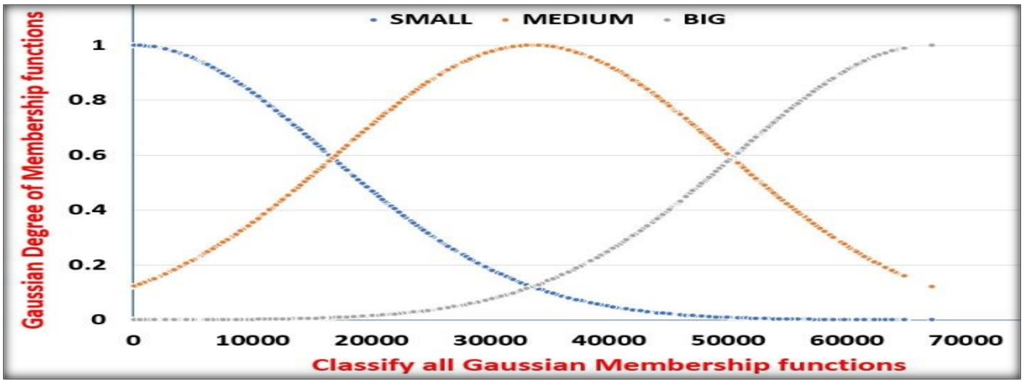

Figure 7.4 Classify all Gaussian MF.

Overall, these experimental evaluations confirm the robustness and adaptability of the proposed mathematical fuzzy logic framework. Compared to traditional Mamdani systems, the new algorithms provide systematic, mathematically grounded processes for classifying data under uncertainty, ensuring precise, flexible, and computationally efficient task classification suitable for cloud computing and other AI applications.

7.4 Validation-Based Comparative Analysis

This study presents a validation study comparing the classical Mamdani fuzzy inference system (FIS) to the proposed mathematical model featuring newly developed algorithms for fuzzy classification. Unlike traditional Mamdani systems that rely on heuristic or manual tuning, the proposed model employs precise mathematical formulations for constructing triangular, trapezoidal, and Gaussian membership functions. These formulations—based on point-slope line equations, linear interpolation, and probabilistic Gaussian distributions—enable systematic input classification and accurate computation of membership degrees, improving computational efficiency and scalability while retaining the interpretability of fuzzy systems. A comparative analysis was performed using ten representative input samples from the universe of discourse, as well as a larger dataset of 10,000 inputs representing diverse task sizes. Results, presented in Tables 7.1 and 7.2, reveal that the proposed model consistently achieves greater precision and adaptability, especially in handling complex and uncertain inputs, compared to the Mamdani FIS. Moreover, the new approach reduces dependency on specialized simulation software, thus minimizing computational resources and storage requirements. This validation confirms that the proposed mathematical framework significantly advances fuzzy logic applications, offering robust, scalable, and efficient solutions for modern intelligent systems.

Table 7.1 Results of the Proposed Method Applied to Selected Samples.

Samples of Degree of Triangular Membership Function			
value	small	medium	big
0	1	0	0
16823	0.499091856	0.001816287	0
17129	0.489980646	0.020038708	0
17361	0.4830728	0.033854399	0
17579	0.476581807	0.046836385	0
25978	0.226499926	0.547000149	0
26931	0.198124163	0.603751675	0
28842	0.141223761	0.717552479	0
31475	0.062825666	0.874348668	0
33565	0.000595504	0.998808992	0
Samples of Degree of Trapezoidal Membership Function			
value	small	medium	big
20162	0.499181182	0.500818818	0
21582	0.393479232	0.606520768	0
23875	0.222792914	0.777207086	0
25331	0.114411195	0.885588805	0
26846	0.001637636	0.998362364	0
46120	0	0.566919756	0.433080244
45451	0	0.616718773	0.383281227
44329	0	0.700238202	0.299761798
42852	0	0.810183117	0.189816883
40336	0	0.997469108	0.002530892
Samples of Degree of Gaussian Membership Function			
value	small	medium	big
0	1	0.120934543	0.000213895
1	0.999999998	0.120949757	0.000213949
10090	0.826402652	0.355634634	0.002238294
32026	0.146469985	0.995458374	0.098946015
49791	0.009627715	0.611475933	0.567984183
54045	0.004209592	0.456574063	0.724241188
61138	0.000911417	0.241274197	0.934125619
64852	0.000379417	0.160259114	0.989987311

65069	0.000359903	0.156223736	0.991766863
67170	0.000213895	0.120934543	1

Table 7.2 Results of the Traditional Method Applied to Selected Samples.

Samples of Degree of Triangular Membership Function			
value	small	medium	big
0	1	0	0
16823	0.499076941,400667	0.001846117,1986660315	0
17129	0.489965459,74273464	0.020069080,51453073	0
17361	0.483057408,28966176	0.033885183,420676514	0
17579	0.476566222,01048116	0.046867555,979037634	0
25978	0.226476893,75893282	0.547046212,4821344	0
26931	0.198100285,8504	0.603799428,2991901	0
28842	0.141198189,61410197	0.717603620,7717961	0
31475	0.062797760,83849452	0.87440447,83230109	0
33565	0.000565745,5931395903	0.998868508,8137208	0
Samples of Degree of Trapezoidal Membership Function			
value	small	medium	big
20162	0.499181182,07533124	0.500818817,9246688	0
This table extends and complements the information presented in Table 6.2.			
21582	0.393479231,7999107	0.606520768,2000894	0
23875	0.222792913,50305197	0.777207086,496948	0
25331	0.114411195,47417002	0.885588804,52583	0
26846	0.001637635,849337502	0.998362364,1506625	0
46120	0	0.783443757,9096255	0.216556242,09037444
45451	0	0.808345120,2263083	0.19165487,977369167
44329	0	0.850107943,1251396	0.149892056,87486043
42852	0	0.905084493,4117472	0.094915506,5882528
40336	0	0.998734459,9121566	0.001265540,0878433707
Samples of Degree of Gaussian Membership Function			
value	small	medium	big
0	1	0.122	0.0002
1	1	0.122	0.0002
10090	0.8418	0.7201	0.0053
32026	0.2931	0.996	0.1097
49791	0.0304	0.5364	0.7211
54045	0.0124	0.2917	0.8431
61138	0.0028	0.1097	0.9959
64852	0.0011	0.0566	0.9881
65069	0.0010	0.0532	0.9926
67170	0.0002	0.1218	1

7.5 Summary

This chapter presented and validated a new mathematical framework for precise fuzzy classification, introducing three algorithms for computing triangular, trapezoidal, and Gaussian membership functions. Unlike traditional Mamdani systems, the model offers systematic input partitioning and efficient membership degree computation, enhancing accuracy, computational speed, and robustness while preserving interpretability. Extensive testing with over 10,000 samples confirmed the model's superior performance in classifying tasks into small, medium, or big categories. The approach is well-suited for diverse AI applications, such as QoS management, and lays groundwork for future research in high-precision, scalable decision-making systems, including integration into IVCBS environments.

8. Intelligent Validation Cloud Broker System

This study presents advancements in Service Level Agreement (SLA) selection through the Intelligent Validation Cloud Broker System (IVCBS), which integrates mathematical modeling with cloud resource management. The proposed system employs mathematical formulations akin to trapezoidal membership functions, using linear equations to define membership degrees for input variables such as VM attributes and user request sizes. This approach enhances resource classification accuracy, optimizes response time, and lowers VM costs and data center processing times. Simulations confirm that IVCBS, especially with the "Optimize Response Time" policy, surpasses traditional methods in metrics like response time, VM costs, and energy efficiency, enabling more cost-effective cloud resource allocation [80][81]. Cloud computing's scalability and flexibility hinge on service models like IaaS, PaaS, and SaaS [82]. Yet, its dynamic nature demands sophisticated SLA management to address resource conflicts and fluctuating user demands [83]. Traditional brokers often fall short in trust, efficiency, and dynamic service matching [84]. Prior research has explored fuzzy logic for resource allocation, leveraging methods like Fuzzy-RLVMrB and PRSF to enhance load balancing and reduce energy use [85]. Simulation tools like CloudSim and Cloud Analyst enable comprehensive modeling of such strategies [86]. The IVCBS refines resource allocation by converting continuous fuzzy membership values into binary decisions for real-time operations. Inputs exceeding a defined threshold receive a score of one, ensuring validated allocation only for the most suitable resources. This two-stage system combines fuzzy classification with crisp decision-making, improving operational efficiency. Testing across six AWS regions with one million users and diverse EC2 instances confirmed IVCBS's superior performance in global cloud environments [87][88]. The chapter also situates IVCBS amid research exploring brokers' roles in SLA negotiation, cloud security, multi-cloud interoperability, and load balancing [27][89]. IVCBS stands out as an adaptive, mathematically grounded solution, addressing gaps in traditional SLA selection while supporting scalable and efficient cloud service management.

8.1 the Proposed System

This study introduces the proposed Intelligent Validation Cloud Broker System (IVCBS), an advanced framework aimed at improving intelligent service identification, resource allocation, and SLA optimization in cloud computing. The system validates only those resources or user requests that achieve a uniform membership score of 1, ensuring precision and reliability in decision-making, as demonstrated by the classification and matching algorithms. This approach optimizes cloud resource management and promotes high service efficiency.

Classification Algorithm
Inputs: Parameter Value (PV)set= {PV1, PV2,,,PV11} *Output=Classification with order Parameter Values.* *//Compute the level for each input parameters.* *1.For each input value (V) from input parameter value set* *2.IF (V >=PV1 and V <=PV2)* *3.MF1 ← (((-1/PV1-PV2)) *((V-PV2))) +1)* *//MF: Membership Functions* *4.Output ← (Poor, MF1)* *5.Output ← ((Fair, Good, V. Good, Excellent),0)* *6.End*

```
IF(V>PV2 and V<=PV3)
MF1 ← 1
Output ← (Poor, MF1)
Output ← ((Fair, Good, V. Good, Excellent),0)
End
IF (V>PV3 and V<=PV4)
MF1 ← (((-1/(PV4-PV3)) *((V-PV3))) +1)
Output ← (Poor, MF1)
Output ← ((Good, V. Good, Excellent),0)
MF2 ← (((-1/PV3-PV4)) *((V-PV4))) +1)
Output ← (Fair, MF2)
End
IF(V>PV4 and V<=PV5)
MF2 ←1
Output ← (Fair, MF2)
Output ← ((Poor, Good, V. Good, Excellent),0)
End
IF(V>PV5 and V<=PV6)
MF2 ← (((-1/(PV6-PV5)) *((V-PV5))) +1)
Outputç ←(Fair, MF2)
Output ← ((Poor, V. Good, Excellent),0)
MF3 ← (((-1/PV5-PV6)) *((V-PV6))) +1)
Output ← (Good, MF3)
Output ← ((Poor, V. Good, Excellent),0)
End
IF (V>PV6 and V<=PV7)
MF3 ←1
Output ←(Good, MF3)
Output ← ((Poor, Fair, V.Good, Excellent),0)
End
IF (V>PV7 and V <=PV8)
MF3 ← (((-1/(PV8-PV7)) *((V-PV7))) +1)
Output ← (Good, MF3)
Output ← (Poor, Fair, Excellent),0)
MF4 ← (((-1/(PV7-PV8)) *((V-PV8))) +1)
Output ← (V. Good, MF4)
Output ←(Poor, Fair, Excellent,0)
End
IF (V>PV8 and V<=PV9)
MF4 ←1
Output ←(V. Good, MF4)
Output ← ((Poor, Fair, Good, Excellent),0)
End
IF (V>PV9 and V<=PV10)
MF4 ← (((-1/(PV10-PV9)) *((V-PV9))) +1)
Output ← (V. Good, MF4)
Output ← ((Poor, Fair, Good),0)
MF5 ← (((-1/(PV9-PV10)) *((V-PV10))) +1)
Output ← (Excellent, MF5)
Output ← ((Poor, Fair, Good),0)
```

57.End
58.IF (V>PV10 and V<=PV11)
59.MF5←1
60.Output← (Excellent, MF5)
61.Output← (Poor, Fair, Good, V.Good),0)
62.End
63.End

Matching Algorithm

1.IF Output (Poor, PV1)

2.Assign: User base Request (App1) ← M6g.medium

3.End

4.IF Output (Poor, PV2)

5.Assign: User base request (App2) ← M6g.large

6.End

7.IF Output (Poor, PV3)

8.Assign: User base request (App3) ← M6g.XLarge

9.End

10.IF Output (Fair, PV4)

11.Assign: User base request (App4) ← M5.2XLarge

12.End

13.IF Output (Fair, PV5)

14.Assign: User base request (App5) ← M5.4XLarge

15.End

16. IF Output (Good, PV6)

17.Assign: User base request (App6) ← M6gd.8XLarge

18.End

19.IF Output (Good, PV7)

20.Assign: User base request (App7) ← M6gd.12XLarge

21.End

22.IF Output (V. Good, PV8)

23.Assign: User base request (App8) ← M6g.metal

24.End

25.IF Output (V. Good, PV9)

26.Assign: User base request (App9) ← M5d.metal

27.End

28.IF Output (Excellent, PV10)

29.Assign: User base request (App10) ← M6i.metal

30.End

31.IF Output (Excellent, PV11)

32.Assign: User base request (App11) ← M6a.metal

33.End

This study focuses on AWS General Purpose EC2 instances, deployed across 31 global data centers in six regions, as shown in Figure 8.1. AWS provides 212 EC2 instance types categorized into 11 groups, offering a balanced mix of compute, memory, and networking resources to support diverse workloads, including web servers and repositories [90]. Cost data and technical specifications are compiled in various tables, such as Table 8.1, which details AWS general-purpose EC2 instances, and Table 8.2, which presents user request sizes, establishing a solid empirical foundation for simulation and analysis. At the core of the IVCBS is a mathematical model based on trapezoidal membership functions. Instead of relying on heuristic tuning, this model employs linear and absolute-value equations to emulate trapezoidal curves, allowing precise computation of membership scores (either 0 or 1) for inputs like VM attributes (vCPU, RAM, storage, bandwidth) and user request sizes [76][91]. Inputs classified as "Poor," "Fair," "Good," "Very Good," or "Excellent" guide decisions on resource allocation, with only those scoring 1 being validated for deployment. This rigorous classification ensures alignment with user-specific SLA requirements while improving system efficiency. The system implements this model in MATLAB, where algorithms systematically assign membership degrees to inputs, ensuring consistent decision-making across VM resources and user request sizes. This approach demonstrates how the mathematical functions mirror the behavior of trapezoidal membership structures in determining and classifying membership degrees.

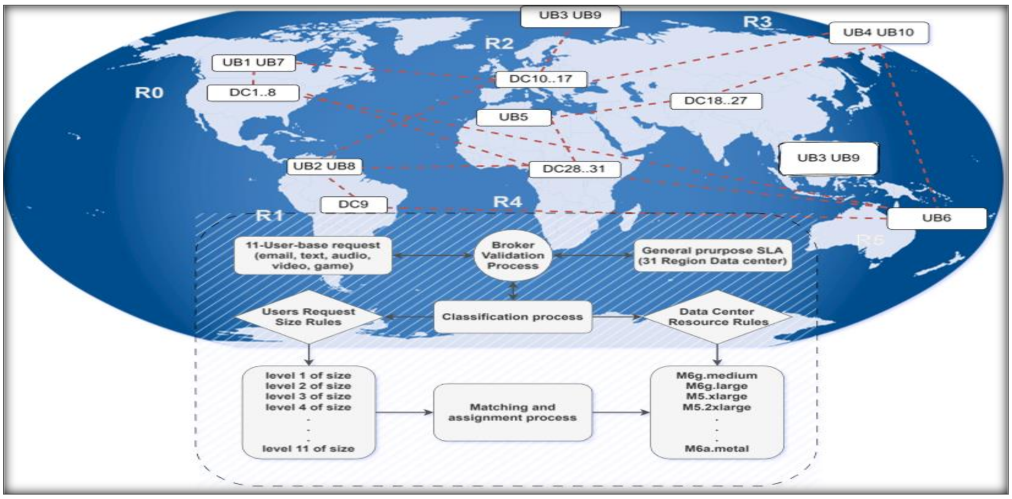

Figure 8.1 Intelligent Validation Cloud Broker System Framework.

Table 8.1 AWS-General purpose instance features.

AWS-General-Purpose series Attributes and specs					
EC2- Series	VCPU	RAM GB	Storage GB	Bandwidth Gbps	VCPU-clock speed GHz
M6g.medium	1	4	1	2	2
M6g.Large	2	8	2	4	2
M6g.Xlarge	4	16	4	8	2.4
M5.2XLarge	8	32	8	10	2.5
M5.4XLarge	16	64	12	12	2.5
M6gd.8XLarge	32	128	16	14	2.5
M6gd.12XLarge	48	192	24	16	2.7
M6g.metal	64	256	32	18	2.7
M5d.metal	96	384	48	24	3.4
M6i.metal	128	512	64	30	3.4
M6a.metal	192	768	88	40	3.4

Table 8.2 Cloud users and sizes of their requests.

Cloud users		User request	
Scenario number	Total number of users	SaaS	Size
1	1000,000	App1	3 MB
2	1000,000	App2	5 MB
3	1000,000	App3	10 MB
4	1000,000	App4	35 MB
5	1000,000	App5	70 MB
6	1000,000	App6	105 MB
7	1000,000	App7	140 MB
8	1000,000	App8	750 MB
9	1000,000	App9	1500 MB
10	1000,000	App10	2250 MB
11	1000,000	App11	3000 MB

To manage workloads, the proposed system deploys the Round-Robin (RR) algorithm. RR ensures fair distribution of user requests across VM clusters by cycling through servers sequentially, with each receiving an equal time quantum [92][93]. This ensures balanced utilization and mitigates bottlenecks during high-demand periods. For resource allocation strategies, IVCBS integrates two service broker policies (SBPs): (1) optimizing response time and (2) dynamic reconfiguration based on load [94][95]. The first policy continuously evaluates data center performance, routing user requests to the center with the lowest response time, thereby ensuring minimal latency [96][86]. The second policy dynamically adjusts VM deployments based on real-time load assessments. It redistributes workloads across data centers if performance dips below thresholds, ensuring balanced resource usage and cost efficiency [97][98]. The IVCBS leverages the CloudAnalyst tool for simulation, extending the capabilities of CloudSim. CloudAnalyst facilitates modeling of geographically distributed systems under varied workloads, producing detailed outputs in tables and charts to analyze response times, costs, and data center processing metrics [99]. This proposed framework builds upon existing research on intelligent decision-making and broker-based systems. It offers significant advancements over traditional methods by eliminating heuristic dependency, enhancing computational efficiency, and ensuring high-precision SLA alignment. As a result, the IVCBS contributes substantially to improving cloud resource allocation, user satisfaction, and operational efficiency in complex cloud environments.

8.2 Experimentation and Analysis

Details the experimental evaluation of the proposed Intelligent Validation Cloud Broker System (IVCBS) using Cloud-Analyst to simulate the “Optimize Response Time” policy. The experiment processed 1,000,000 user requests distributed across ten user bases and 31 AWS data centers in six global regions, each configured with a single VM based on 11 EC2 specifications. Network delays were standardized using AWS latency data. User workloads varied from 10,000 off-peak to 100,000 peak users per base, determining metrics such as data size per request and instruction length. Round-robin load balancing was employed within data centers to manage VM workloads. Each of the 31 data centers was tested across 11 scenarios reflecting AWS General Purpose EC2 instances. The proposed IVCBS was benchmarked against traditional random allocation methods, which indiscriminately assign VM resources without considering request characteristics. Both systems were tested under two policies: optimizing response times and dynamic load reconfiguration. Results indicated that the IVCBS’s strategic allocation approach outperforms traditional methods, offering improved resource efficiency and responsiveness while maintaining consistent EC2 specifications.

8.3 Results and Comparative Analysis

This study presents a thorough evaluation of the Intelligent Validation Cloud Broker System (IVCBS) under two service broker policies—Optimized Response Time and Dynamic Reconfiguration with Load Balancing (LB)—implemented via the CloudAnalyst simulator. The IVCBS intelligently routes user requests from ten user bases (UBs) to 31 AWS data centers globally, adjusting resource allocation to match workload demands using EC2 instance types, such as EC2-M6a.metal for high-volume tasks. Simulations revealed that the Optimized Response Time Policy consistently outperforms Dynamic Reconfiguration in metrics such as Average Overall Response Time, Average Data Center Processing Time, and Total VM Cost, though Data Center Request Servicing Times were comparable or slightly higher under optimized routing, indicating a nuanced performance trade-off. Results, illustrated in Tables 8.3 and 8.4, underscore the optimized policy’s ability to lower response times by globally distributing requests rather than confining them to geographically close data centers, as in the Dynamic Reconfiguration approach. The optimized strategy also improves energy efficiency, as longer servicing times in dynamic reconfiguration can lead to higher energy consumption due to inefficient processor and memory utilization and greater cooling demands. In contrast, traditional methods lack the intelligent classification mechanisms of IVCBS.

Table 8.3 Implementing IVCBS with optimize response time policy.

AWS-EC2	Overall Response Time (ms)	Data Center Processing (ms)	Total VM Cost ($)	Total Data Transfer Cost ($)
M6g.medium	2475,8	2373,38	83,29	$298,59
M6g.Large	3853,10	3740,25	167,24	497,65
M6g.Xlarge	14325,08	10798,69	334,48	1255,96
M5.2XLarge	140667,03	137632,98	853,50	3483,46
M5.4XLarge	1010570,86	1031103,10	1707,06	6963,47
M6gd.8XLarge	2151917,72	1947568,70	3140,37	9966,88
M6gd.12XLarge	3684599,83	3335444,58	4709,26	13114,84
M6g.metal	38334990,80	38234416,58	5351,62	25236,98
M5d.metal	79337433,27	79315311,43	12090,55	14482,63
M6i.metal	93529270,35	93372293,67	13730,36	6863,40
M6a.metal	94549552,26	94331238,90	17150,67	3320,20

Table 8.4 Implementing IVCBS with Dynamic Reconfiguration Load Service Broker Policy.

AWS-EC2	Overall Response Time (ms)	Data Center Processing (ms)	Total VM Cost ($)	Total Data Transfer Cost ($)
M6g.medium	6353,58	6324,05	166,32	$298,59
M6g.Large	55390,42	55364	667,5	497,65
M6g.Xlarge	275390,88	270714,32	2666,54	1255,83
M5.2XLarge	2556092	2556270,05	8502,06	3483,45
M5.4XLarge	3252254,20	3255057,05	20401,48	6234,76
M6gd.8XLarge	3915809,21	3921022,05	43758,17	8915,92
M6gd.12XLarge	3573677,62	3584236,77	74944,34	11618,91
M6g.metal	37016372,94	37016688,54	95138,79	25828,65
M5d.metal	81818244,66	81883142,21	273382,89	14705,94
M6i.metal	93919067,50	93689019,40	379237,75	6796,75
M6a.metal	96334126,87	96128434,12	607000,72	3341,66

They distribute user requests randomly across 31 data centers without considering VM specifications or workload sizes, leading to inefficiencies such as assigning high-cost EC2-M6a.metal instances to handle small tasks that could be processed more efficiently by EC2-M6g.medium machines. Consequently, traditional approaches showed higher average response times and VM costs than IVCBS, despite sometimes achieving lower total data transfer costs, as detailed in Tables 8.5 and 8.6. Visual comparisons confirm the superiority of IVCBS, particularly in managing data center request servicing times and enhancing energy efficiency.

Table 8. 5 Implementing traditional with optimize response time policy.

AWS-EC2	Overall Response Time (ms)	Data Center Processing (ms)	Total VM Cost ($)	Total Data Transfer Cost ($)
M6g.medium	2648,32	2544,20	5039,17	298,59
M6g.Large	3979,79	3866,43	5039,17	497,65
M6g.Xlarge	16565,20	16507,91	5039,17	995,31
M5.2XLarge	200877,44	206148,60	5039,17	3483,25
M5.4XLarge	1012024,16	1045751,95	5039,17	6965,51
M6gd.8XLarge	2784038,22	2523254,74	5039,17	9907,33
M6gd.12XLarge	4246474,38	3977103,11	5039,17	13054,04
M6g.metal	44420610,74	43609256,19	5039,17	17375,69
M5d.metal	80927473,71	80639117,03	5039,17	7093,73
M6i.metal	95412416,34	95769447,44	5039,17	3711,87
M6a.metal	97606171,17	98736234,17	5039,17	1686,10

Table 8.6 Implementing traditional with Dynamic reconfiguration policy.

AWS-EC2	Overall Response Time (ms)	Data Center Processing (ms)	Total VM Cost ($)	Total Data Transfer Cost ($)
M6g.medium	2950,74	2918.84	137867,12	298,59
M6g.Large	4501,42	4481,36	137962,28	497,65
M6g.Xlarge	49465,79	49405,39	137677,42	995,31
M5.2XLarge	1275803,03	1276385,26	137762,59	3483,52
M5.4XLarge	3599233,17	3600108,32	137634,08	6234,08
M6gd.8XLarge	5282197,57	5322005,63	137742,44	8914,56
M6gd.12XLarge	7432190,15	7473084,39	137624,85	11566,42
M6g.metal	48005803,13	47769425,91	136059,33	14250,25
M5d.metal	84937790,73	85306107,68	134039,80	5810,42
M6i.metal	93010845,72	93028448,77	131046,97	3042,69
M6a.metal	91124687,42	90537061,27	124762,54	1462,37

Moreover, IVCBS's integration of mathematical modeling with trapezoidal membership functions enables precise matching between VM capabilities and user request sizes. This contrasts sharply with the unstructured resource allocation of traditional methods, making IVCBS more adaptable and efficient for large-scale cloud environments. These findings demonstrate that IVCBS can meet growing cloud service demands by optimizing resource use, reducing operational costs, and delivering superior performance across various scenarios, solidifying its potential for future cloud computing advancements.

8.4 Summary

This research advances cloud computing by optimizing VM allocation based on user request sizes, minimizing processing times, costs, and improving global response times. A novel simulation was developed, ensuring workloads align with SLA standards and VM capacities, preventing resource over- or underutilization. Introducing IVCBS, the study enhances data center selection by factoring in VM attributes, job size, and network conditions, outperforming traditional routing methods. Simulations using Cloud Analyst showed significant improvements, suggesting that integrating throttled load balancing could further boost efficiency. This refined approach ensures scalable, cost-effective, and responsive cloud services, laying groundwork for future research on workload classification and performance optimization.

9. A Broker-Driven Approach Integrating Fuzzy Logic for Optimizing Virtual Machine Allocation

introduces a broker-driven framework that integrates fuzzy logic to optimize virtual machine (VM) allocation in cloud environments, advancing resource management practices. Unlike traditional allocation methods that distribute VMs randomly without considering user request sizes, this new approach dynamically assigns VMs based on the size of incoming request packets and CPU utilization, aligning resources with actual workloads for higher efficiency and cost-effectiveness [100][101]. Cloud services' rapid growth has increased the need for sophisticated VM allocation strategies capable of handling heterogeneous and dynamic workloads. Traditional methods typically emphasize physical parameters such as CPU, memory, and storage but often neglect request packet sizes, leading to inefficiencies and bottlenecks [102][102]. Recent research underscores the necessity of adaptive, intelligent allocation mechanisms that consider real-time workload characteristics [103][104]. In this context, the proposed system deploys a centralized broker that analyzes network traffic and redirects requests to appropriately sized VMs using fuzzy logic. This approach enhances VM utilization, reduces latency, and improves overall system responsiveness [105]. The fuzzy logic system manages imprecise inputs, enabling precise decision-making under uncertainty [53]. Tools like Cloud Analyst facilitate the simulation and evaluation of such broker-driven allocation strategies, offering valuable insights into their practical performance [106][107][108]. This research contributes to addressing the limitations of conventional VM allocation by focusing on dynamic optimization guided by request packet size and workload classification. It represents a significant step toward delivering high-quality cloud services while maintaining efficient resource utilization. Despite progress, challenges persist in optimizing VM allocation. Traditional strategies rarely accommodate the fluctuating sizes of request packets, an omission that significantly impacts network performance [109]. In contrast, broker-driven methods dynamically allocate VMs based on packet sizes, enabling real-time optimization and reduced latency. For instance, broker-based models proposed by [29] leverage multi-criteria decision-making to maximize profits and customer satisfaction while minimizing energy consumption in data centers. Further, traditional traffic engineering lacks flexibility for modern cloud demands. Studies like [110] propose fuzzy controllers (Mamdani and Sugeno) to improve VM allocation by accounting for uncertainty, validated through simulation tools like Cloud Analyst [111]. These tools support the modeling of various allocation strategies, helping evaluate their energy consumption and resource management effectiveness. Advanced methods incorporating AI are also emerging. Deep reinforcement learning (DRL) systems like DeepBS improve VM scheduling under uncertainty by learning from previous allocation outcomes, demonstrating significant cost optimizations [112]. Research in mobile terminal cloud computing migration emphasizes efficient data access and minimal latency, leveraging machine learning for dynamic resource allocation [113]. Similarly, IMARM uses a multi-agent reinforcement learning framework to dynamically allocate resources, achieving superior energy efficiency and fault tolerance [114]. Moreover, the field is expanding into cloud robotics and multi-agent systems, exploring challenges such as data transmission delays and heterogeneous energy consumption [30]. AI-driven resource management, including predictive analytics and genetic algorithms, is increasingly employed for intelligent workload management and predictive maintenance [115]. Energy efficiency remains a core concern, with hybrid heuristic algorithms showing significant reductions in resource consumption [116]. Overall, the proposed broker-driven, fuzzy logic-based VM allocation method offers significant potential for enhancing cloud performance, aligning with trends toward intelligent, adaptive resource management. It stands as a robust solution to contemporary challenges in VM allocation, bridging gaps left by traditional strategies and contributing to the ongoing evolution of efficient, scalable cloud computing.

9.1 the proposed system

The proposed methodology for optimizing virtual machine (VM) allocation in cloud computing leverages a broker-driven approach enhanced by fuzzy logic, enabling dynamic resource distribution based on the size of incoming request packets. This technique is intended to improve VM efficiency, reduce latency, and boost overall system performance. It integrates several key components: broker architecture, fuzzy logic modeling, Cloud Analyst simulation, and defined evaluation metrics. Table 9.1 provides workload sizes alongside specifications for Google Cloud’s T2D standard machine types, using pricing data from the Google Compute Engine. The system capitalizes on real-time data to smartly allocate VMs, demonstrating adaptability in adjusting resources in response to network and workload changes.

Table 9.1 workload size machine series specifications.

Workload Size	Machine type Series	VCPU	RAM (GB)	Storage (GB)	BW (GBPS)	Price per hour ($)
Small (<1 GB)	t2d-Standard-1	1	4	2	2	0.054427
Medium (1-10 GB)	t2d-Standard-2	2	8	10	4	0.108854
Large (10-100 GB)	t2d-Standard-4	4	16	16	8	0.217708
Very Large (>100 GB)	t2d-Standard-8	8	32	32	10	0.435416
Massive (Big Data Processing)	t2d-Standard-16	16	64	100	14	0.870832

The broker architecture integrates traffic monitoring, data analysis, and intelligent traffic routing. It continuously observes packet sizes and related metrics, analyzes this data in real time to identify patterns, and directs traffic to appropriate VMs to ensure optimal resource allocation [117]. To address uncertainty and variability in cloud environments, the broker incorporates fuzzy logic [118]. The fuzzy logic model relies on two primary inputs—workload request packet size and CPU utilization—and produces one output that categorizes VM types. For example, packet sizes range from Small (0–5 MB) to Massive (150–250 MB), while CPU utilization spans from Poor (10–40%) to Excellent (85–100%). The output then classifies VMs from Simple to High-Performance levels [119]. These fuzzy rules drive optimal VM allocation, with Table 9.2 illustrating the proposed system and Table 9.3 depicting the traditional approach. The methodology is simulated and evaluated using Cloud Analyst, which enables detailed modeling of cloud scenarios and testing of VM allocation strategies [120]. Cloud Analyst's environment modeling includes configuring data centers with individual VMs and simulating workloads across five different scenarios.

Table 9.2 Proposed Allocate VM according to request size.

Request Packet Size	User Bases	VM	Load balance Algorithm	Broker policy	Price per hour ($)
(<1 GB)	[UB1 -UB10]	t2d-Standard-1	Throttling algorithm.	Optimize response time.	0.054427
(1-10 GB)	[UB1 -UB10]	t2d-Standard-2	Throttling algorithm.	Optimize response time.	0.108854
(10-100 GB)	[UB1 -UB10]	t2d-Standard-4	Throttling algorithm.	Optimize response time.	0.217708
(>100<150 GB)	[UB1 -UB10]	t2d-Standard-8	Throttling algorithm.	Optimize response time.	0.435416
(>150 GB)	[UB1 -UB10]	t2d-Standard-16	Throttling algorithm.	Optimize response time.	0.870832

Table 9.3 Traditional Allocate VM according to request size.

Scenario number	User Bases	Request Packet Size (GB)	Machine type Series
1	[UB1 UB10]	[0.5 200]	t2d-Standard-1
2	[UB1 UB10]	[0.5 200]	t2d-Standard-1
3	[UB1 UB10]	[0.5 200]	t2d-Standard-4
4	[UB1 UB10]	[0.5 200]	t2d-Standard-8
5	[UB1 UB10]	[0.5 200]	t2d-Standard-16

Throttling algorithms further ensure resource fairness and avoid system overloads by regulating CPU, bandwidth, and memory usage [121]. Additionally, a broker policy focused on response time management dynamically allocates resources to minimize latency, ensuring critical tasks are prioritized and enhancing system efficiency and user satisfaction [122]. The broker's logic, including traffic monitoring and fuzzy logic-based decision-making, is implemented within Cloud Analyst to dynamically adjust VM allocation based on real-time operational data. This integrated broker-driven, fuzzy logic-based methodology marks a significant step forward in intelligent cloud resource management, addressing traditional limitations by ensuring precise, adaptive VM allocation tailored to dynamic workloads and diverse operational conditions.

9.2 Simulation and Evaluation of Results

This study presents a thorough simulation-based evaluation of the proposed Intelligent Validation Cloud Broker System (IVCBS), carried out using the Cloud Analyst tool [123]. The experiments involved five scenarios, each deploying ten distinct user bases as previously described. Scenarios escalated in workload size, starting from 500 million bytes processed on t2d-Standard-1, and scaling up to 200 billion bytes on t2d-Standard-16. The same workload configurations and user base behaviors were replicated in simulations for both the proposed and traditional methods to enable a direct performance comparison. However, the traditional approach differed significantly in how it distributed and processed workloads, each simulation tested varying request packet sizes and VM resource requirements to examine the robustness and adaptability of the broker-driven method. The experiments modeled realistic cloud environments, incorporating the fluctuating nature of workloads to evaluate system responsiveness under dynamic conditions. The proposed broker-driven system uniquely integrates fuzzy logic, using workload packet size and CPU utilization as input parameters to classify VMs into five workload intensity levels. Performance metrics collected included overall response time, data center processing time, request serving time, total VM costs, and data transfer costs. Comparative analysis between the traditional method (Table 9.4) and the proposed method (Table 9.5) revealed significant improvements. The broker-driven approach achieved up to a 68% reduction in response time and approximately 20% reductions in processing and serving times. Moreover, it substantially lowered costs, especially in VM provisioning and data transfers. A key innovation of this research is incorporating packet size classification into VM allocation—an aspect often overlooked in traditional strategies that focus primarily on resource scalability without accounting for workload heterogeneity. By factoring in packet size alongside real-time CPU utilization, the proposed approach enables more precise and intelligent resource allocation. Fuzzy logic contributes critical adaptability, allowing the system to handle uncertainty and align resource allocation with dynamic workload patterns more effectively than static methods.

Table 9.4 Summary of the results of the traditional method.

Scenario	Overall response time Avg(ms)	Datacenter processing time Avg(ms)	Datacenter request serving times Avg(ms)	Total data transfer cost ($)
1	571309,86	58,06	58,06	33959999,08
2	548272,30	59,31	59,31	30557098,39
3	565510,88	60,39	60,386	33791313,17
4	558790,62	58,03	58,026	33726768,49

5	574401,10	59,35	59,348	32435417,18

Table 9.5 Summary of the results of the proposed Method.

Scenario Number	Overall response time Avg(ms)	Datacenter processing time Avg(ms)	Datacenter request serving times Avg(ms)	Total data transfer cost ($)
1	333748,21	56,41	56,141	4186420,44
2	278151,12	49,88	49,875	6354904,17
3	183111	44,30	44,297	9916305,54
4	0	39,32	39,323	4909515,38
5	0	40,26	40,264	4531860,35

9.3 Summary

Details a comprehensive simulation and evaluation of the proposed broker driven VM allocation method, executed using the Cloud Analyst tool. Five scenarios tested varying workload sizes, scaling from 500 million to 200 billion bytes, matched to different Google t2d-Standard VM types, ensuring alignment between request size and resource capacity. While the traditional VM allocation method used identical parameters for user bases and peak hours, it diverged fundamentally in how workloads were distributed and managed, lacking the adaptive intelligence of the proposed approach (as shown in Table 9.2). The simulations assessed robustness and adaptability under realistic, dynamic cloud workloads, comparing the new fuzzy logic-based method against traditional strategies. The broker-driven system utilized fuzzy logic, incorporating workload packet size and CPU utilization as inputs to classify and allocate VMs across five workload intensity levels. Key performance metrics were analyzed, including overall response time, data center processing time, serving time, total VM costs, and data transfer expenses. The broker-driven approach achieved significant gains: reducing response times by up to 68%, cutting processing and serving times by about 20%, and lowering costs, particularly in data transfer and VM provisioning, as highlighted in Tables 9.4 and 9.5. The novelty of this research lies in integrating fuzzy logic with packet size classification—a dimension typically neglected in conventional VM allocation, which focuses mainly on scalability without considering packet heterogeneity. By factoring in packet size and real-time CPU load, the proposed model delivers finer, intelligent resource allocation. Fuzzy logic enables the system to manage uncertainty and dynamically align resources with varying demands, outperforming rigid rule-based methods. This proactive, precise allocation helps prevent bottlenecks, enhances energy efficiency, and improves system responsiveness. Practically, the methodology offers scalable, cost-effective, and energy-aware cloud resource management. It ensures equitable VM distribution, optimizes operational costs, and significantly improves service reliability and performance in heterogeneous, high-demand cloud environments, advancing intelligent cloud infrastructure design.

10. Reliable and Cost-Effective Fuzzy-based Cloud Broker

This study proposes a reliable and cost-effective cloud broker architecture leveraging fuzzy logic to enhance decision-making in cloud environments, addressing the growing complexity faced by users selecting suitable cloud services among numerous Cloud Service Providers (CSPs) [124]. The broker functions as an intelligent intermediary, balancing user needs with CSP interests, while analyzing user mobility scenarios—including stationary and mobile users—and evaluating the effects of service migration on performance and costs. This dynamic management approach improves service reliability, resource efficiency, and cost-effectiveness compared to traditional methods [125]. Cloud brokerage has become critical for efficiently provisioning resources and managing the complexity of modern distributed systems such as URLLC, eMBB, and mMTC applications promised by 5G and beyond networks [126][127]. Existing cloud brokers adopt various strategies—customer-centric, profit-centric, or balanced approaches—yet face challenges in scalability, decision speed, and usability. Techniques used include game theory [32], reinforcement learning, weighted algorithms, ontology-based methods, and multi-criteria decision-making approaches such as AHP combined with TOPSIS [128]. However, these methods often struggle with increased complexity, high computational costs, or the burden of requiring precise user input definitions, which can deter nonprofessional users [33]. Fuzzy logic-based brokers have demonstrated significant potential. However, they encounter challenges when managing many input parameters, resulting in an exponential increase in rule sets and complexity for the inference engine. Data collection can also become problematic due to privacy concerns and CSPs' reluctance to disclose sensitive parameters [129]. The proposed study mitigates these limitations by integrating fuzzy logic with a modified TOPSIS technique, focusing on two practical, easily measurable parameters per fuzzy system. This design significantly reduces inference complexity, enabling scalable, real-time decision-making in heterogeneous cloud environments. The methodology was tested across multiple data centers of major providers (AWS, Google Cloud, Azure), evaluating different VM types, and effectively ranked services and users to optimize resource matching and ensure both user satisfaction and CSP operational efficiency.

10.1 System Design

The proposed fuzzy-based cloud broker system is designed to simplify cloud service selection for both novice and expert users, addressing usability issues prevalent in commercial brokers [130]. The system architecture (Figure 10.1) centers around three key phases: service discovery, ranking, and matching. **Service Discovery** enables users to specify requirements such as service type, budget, and quality. The broker then identifies relevant cloud service instances from major CSPs like AWS, Google Cloud, and Azure, distributed globally across the USA, Europe, and Southeast Asia. **Ranking** employs two fuzzy logic systems: one ranks VMs based on CPU availability and cost, and the other ranks users considering budget and task size. VM rankings use trapezoidal and triangular membership functions (Figures 10.2), processed via an IF-THEN inference engine with rules such as: *IF VM CPU is Low AND Cost is Low THEN Rank = Silver*. User rankings rely on fuzzy sets for budget and task length (Figures 10.3), also defuzzified via the centroid-of-gravity (COG) method. This approach reduces complexity compared to traditional methods while ensuring accurate classifications [131]. **Matching** then pairs users to VMs based on ranks: Gold users match Gold VMs for high performance, Silver for balanced service, and Bronze for economical options. This structured matching ensures optimal compatibility, balanced load distribution, and user satisfaction, streamlining decision-making for real-time allocations.

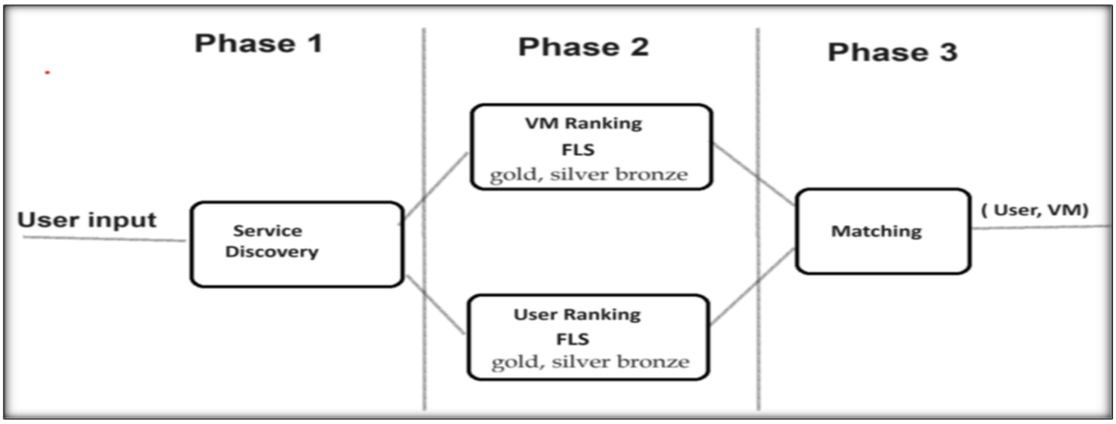

Figure 10.1 Proposed System Architecture.

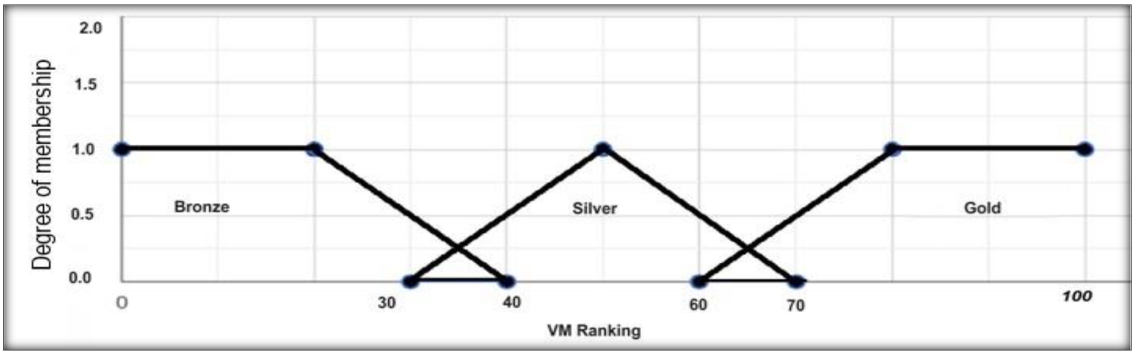

Figure 10.2 VM's ranking membership function.

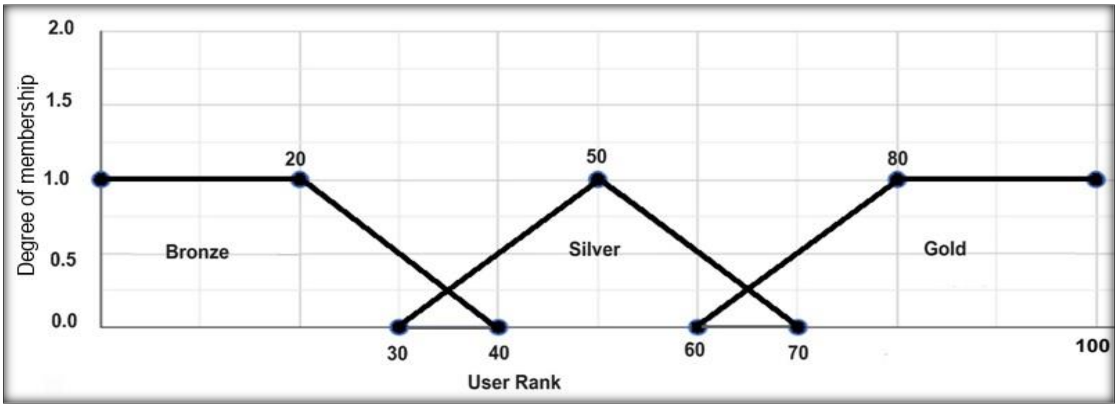

Figure 10.3 User rank membership function.

To validate the design, simulations were executed using Edge CloudSim under the Mobile Edge Computing (MEC) paradigm, chosen for its sensitivity to delays [132]. The simulation models realistic network conditions using WLAN and WAN delays, with data centers connected via WAN or MAN networks. Various VM types were tested—including general-purpose and compute-optimized instances (Table 10.1)—and four delay-sensitive services were evaluated (Table 10.2). User workloads were modeled by task length in millions of instructions, allowing the broker to allocate resources dynamically and optimize Quality of Service (QoS). This architecture demonstrates a scalable, intelligent framework for cloud brokerage, effectively bridging user requirements and CSP offerings while managing uncertainty through fuzzy logic. The proposed scenario employs Edge CloudSim [133] to implement a fuzzy-based cloud broker within the MEC paradigm, targeting the latency sensitivity of edge services. The simulation integrates diverse data centers from AWS, Google Cloud, and Azure, distributed across the USA, Western Europe, and Southeast Asia, interconnected via WAN and MAN networks. Various VM types—general-purpose, compute-optimized, memory-optimized, and accelerator-optimized—are selected based on official CSP specifications. Users' service selections depend on budgets and task sizes, influencing VM allocation. Four delay-intolerant services are modeled, considering traffic characteristics and delay sensitivity. Network delays are simulated using empirical data from WLAN and WAN measurements.

Table 10.1 Official Application Specifications from the Three Cloud Providers' Websites.

Name	CSP	Type	Number of vcpu	Memory
T2A	GC	General purpose	2	4
E2	GC	Cost optimized	2	1
M1	GC	Memory optimized	40	961
C2	GC	Compute optimized	4	6
A2	GC	Accelerator optimized	12	85
t2. small	AWS	General purpose	1	2
i4i.large	AWS	Storage optimized	2	16
r7a.medium	AWS	Memory optimized	1	8
r7a.large	AWS	Memory optimized	2	16
c7a.medium	AWS	Compute optimized	1	2
c7a.large	AWS	Compute optimized	2	4
p3.2xlarge	AWS	Accelerator optimized	8	61
hpc7g.4xlarge	AWS	HPC optimized	16	128
B2ls v2	AZURE	General purpose	2	4
F2s v2	AZURE	Compute optimized	2	4
E2as v5	AZURE	Memory optimized	2	16
L8as v3	AZURE	Storage optimized	8	64
NC6	AZURE	GPU optimized	6	56

H8	AZURE	High performance compute	8	56

Table 10.2 Types and Specifications of Delay-Intolerant Services in the Simulation Setup.

Type	Average of upload data	Average of download data	Task Length	Delay sensitivity
Health App	1500	25	9000	0.7
Augmented Reality	20	1250	3000	0.9
Heavy Computing	2500	200	45000	0.1
Infotainment	25	1000	15000	0.3

10.2 Results Analysis and Discussion

This study evaluates the proposed fuzzy-based broker system against two alternatives: a random service selection approach and a least-loaded (LL) broker that allocates users to the VM with the highest available processing power. The comparison focuses on service delay and user costs across three key scenarios. In the **first scenario** (stationary users under reserved VM policies), the fuzzy broker consistently outperformed both LL and random approaches, achieving lower average service delays and more stable performance as user numbers grew. LL performed poorly since users remained tied to the same VM throughout the simulation, negating the benefits of load balancing. Cost analysis revealed that the fuzzy system kept user expenses constant while the LL and random methods drove costs higher as user numbers increased due to imbalanced allocation to expensive VMs. This stability is crucial for maintaining SLA compliance and user satisfaction. Figures 10.4 and 10.5 present the results of the scenario.

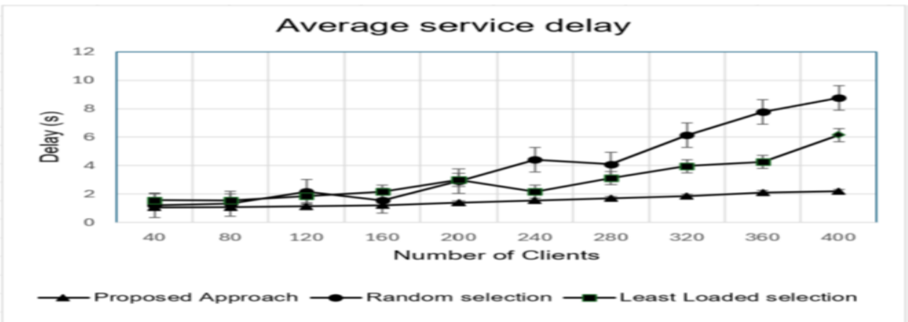

Figure 10.4 Average service delay for immobile users.

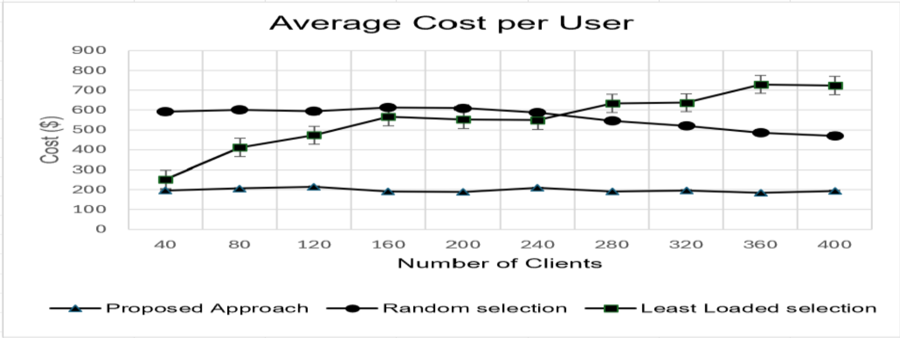

Figure 10.5 The average of monthly client payment.

The **second scenario** simulated **nomadic users** under a reserved instance policy, where users remained linked to the same data center despite relocating geographically. All approaches suffered from increased delays due to rising communication distances, but the fuzzy broker showed lower delay variance than LL and random selection, an important advantage for SLA adherence, though it could not significantly reduce absolute delay times under this fixed-association constraint. Figures 10.6 present the results of the scenario.

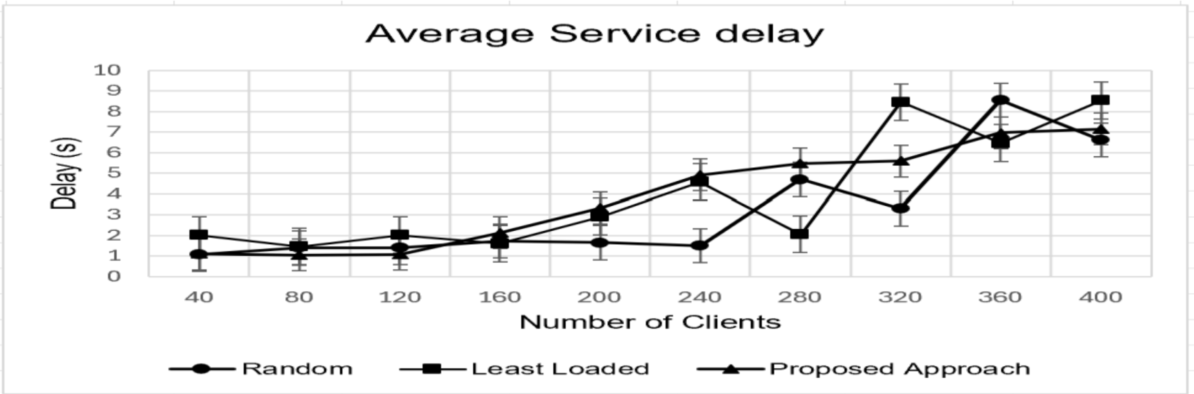

Figure 10.6 Average service delay for mobile users.

The **third scenario** introduced **cross-cloud service migration** under a pay-as-you-go (PAYG) policy. Here, the fuzzy broker and LL method achieved similar service delays, yet the fuzzy system offered more consistent service quality while maintaining predictable costs for users regardless of growing demand. This advantage underscores the fuzzy broker's capacity to manage dynamic user mobility and resource shifts while respecting cost constraints and SLA requirements. Figures 10.7 and 10.8 present the results of the scenario.

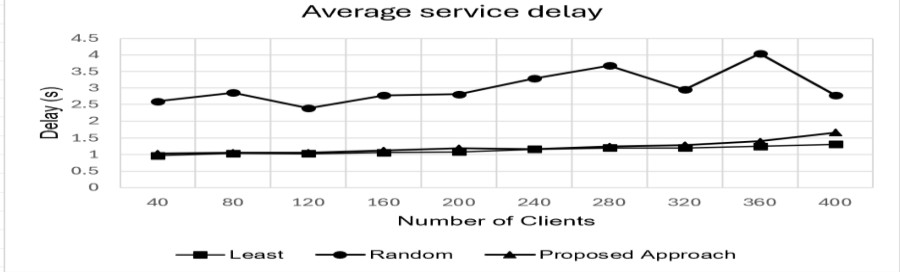

Figure 10.7 Avg service delay with mobile users and service migration.

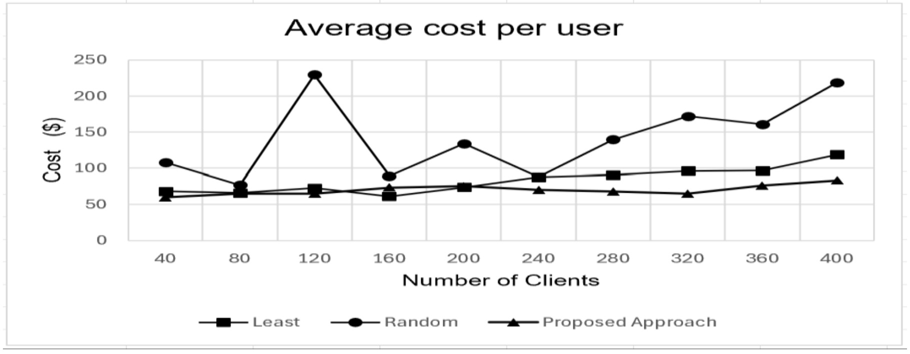

Figure 10.8 Average monthly payment in case of service migration.

Practical evaluation in this study confirmed that the fuzzy-based broker is feasible for real-world deployment but faces computational challenges as user numbers scale. Unlike random and LL methods, fuzzy logic requires intensive processing for large user bases. To mitigate this, strategies like user clustering, flow-based ranking, and user profiling can be applied. For example, users running identical applications, such as video gamers or enterprise teams, could be grouped and collectively ranked, reducing computation. Additionally, high-priority services (e.g., health applications) can be pre-assigned to high-rank classes (Gold) to streamline decision-making. Overall, the fuzzy-based broker demonstrates superior performance, cost efficiency, and SLA stability compared to conventional methods, positioning it as a promising solution for future intelligent, dynamic cloud resource allocation.

10.3 Summary

This study presents a novel fuzzy logic-based cloud broker designed to balance the needs of clients and cloud service providers (CSPs). The proposed system has been evaluated across diverse scenarios, demonstrating its feasibility and effectiveness. For future work, the authors plan to enhance the broker by integrating additional decision parameters, including application delay sensitivity and client mobility profiles, recognizing that network delays significantly impact mobile users, particularly where service migration is not supported. To address this, new mechanisms will be developed to minimize mobility-related performance issues. As discussed in earlier chapters, employing a third-party broker is crucial in modern cloud environments. A cloud broker acts as an intermediary between customers and CSPs, facilitating efficient, cost-effective service provision, resource management, and load balancing across multiple clouds or instances within the same cloud. Cloud broking is a rapidly expanding field, driven by widespread cloud adoption and the increasing complexity of multi-cloud and hybrid environments. Cloud Service Brokers (CSBs) are essential for optimizing costs, managing resources, and integrating advanced technologies like AI, big data, and IoT. Future trends in cloud broking are expected to emphasize deeper AI integration, improved security, growth into new markets, and greater automation, highlighting its potential as a dynamic sector for innovation and development.

11. Theses

Cloud computing underpins modern IT, offering scalable resources via SLAs that define performance guarantees. However, challenges remain, including compliance issues, vendor lock-in, variable QoS, and high energy consumption from expanding data centers. Geographical dispersion increases RTT variability and complicates latency management, while

CSPs often lack precise network performance metrics. Ensuring reliable, efficient cloud services requires intelligent management, advanced resource allocation, and predictive modeling. Addressing these gaps, this doctoral research contributes three innovative systems leveraging fuzzy logic and decision-making models, enhancing SLA optimization, resource management, and sustainability in cloud computing to meet evolving IT demands.

Thesis I: Intelligent SLA Guarantee Model for Cloud Computing

I have developed an **Intelligent SLA Guarantee Model for Cloud Computing**, employing fuzzy logic for the estimation of (RTT) and the classification of (SLAs). This model transforms complex technical measurements into linguistically interpretable terms, enabling clearer SLA assessments and more user-friendly decision-making processes. The results of this research have been published in the following conference proceedings:

- Sekhi, I. (2023). *Estimating Cloud Computing Round-Trip Time (RTT) Using Fuzzy Logic for Inter-Region Distances*. International Journal on Cybernetics & Informatics (IJCI), 12(12), 95.
- Sekhi, I. (2023). *Selecting the SLA Guarantee by Evaluating the QoS Availability*. Multidiszciplináris Tudományok: A Miskolci Egyetem Közleménye, 13(4), 80–102. https://doi.org/10.35925/j.multi.2023.4.8

Thesis II: Intelligent Validation Cloud Broker System (IVCBS)

I have created the (IVCBS), a fuzzy logic-based framework designed to optimize (VM) allocation and improve cloud computing efficiency. The system dynamically adjusts VM distribution based on the analysis of incoming request packet sizes, enhancing resource utilization, reducing latency, and maintaining consistent service quality.

The outcomes of this research have been documented in the following journals:

- Sekhi, I., & Nehéz, K. (2024). *Intelligent SLA Selection Through the Validation Cloud Broker System*. IEEE Access. DOI: 10.1109/ACCESS.2024.3439617
- Sekhi, I. (Accepted). *Efficient Broker-Driven Request Packet Size*. International Journal on Informatics Visualization.

Additionally, related foundational concepts and fuzzy logic optimization techniques were published in:

- Sekhi, I., Kovács, S., & Nehéz, K. (2025). *Enhancing Decision-Making in Uncertain Domains through Optimized Fuzzy Logic Systems*. Periodica Polytechnica Electrical Engineering and Computer Science, 69(1), 63–78. https://doi.org/10.3311/PPee.38729

Thesis III: Intelligent Cloud Brokerage System

I have designed an **Intelligent Cloud Brokerage System** that combines fuzzy logic with the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to optimize cloud service selection and resource management across multiple (CSPs). This intelligent brokerage system serves as an intermediary, aligning user requirements with provider capabilities to improve service quality, cost efficiency, and operational performance.

The findings related to this research are published in:

- Sekhi, I. R., Abdah, H., & Nehéz, K. (2025). *Reliable and Cost-Effective Fuzzy-Based Cloud Broker*. International Journal of Networked and Distributed Computing, 13(1), 1–9. https://doi.org/10.1007/s44227-024-00052-x

These three theses collectively address critical challenges in cloud computing, contributing innovative solutions for enhancing performance, reducing latency, and improving the efficiency of resource management. The integration of fuzzy logic and advanced decision-making techniques in my research provides new pathways for achieving scalable, reliable, and cost-effective cloud services.

12. Future Research Direction

- Future research should focus on integrating IoT, edge computing, and 5G to enhance cloud computing scalability and interoperability. Real-world testing is crucial to evaluate performance, adaptability, and SLA management. Incorporating ML and fuzzy logic can optimize SLA classification and QoS adjustments, improving efficiency and reliability. Additionally, adaptive traffic management should be explored to enhance QoS, resource allocation, and fault recovery. Further research on SLA prioritization will optimize cloud resource utilization and user satisfaction. These advancements will contribute to intelligent, adaptive, and efficient cloud brokerage systems, ensuring better service selection and resource optimization in dynamic cloud environments.
- Enhance cross-cloud compatibility through standardized integration methods, ensuring seamless workload distribution across heterogeneous platforms for individual users and enterprises. This will also improve energy efficiency, reducing data centers' carbon footprint while maintaining high performance. Leveraging ML-driven workload distribution enables real-time optimization, dynamically adapting to service demands and enhancing resource efficiency. Addressing security and compliance challenges is crucial to mitigating vulnerabilities, improving data privacy, and maintaining regulatory standards in multi-cloud environments. Additionally, context-aware decision-making in cloud brokerage systems should incorporate application delay sensitivity and client mobility profiles. Developing adaptive mechanisms to adjust resource allocation dynamically will help mitigate network delay, ensuring seamless service quality, minimal latency, and optimal performance in mobile cloud environments.

References

[1] Aravinth, S. S., Krishnan, A. S. R., Ranganathan, R., Sasikala, M., Kumar, M. S., & Thiyagarajan, R. (2024). Cloud Computing—Everything as a Cloud Service in Industry 4.0. In Digital Transformation: Industry 4.0 to Society 5.0 (pp. 103-121). Singapore: Springer Nature Singapore.

[2] Qazi, F., Kwak, D., Khan, F. G., Ali, F., & Khan, S. U. (2024). Service Level Agreement in cloud computing: Taxonomy, prospects, and challenges. Internet of Things, 101126.

[3] Sissodia, R., Rauthan, M. S., & Barthwal, V. (2024). Service Level Agreements (SLAs) and Their Role in Establishing Trust. In Analyzing and Mitigating Security Risks in Cloud Computing (pp. 182-193). IGI Global.

[4] Fernandes, S. (2017). Performance evaluation for network services, systems and protocols (pp. 1-175). Heidelberg: Springer.

[5] Bose, R., Sengupta, S., & Roy, S. (2023). Interpreting SLA and related nomenclature in terms of Cloud Computing: a layered approach to understanding service level agreements in the context of cloud computing. Lambert Academic Publishing.

[6] Shan, L., Sun, L., & Rezaeipanah, A. (2024). Towards a novel service broker policy for choosing the appropriate data center in cloud environments. Computer Communications, 107939.

[7] Mendel, J., & Wu, D. (2010). Perceptual computing: aiding people in making subjective judgments. John Wiley & Sons.

[8] Tallón-Ballesteros, A. J., & Beltrán-Barba, R. (Eds.). (2023). Fuzzy Systems and Data Mining IX: Proceedings of FSDM 2023 (Vol. 378). IOS Press.

[9] Wickremasinghe, B., Calheiros, R. N., & Buyya, R. (2010, April). Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In 2010 24th IEEE international conference on advanced information networking and applications (pp. 446-452). IEEE.

[10] Sonmez, C., Ozgovde, A., Ersoy, C.: Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies 29(11), 3493 (2018).

[11] Haitjema, M. A. (2013). Delivering Consistent Network Performance in Multi-tenant Data Centers. Washington University in St. Louis.

[12] McCabe, J. D. (2010). Network analysis, architecture, and design. Elsevier.

[13] Hopgood, A. A. (2021). Intelligent systems for engineers and scientists: a practical guide to artificial intelligence. CRC press

[14] Alhamad, M., Dillon, T., & Chang, E. (2010, April). Conceptual SLA framework for cloud computing. In 4th IEEE international conference on digital ecosystems and technologies (pp. 606-610). IEEE. https://doi:10.1109/DEST.2010.5610586

[15] Qiu, M. M., Zhou, Y., & Wang, C. (2013, June). Systematic analysis of public cloud service level agreements and related business values. In 2013 IEEE International Conference on Services Computing (pp. 729-736). IEEE. https://doi: 10.1109/SCC.2013.24

[16] Brunnström, K., Beker, S. A., De Moor, K., Dooms, A., Egger, S., Garcia, M. N., ... & Zgank, A. (2013). Qualinet white paper on definitions of quality of experience.

[17] Godhrawala, H., & Sridaran, R. (2023). Apriori Algorithm Based Approach for Improving QoS and SLA Guarantee in IaaS Clouds Using Pattern-Based Service-Oriented Architecture. SN Computer Science, 4(5), 700.

[18] Akbari-Moghanjoughi, A., Amazonas, J. R. D. A., Santos-Boada, G., & Solé-Pareta, J. (2023). Service level agreements for communication networks: A survey. arXiv preprint arXiv:2309.07272.

[19] Saqib, M., Elbiaze, H., & Glitho, R. (2024). Adaptive In-Network Traffic Classifier: Bridging the Gap for Improved QoS by Minimizing Misclassification. IEEE Open Journal of the Communications Society.

[20] Bauer, E., & Adams, R. (2012). Reliability and availability of cloud computing. John Wiley & Sons.

[21] Gupta, K., Kumar, P., Upadhyaya, S., Poriye, M., Aggarwal, S. "Fuzzy logic and machine learning integration: Enhancing healthcare decision-making," International Journal of Computer Information Systems and Industrial Management Applications, 16(3), pp. 20–20, 2024.

[22] Wang, L. X., Mendel, J. M. "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," IEEE Transactions on Neural Networks, 3(5), pp. 807–814, 1992. https://doi.org/10.1109/72.159070
[23] Perera, L. P., Carvalho, J. P., Soares, C. G. "Solutions to the failures and limitations of Mamdani fuzzy inference in ship navigation," IEEE Transactions on Vehicular Technology, 63(4), pp. 1539–1554, 2013. https://doi.org/10.1109/TVT.2013.2288306
[24] Zadeh, L. A. "Fuzzy logic = computing with words," IEEE Transactions on Fuzzy Systems, 4(2), pp. 103–111, 1996. https://doi.org/10.1109/91.493904
[25] Gen, M., Lin, L. "Genetic algorithms and their applications," In: Pham, H. (ed.) Springer Handbook of Engineering Statistics, Springer Handbooks, Springer, London, 2023, pp. 635–674. ISBN 978-1-4471-7502-5. https://doi.org/10.1007/978-1-4471-7503-2_33
[26] Singh, H., Tyagi, S., & Kumar, P. (2021). Comparative analysis of various simulation tools used in a cloud environment for task-resource mapping. In Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences: PCCDS 2020 (pp. 419-430). Springer Singapore.
[27] Chauhan, S. S., Pilli, E. S., Joshi, R. C., Singh, G., & Govil, M. C., Brokering in interconnected cloud computing environments: A survey. Journal of Parallel and Distributed Computing, 133, 193-209. (2019)
[28] Chafai, Z., Nacer, H., Lekadir, O., Gharbi, N., & Ouchaou, L., A performance evaluation model for users' satisfaction in federated clouds. Cluster Computing, 1-22. (2024)
[29] Kansal, N. J., & Chana, I. (2012). Cloud load balancing techniques: A step towards green computing. International Journal of Computer Science Issues, 9(1), 238-246.
[30] Li, X., Pan, L., & Liu, S. (2023). A DRL-based online VM scheduler for cost optimization in cloud brokers. World Wide Web, 26(5), 2399-2425.
[31] Afrin, M., Jin, J., Rahman, A., Rahman, A., Wan, J., & Hossain, E. (2021). Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(2), 842-870
[32] Valarmathi, R., Sheela, T.: Differed service broker scheduling for data centres in cloud environment. Computer Communications 146, 186–191 (2019)
[33] Vimercati, S.D.C., Foresti, S., Livraga, G., Piuri, V., Samarati, P.: A fuzzy-based brokering service for cloud plan selection. IEEE Systems Journal 13(4), 4101–4109 (2019)
[34] Ikram, M. A., & Hussain, F. K. (2018). Software as a service (SaaS) service selection based on measuring the shortest distance to the consumer's preferences. In Advances in Internet, Data & Web Technologies: The 6th International Conference on Emerging Internet, Data & Web Technologies (EIDWT-2018) (pp. 403-415). Springer International Publishing
[35] Kavis, M. (2014). Architecting the cloud: design decisions for cloud computing service models (SaaS, PaaS, and IaaS). John Wiley & Sons, Inc., Hoboken, New Jersey.
[36] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. NIST special publication, 500(2011), 1-28.
[37] Wang, L., Ranjan, R., Chen, J., & Benatallah, B. (Eds.). (2011). Cloud computing: methodology, systems, and applications. CRC press.
[38] Rountree, D., & Castrillo, I. (2013). The basics of cloud computing: Understanding the fundamentals of cloud computing in theory and practice. Newnes.
[39] Cloud, H. (2011). The nist definition of cloud computing. National institute of science and technology, special publication, 800(2011), 145.
[40] Chandrasekaran, K. (2014). Essentials of cloud computing. CrC Press.
[41] Kingsley, M. S. (2023). Cloud Technologies and Services: Theoretical Concepts and Practical Applications. Springer Nature.
[42] Wagdy, M., Babulak, E., & Al-Dabass, D. (2021). Network function virtualization over cloud-cloud computing as business continuity solution. Intechopen, Published: July 14th.
[43] L'Esteve, R. C. (2023). The cloud leader's handbook: strategically innovate, transform, and scale organizations.
[44] Hiran, K. K., Doshi, R., Fagbola, T., & Mahrishi, M. (2019). Cloud computing: master the concepts, architecture and applications with real-world examples and case studies. Bpb Publications.
[45] Amankwah, R., Asianoa, R., & Birago, B. Virtualization and Cloud Computing. International Journal of Computer Applications, 975, 8887.
[46] Kocaleva, M., Zlatanovska, B., Karamazova Gelova, E., & Zlatev, Z. (2024). Cloud computing and virtualization: can cloud computing exist separately from virtualization?.

[47] Furht, B., & Escalante, A. (2010). Handbook of cloud computing (Vol. 3). New York: springer.
[48] Zerwas, J., Györgyi, C., Blenk, A., Schmid, S., & Avin, C. (2023). Duo: A high-throughput reconfigurable datacenter network using local routing and control. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 7(1), 1-25.
[49] Borra, P. (2024). Comparison and Analysis of Leading Cloud Service Providers (AWS, Azure and GCP). International Journal of Advanced Research in Engineering and Technology (IJARET), 15(3).
[50] S. Mathew and J. Varia, "Overview of amazon web services," Amazon Whitepapers, vol. 105, no.1, p. 22, 2014.
[51] Ludwig, H. (2003, December). Web services QoS: external SLAs and internal policies or: how do we deliver what we promise?. In Fourth International Conference on Web Information Systems Engineering Workshops, 2003. Proceedings. (pp. 115-120). IEEE.
[52] Kadaskar, H. R., & Kamthe, V. R. (2024). An overview of AWS. International Journal of Scientific Research in Modern Science and Technology, 3(7), 22-30.
[53] Marinescu, D. C. (2022). Cloud computing: theory and practice. Morgan Kaufmann.
[54] Selimi, M., Freitag, F., Cerdà-Alabern, L., & Veiga, L. (2016). Performance evaluation of a distributed storage service in community network clouds. Concurrency and Computation: Practice and Experience, 28(11), 3131-3148.
[55] Yadav, R. K., Chattopadhyay, S., Jaidka, P., & Upadhyay, P. (2024, March). Performance Analysis of Cloud-Assisted Resource Allocation Algorithms in 6G Networks. In 2024 2nd International Conference on Disruptive Technologies (ICDT) (pp. 1038-1043). IEEE.
[56] Arslan, S., Li, Y., Kumar, G., & Dukkipati, N. (2023). Bolt:{Sub-RTT} Congestion Control for {Ultra-Low} Latency. In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23) (pp. 219-236).
[57] Buyya, R., Yeo, C. S., & Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In 2008 10th IEEE international conference on high performance computing and communications (pp. 5-13). Ieee.
[58] Le Thi, H. A., Le, H. M., Dinh, T. P., & Nguyen, N. T. (2015). Advanced computational methods for knowledge engineering. Cham: Springer International Publishing.
[59] Dang, T. K., Mohan, N., Corneo, L., Zavodovski, A., Ott, J., & Kangasharju, J. (2021, November). Cloudy with a chance of short RTTs: analyzing cloud connectivity in the internet. In Proceedings of the 21st ACM Internet Measurement Conference (pp. 62-79).
[60] Huang, H., Wang, Y., Cai, Y., & Wang, H. (2024). A novel approach for energy consumption management in cloud centers based on adaptive fuzzy neural systems. Cluster Computing, 1-24.
[61] Nithya, S., Maithili, K., Kumar, T. S., Nethani, S., Sharath, M. N., Gupta, K. G., & Bhuvaneswari, G. (2024). A fuzzy logic and cross-layered optimization for effective congestion control in wireless sensor networks to improve efficiency and performance. In MATEC Web of Conferences (Vol. 392, p. 01145). EDP Sciences.
[62] Mirkovic, d., armitage, g., & branch, p. (2018). A survey of round-trip time prediction systems. Ieee communications surveys & tutorials, 20(3), 1758-1776.
[63] GeeksforGeeks. (2023, April 13). What is RTT (Round-Trip Time)? GeeksforGeeks. https://www.geeksforgeeks.org/what-is-rttround-trip-time.
[64] Klir, g. J., & yuan, b. (1996). Fuzzy sets and fuzzy logic: theory and applications. Possibility theory versus probab. Theory, 32(2), 207-208.https://doi.10.5860/choice.33-2786.
[65] Pedrycz, w. (1994). Why triangular membership functions? Fuzzy sets and systems, 64(1), 21-30. Https://doi.org/10.1016/0165-0114(94)90003-5
(CCEM) (pp. 1-6). IEEE. https://doi: 10.1109/CCEM.2013.6684439
[66] Alhamad, M., Dillon, T., & Chang, E. (2011). A trust-evaluation metric for cloud applications. International Journal of Machine Learning and Computing, 1(4), 416. https://doi: 10.7763/IJMLC. 2011.V1.62
[67] Xiaoyong, Y., Ying, L., Tong, J., Tiancheng, L., & Zhonghai, W. (2015, July). An analysis on availability commitment and penalty in cloud sla. In 2015 IEEE 39th Annual Computer Software and Applications Conference (Vol. 2, pp. 914-919). IEEE. https://doi:10.1109/COMPSAC.2015.39
[68] Al Moteri, M. A. (2017). Decision Support for Shared Responsibility of Cloud Security Metrics.
[69] Abery, B., Bonner, M., Fossum, P., Koch, T., Montie, J., Nordness, K., ... & Vandercook, T. (1998). The Shared Responsibility Framework of Social Interaction for Collective Investment: Introducing a Model to Enhance School Improvement.

[70] Tran, V. X., & Tsuji, H. (2008, October). QoS based ranking for web services: Fuzzy approaches. In 2008 4th international conference on next generation web services practices (pp. 77-82). IEEE. https://doi: 10.1109/NWeSP.2008.41
[71] Patel, P., Ranabahu, A. H., & Sheth, A. P. (2009). Service level agreement in cloud computing.
[72] Dubois, D., Prade, H. Fuzzy Sets and Systems: Theory and Applications, Academic Press, 1980. ISBN 0-12-222750-6.
[73] Zheng, Y., Xu, Z., Wu, T., et al. "A systematic survey of fuzzy deep learning for uncertain medical data," Artificial Intelligence Review, 57, pp. 230, 2024. https://doi.org/10.1007/s10462-024-10871-7
[74] Guerrero Granados, B., Quintero M, C. G., Núñez, C. V. "Improved genetic algorithm approach for coordinating decision-making in technological disaster management," Neural Computing and Applications, 36(9), pp. 4503–4521, 2024. https://doi.org/10.1007/s00521-023-09218-0
[75] Tahmasebi, M., Gohari, M., Emami, A. "An autonomous pesticide sprayer robot with a color-based vision system," International Journal of Robotics and Control Systems, 2(1), pp. 115–123, 2022. https://doi.org/10.31763/ijrcs.v2i1.480
[76] Mitiku, T., Manshahia, M. S. "Neuro fuzzy inference approach: A survey," International Journal of Scientific Research in Science, Engineering and Technology, 4(7), pp. 505–519, 2018. Print ISSN: 2395-1990, Online ISSN: 2394-4099.
[77] Zadeh, L. A., Klir, G. J., Yuan, B. "Fuzzy sets, fuzzy logic, and fuzzy systems: Selected papers," World Scientific, 1996. ISBN: 978-981-02-2421-9. https://doi.org/10.1142/2895
[78] Hasan, M. H., Jaafar, J., Hassan, M. F. "Fuzzy C-Means and two clusters' centers method for generating interval type-2 membership function," In: 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 2016, pp. 627–632. https://doi.org/10.1109/ICCOINS.2016.7783288
[79] Takagi, T., Sugeno, M. "Fuzzy identification of systems and its applications to modeling and control," IEEE Transactions on Systems, Man, and Cybernetics, 15(1), pp. 116–132, 1985. https://doi.org/10.1109/TSMC.1985.6313399
[80] Simon, M. K. "Probability distributions involving Gaussian random variables: A handbook for engineers and scientists," Kluwer Academic Publishers, Boston-Dordrecht-London, 2002. ISBN 978-0-387-34657-1.
[81] Miller, S. "Probability and random processes: With applications to signal processing and communications," Academic Press, 2012. ISBN 978-0-12-386981-4.
[82] Elgendi, M. "PPG signal analysis: An introduction using MATLAB®" [e-book], CRC Press, 2020. ISBN 9780429449581. https://doi.org/10.1201/9780429449581
[83] Ruparelia, N. B. (2023). Cloud computing. Mit Press.
[84] Rao, M. N. (2015). Cloud computing. PHI Learning Pvt. Ltd.
[85] Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010, September). The characteristics of cloud computing. In 2010 39th International Conference on Parallel Processing Workshops (pp. 275-279). IEEE.
[86] Dilli, R., Argou, A., Pilla, M., Pernas, A. M., Reiser, R., & Yamin, A. (2018, April). Fuzzy logic and MCDA in IoT resources classification. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (pp. 761-766).
[87] Ghasemi, A., Toroghi Haghighat, A., & Keshavarzi, A. (2023). Enhanced multi-objective virtual machine replacement in cloud data centers: combinations of fuzzy logic with reinforcement learning and biogeography-based optimization algorithms. Cluster Computing, 26(6), 3855-3868.
[88] Zhao, W., Peng, Y., Xie, F., & Dai, Z., Modeling and simulation of cloud computing: A review. In 2012 IEEE Asia Pacific cloud computing congress (APCloudCC) (pp. 20-24). IEEE. (2012)
[89] Calheiros, R. N., Toosi, A. N., Vecchiola, C., & Buyya, R., A coordinator for scaling elastic applications across multiple clouds. Future Generation Computer Systems, 28(8), 1350-1362. (2012)
[90] Mounika, G., Rajyalakshmi, K., Rajkumar, G. V. S., Sravani, D. "Prediction and optimization of process parameters using design of experiments and fuzzy logic," International Journal on Interactive Design and Manufacturing (IJIDeM), 18(4), pp. 2333–2343, 2024. https://doi.org/10.1007/s12008-023-01446-x
[91] Wittig, A., & Wittig, M. (2023). Amazon Web Services in Action: An in-depth guide to AWS. Simon and Schuster.
[92] Manvi, S., & Shyam, G. (2021). Cloud computing: Concepts and technologies. CRC Press.

[93] Petcu, D., Portability and interoperability between clouds: challenges and case study. In Towards a Service-Based Internet: 4th European Conference, ServiceWave 2011, Poznan, Poland, October 26-28, 2011. Proceedings 4 (pp. 62-74). Springer Berlin Heidelberg. (2011)
[94] Arseniev, D. G., Overmeyer, L., Kälviäinen, H., & Katalinić, B. (Eds.), Cyber-Physical Systems and Control (Vol. 95). Springer Nature. (2019)
[95] Puri, T., Challa, R. K., & Sehgal, N. K., Energy-efficient delay-aware preemptive variable-length time slot allocation scheme for WBASN (edpvt). In Proceedings of 2nd International Conference on Communication, Computing and Networking: ICCCN 2018, NITTTR Chandigarh, India (pp. 183-194). Springer Singapore. (2019)
[96] Benlalia, Z., Beni-hssane, A., Abouelmehdi, K., & Ezati, A. A new service broker algorithm optimizing the cost and response time for cloud computing. Procedia Computer Science, 151, 992-997. (2019)
[97] Mesbahi, M. R., Hashemi, M., & Rahmani, A. M., Performance evaluation and analysis of load balancing algorithms in cloud computing environments. In 2016 Second International Conference on Web Research (ICWR) (pp. 145-151). IEEE. (2016)
[98] Achhra, S. N. M., Shah, R., Tamrakar, A., & Joshi, P. K., Prof Sowmiya Raksha,"Analysis OF Service Broker And Load Balancing In Cloud Computing,". International Journal Of Current Engineering And Scientific Research (IJCESR), 2(4), 92-98. (2015)
[99] Bahwaireth, K., Tawalbeh, L. A., Benkhelifa, E., Jararweh, Y., & Tawalbeh, M. A. (2016). Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. EURASIP Journal on Information Security, 1-14. (2016)
[100] Mondal, S., Faruk, F. B., Rajbongshi, D., Efaz, M. M. K., & Islam, M. M. (2023). GEECO: Green Data Centers for Energy Optimization and Carbon Footprint Reduction. Sustainability, 15(21), 15249.
[101] Liu, J., Yan, L., Yan, C., Qiu, Y., Jiang, C., Li, Y., ... & Cérin, C. (2023). Escope: An energy efficiency simulator for internet data centers. Energies, 16(7), 3187.
[102] Rimal, B. P., Choi, E., & Lumb, I. (2009). A taxonomy and survey of cloud computing systems. Network and Communication Technologies, 4(4), 1-10.
[103] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616.
[104] Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7-18.
[105] Biswas, A., Majumdar, S., Nandy, B., & El-Haraki, A. (2017). A hybrid auto-scaling technique for clouds processing applications with service level agreements. Journal of Cloud Computing, 6, 1-22.
[106] Hwang, K., Fox, G. C., & Dongarra, J. (2012). Distributed and cloud computing: From parallel processing to the internet of things. Morgan Kaufmann.
[107] Calheiros, R. N., Ranjan, R., De Rose, C. A. F., & Buyya, R. (2009). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23-50.
[108] Beloglazov, A., Buyya, R., Lee, Y. C., & Zomaya, A. Y. (2011). A taxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in Computers, 82, 47-111.
[109] Xiao, Z., Song, W., & Chen, Q. (2013). Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Transactions on Parallel and Distributed Systems, 24(6), 1107-1117.
[110] Wided, A., Çelebi, N., & Fatima, B. (2023). Effective Cloudlet Scheduling Algorithm for Load Balancing in Cloud Computing Using Fuzzy Logic. In Privacy Preservation and Secured Data Storage in Cloud Computing (pp. 226-243). IGI Global.
[111] Sangaiah, A. K., Javadpour, A., Pinto, P., Rezaei, S., & Zhang, W. (2023). Enhanced resource allocation in distributed cloud using fuzzy meta-heuristics optimization. Computer Communications, 209, 14-25.
[112] Aljuhani, A., & Alhubaishy, A. (2023). Dynamic Cloud Resource Allocation: A Broker-Based Multi-Criteria Approach for Optimal Task Assignment. Applied Sciences, 14(1), 302.
[113] Adami, D., Gabbrielli, A., Giordano, S., Pagano, M., & Portaluri, G. (2015, December). A fuzzy logic approach for resources allocation in cloud data center. In 2015 IEEE Globecom Workshops (GC Wkshps) (pp. 1-6). IEEE.

[114] Zaidi, R. T. (2018). Virtual Machine Allocation Policy in Cloud Computing Environment using CloudSim. International Journal of Electrical & Computer Engineering (2088-8708), 8(1).
[115] Gong, Y., Huang, J., Liu, B., Xu, J., Wu, B., & Zhang, Y. (2024). Dynamic resource allocation for virtual machine migration optimization using machine learning. arXiv preprint arXiv:2403.13619.
[116] Belgacem, A., Mahmoudi, S., & Kihl, M. (2022). Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. Journal of King Saud University-Computer and Information Sciences, 34(6), 2391-2404.
[117] Kang, J., Yu, S., & Yang, K. (2020). Energy-efficient resource allocation for cloud data centers using a hybrid heuristic algorithm. Journal of Supercomputing, 76(3), 1631-1649.
[118] Zhang, Q., Cheng, L., & Boutaba, R. (2020). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 11(1), 1-23.
[119] Taheri, H., Abrishami, S., & Naghibzadeh, M. (2023). A cloud broker for executing deadline-constrained periodic scientific workflows. IEEE Transactions on Services Computing.
[120] Balachandar, S., & Chinnaiyan, R. (2023). Intelligent Broker Design for IoT Using a Multi-Cloud Environment. In Convergence of Deep Learning and Internet of Things: Computing and Technology (pp. 23-41). IGI Global.
[121] Ramakrishnan, S. (Ed.). (2017). Modern Fuzzy Control Systems and Its Applications. BoD–Books on Demand.
[122] Gaur, A.; Garg, K. Survey paper on cloud computing with load balancing policy. Int. J. Eng. Res. 2015, 2, 7.
[123] Mateen, M., Hayat, S., Tehreem, T., & Akbar, M. A. (2020). A self-adaptive resource provisioning approach using fuzzy logic for cloud-based applications. International Journal of Computing and Digital Systems, 9(03).
[124] Shahid, M. A., Alam, M. M., & Su'ud, M. M. (2023). A systematic parameter analysis of cloud simulation tools in cloud computing environments. Applied Sciences, 13(15), 8785.
[125] Velte, A.T.V.T.J., Elsenpeter, P.D.R.: Cloud Computing, (2010).
[126] Giust, F., Costa-Perez, X., Reznik, A.: Multi-access edge computing: An overview
of etsi mec isg. IEEE 5G Tech Focus 1(4), 4 (2017)
[127] D'Agostino, D., Galizia, A., Clematis, A., Mangini, M., Porro, I., Quarati, A.: A qos-aware broker for hybrid clouds. Computing 95, 89–109 (2013)
[128] Shannaq, F., Alshorman, A., Al-Sayyed, R., Shehab, M., Alomari, W.: Weighted service broker algorithm in cloud environment. Informatica 48(7) (2024).
[129] Chauhan, S.S., Pilli, E.S., Joshi, R.C.: Bss: a brokering model for service selection using integrated weighting approach in cloud environment. Journal of Cloud Computing 10, 1–14 (2021).
[130] Shivakumar, U., Ravi, V., Gangadharan, G.: Ranking cloud services using fuzzy multi-attribute decision making. In: 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–8 (2013). IEEE
[131] Qu, L., Wang, Y., Orgun, M.A.: Cloud service selection based on the aggregation of user feedback and quantitative performance assessment. In: 2013 IEEE International Conference on Services Computing, pp. 152–159 (2013). IEEE
[132] Ionescu, S.: Best cloud broker of 2024 (2023). https://www.techradar.com/best/best-cloud-brokers.
[133] Sonmez, C., Ozgovde, A., Ersoy, C.: Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies 29(11), 3493 (2018)