Intelligent Cloud Automation for Smart City Using Fuzzy Inference System

Aqsa Noor[1], Muhammad Ibrahim[2], Muhammad Asif[3], Amna Ilyas [4]

[1,2] University of the Punjab, Lahore
[3] Department of Information Science, Division of Science and Technology, University of Education, Lahore, Pakistan
[4]University of South Asia
Corresponding Author: Aqsa Noor: f21mb009@ibitpu.edu.pk

Abstract:

Cloud computing is a comprehensive internet-based computing solution that enables computer services through the Internet. It has become an emerging technology to do business due to recent advances in cloud computing. Cloud services are essential while providing subscription, based on-demand services. A couple of available organizations provide their consumers with a wide range of Cloud services with similar characteristics. Because of the rapid development and broad availability of Cloud services, it isn't very easy for users to find the best service providers and the reasons behind the selection. Selecting a suitable cloud service provider can be a challenging task involving quantitative and qualitative criteria. A research study proposed a hierarchical Mamdani fuzzy expert system to overcome this challenge. This system efficiently handles the uncertainty of the data involved in multi-criteria decision-making problems. The study shows that the Mamdani fuzzy expert system is an effective tool for selecting a cloud service provider that can handle quantitative and qualitative decision criteria in the existing business scenario.

Keywords: Mamdani; cloud computing; characteristics; fuzzy expert system

Introduction:

The most appealing aspect of cloud computing was that resources could be accessed on-demand and when necessary. The Information Technology (IT) teams and developers can automatically build, change, and decommission cloud services by cloud automation. In the real world, individuals need to set up tools, test their functionality, decide when they are no longer necessary, and subsequently remove them, which can consume a lot of time. As virtualization and cloud computing are increasingly adopted, the workload related to managing these systems is increasing. Manual tasks such as scaling, provisioning, configuring resources, setting up virtual machines, and measuring performance are repetitive, expensive, and prone to errors that may affect system availability.

Cloud automation tools are designed to streamline cloud computing operations and ensure the system's and its services' optimal performance. Although some upfront costs may be associated with purchasing and configuring a Cloud Automation Service (CAS), the long-term benefits outweigh the initial investment. Once established, procedures can be easily replicated with minimal human intervention and even scheduled to run automatically regularly. Applying cloud automation to manage cloud resources provides a consistent collection of predictable processes and policies that can quickly adapt to evolving customer demands, giving more flexibility and versatility to the system. Cloud automation also allows cloud services to be used effectively and prevent security flaws in situations where teams depend highly on manual, bug-prone workflows.

Cloud computing permits us to create a counterfeit digital computer that performs like a real computer with its hardware, technically known as a virtual machine. Cloud computing delivers computing services as a platform, with the cloud provider controlling and operating the resources rather than the end user. It allows small companies that cannot afford their internal structure to subcontract their necessities to the cloud at an affordable rate. Cloud computing significantly reduces IT costs as companies don't have to update or maintain their servers because the cloud provider will do it for them. Cloud users can avoid the expenses, efforts, and specialized knowledge required for purchasing and managing computing resources by transitioning from outdated, principal software and hardware to remote distributed resources retrieved through a network.

Cloud computing is different from the client-server model because cloud servers respond to client requests and run programs and store data. Cloud providers can serve more users at a lower cost by running multiple virtual machines simultaneously. The location of cloud servers and who manages them are described by different cloud deployment types like public, private, hybrid, and multi-cloud. Cloud automation involves automating manual tasks associated with running cloud-based IT services using advanced software techniques and processes. It helps minimize administration costs and achieve workflow objectives such as continuous integration and deployment. Cloud automation uses cloud management tools to automate tasks such as auto-provisioning servers, backing up files, and identifying and removing unused processes without human intervention. Ultimately, cloud automation enhances productivity and reduces manual workload by automating operations and processes.

Today, IT enterprises use cloud orchestration and automation techniques to automate routine tasks like provisioning virtual services, identifying common configuration objects, and creating IaC in a virtualized environment. Cloud automation has several advantages to saving IoT costs by balancing workload placement so that the least costly hardware is used, and essential programs are prioritized. The continuous deployment aims to automate the application deployment pipeline to improve the speed at which applications are updated and make it easier to do quality assurance and monitoring. It may promote organizational creativity by streamlining routine operations, allowing the most creative developers, IT managers, and security specialists and providing constant control and monitoring.

Machine learning (ML) enables computers to learn from data without human interaction and make decisions. Research promotes machines to recognize and develop programs that make their actions and decisions more humane. ML is a powerful technique that allows computers to learn from data iteratively instead of being explicitly programmed. It enables the analysis, prediction, and sorting of large amounts of data to extract valuable insights. The process begins with data, rules, and assumptions to make more informed decisions in the future. ML methods in cloud computing encourage the cloud to draw assumptions, decide, and actions deprived of human contribution in real-life environments. The automation of user troubleshooting is cloud automation, which must solve the problems consumers face based on the past performance delivered to the customer. It allows one to use ML models and methods to take automation to a different level. Another factor that may benefit from ML algorithms to optimize customer experience is examining user behaviour to understand the needs and preferences. ML may also track usage patterns and draw suitable conclusions in appropriate circumstances [10,11, 12].

Literature Review:

Cloud automation systems have been developed by many researchers who have applied multiple techniques. A few of the research works regarding this research are discussed in this section. A research study [5] describes developing a system that addresses the trade-off between computing cost and efficiency. The authors proposed an online ML answer for resource provision, which enables users to choose between spot and on-demand instances. The ML algorithm proposed in the study dynamically adjusted the resource provision policy via learning from its act on past task performances, although considering historical spot costs and capacity characteristics. Furthermore, their approach permitted the interpretation of these policies' allocation strategies and allowed consumers to adapt them to their needs. They claim that the online machine learning solution may be helpful for cloud computing resource allocation.

In [3], the researchers used RL to allocate resources autonomously in clouds to cope with the key desires of self-adapting cloud systems. This strategy proved well-suited to cloud computing as it was not required to understand the application performance. Reinforcement learning (RL) has faced various challenges, including creating efficient policies in the initial stages of learning, ensuring enough time for convergence towards an optimal policy, and adapting to changes in the application's behaviour. To tackle these issues, the authors have implemented appropriate initialization techniques during the early stages, accelerated the convergence process during the learning phase, and developed mechanisms to detect any changes in the model's performance.

The authors in [1] proposed an ML-based architecture that predicts performance issues, manages cloud resources, and meets SLAs without downtime. The study also explored the feasibility of using ML performance models to estimate implementation costs. The challenge is that simple models will not capture cloud workload/performance relationships and will be unable to keep up with evolving circumstances that potentially invalidate the models. In a research work [2], it is claimed that a collection of modelling, control, and analysis methods embedded in SML will solve these problems related to cloud automation. They proposed a three-component control system. The architecture's base is built on rich SPMs, which allow system performance prediction for future setups and workloads. Second, they use a CPS that simulates the performance model to compare various policies for adding and removing resources to identify a management policy that minimizes resource usage while retaining performance. Finally, they use model management tools like online training and CPD to update the models when improvements in application performance are detected.

In [8,9], researchers familiarized "SmartSLA", a cost-sensitive resource organization system. It is made up of two meaningful parts: SMM and RADM. The SMM learns a model applying ML approaches that characterize each customer's possible profit limitations under varying allocations of resources. The RADM component of SmartSLA adjusts resource allocations based on the learned model to maximize profits. The results demonstrate that SmartSLA can accurately generate predictive models for hardware and database-specific resources (e.g., number of database replicas) while achieving effective performance outcomes. The test results also demonstrate that SmartSLA can offer smart distinguishing services by factors including flexible workload, SLA levels, resource expenses, and increased margins for profit.

According to [4], the researchers introduced "SysWeka", a middleware platform, while applying ML techniques based on system-oriented Weka. It offered a cloud-based resource management software platform that could be used for higher-level cloud applications. This architecture made it simple to build frameworks using lower-level machinemade, simple to build frameworks using lower-level machinemade and simple to build frameworks using lower-levelmachine-learning infrastructures [13]. While LR has been commonly used to create models with promising results in various fields, it cannot deliver beneficial outcomes.

Multiple VMs can share resources on a single host using virtual machine technology. To respond to changes in application demands or resource availability, the assigned resources of the VMs must be dynamically reconfigured. Since VM implementation uses an advantaged domain and a VM monitor, there are certain inconsistencies in VMs' resource to recital mapping and difficulties in determining suitable VM configurations in real time. To automate the VM configuration operation in the cloud environment, [7] suggested an RL-based solution called "VCONF". It used model-based RL procedures to solve scalability and adaptability problems in implementing RL in organizations' administration. VCONF automates VM reconfiguration using learned policies, demonstrated to be effective in managed environments and cloud testbeds with Xen VMs and server workloads, achieving suitable configurations for small-scale applications with high adaptability and scalability.

Virtualized cloud resource management is a crucial and difficult challenge, particularly when interacting with varying workloads and multifaceted multi-tier server applications. Control theory-based resource organization can optimize resource allocations to meet varying workloads. In [6], authors demonstrated a novel resource management scheme that automatically incorporates the Kalman filter into response managers to distribute CPU properties to VMs hosting server claims. The key controller's display and self-adapt to unexpected workload rate variations constantly. Using historical usage insights, the adaptive controller configures its settings online. Their modern controller adapts to any workload situation without requiring any previous knowledge.

Numerous research studies could be used for cloud automation like Fuzzy based [14], and different techniques [15,16,17,18,19,20] have been applied in a novel era for researchers.

Proposed Methodology

Cloud automation resolves are instrumental in minimizing IT infrastructure costs by enabling speedy integration, management, and deployment of virtual machines, network devices, and routers. Implementing cloud automation in business operations can be performed and executed rapidly with increased agility. This research proposed a cloud automation system to efficiently tackle the existing business scenario's quantitative and qualitative decision criteria. The proposed system is shown in Fig. 1.

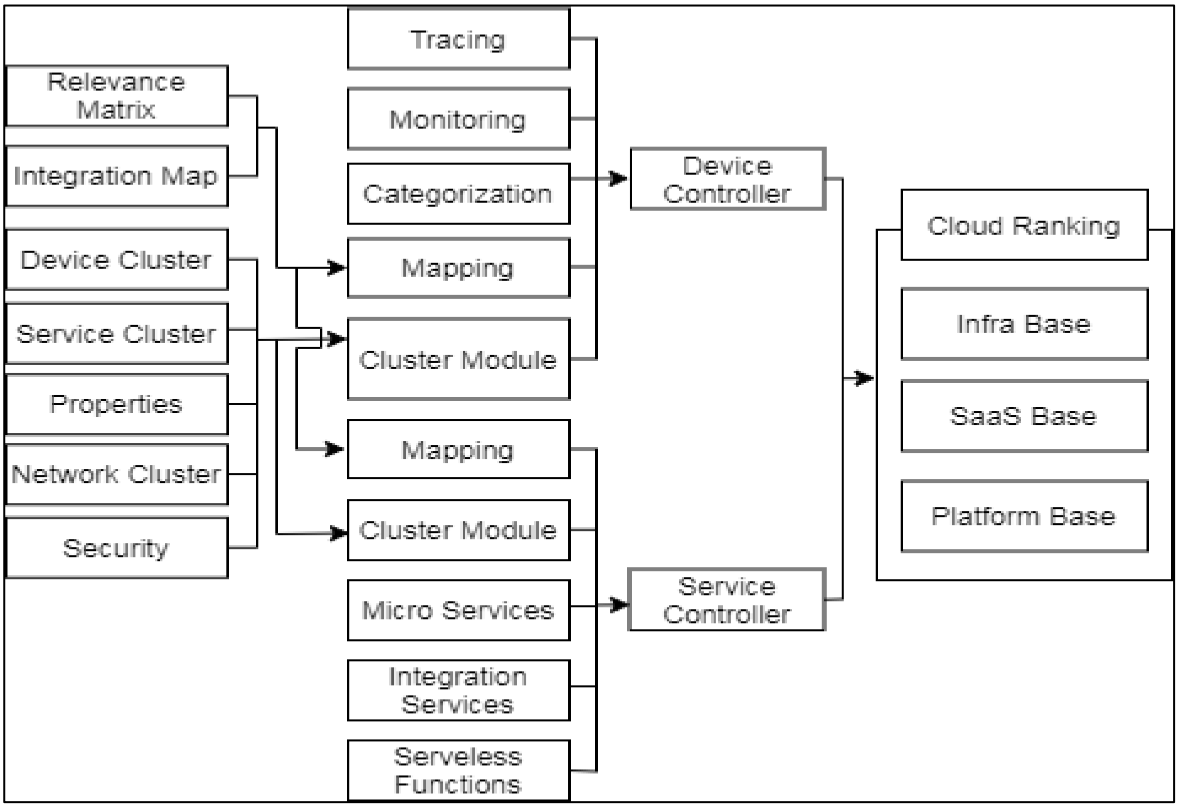


Figure 1: Proposed cloud automation system

Fig. 1 shows the proposed Cloud Ranking (CR) model, a layered architecture. It is clearly shown that CR's decision is based on the device and Service Controller (SCo). Device Controller (DC) decision is based on tracing, monitoring, categorization, Mapping (M) and clustering modules. While in SCo decision-making is based on mapping, Cluster Module (CM), Micro Services (MS), Integration Services (IS), as well as Services Functions (SF). Mapping decision-making is based on a Relevant Matrix (RM) and Integration Map (IM). At the same time, CM decision-making is based on Device Cluster (DC), Service Cluster (SC), properties, Network Cluster (NC), in addition, security.

Membership functions:

A Membership Function (MF) is a statistical tool to describe how input and output variables are related and how they are mapped to values ranging from 0 to 1. The proposed model's Input/Output Variables are presented in Table 1, both graphically and mathematically. The input member function is represented by rows 2 through 11, and the output member function is represented by rows 12 through 13.

Table 1: MFs of mapping

Input/Output	MFs	Graphical Depiction of MF

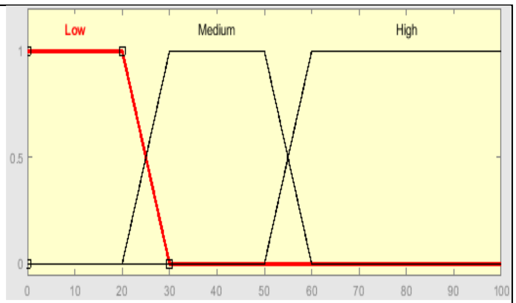
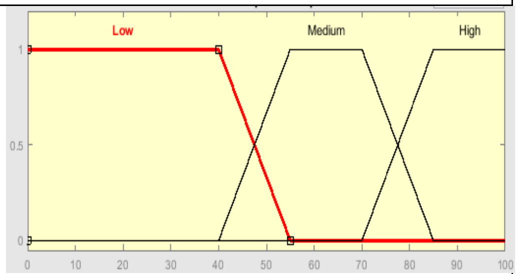
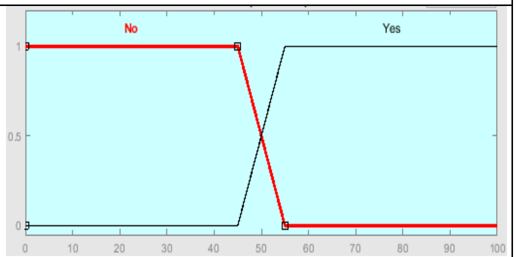
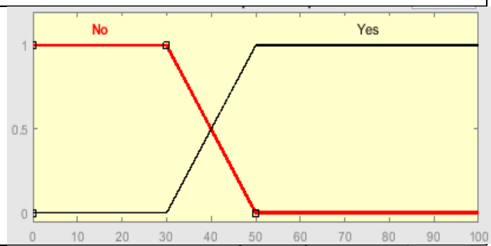
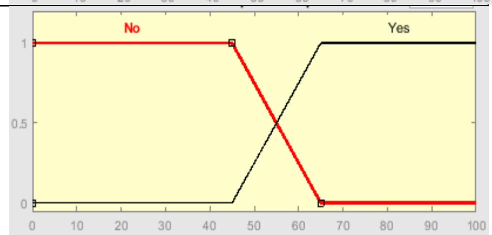
RM= $\mu_{RM}(rm)$	$\mu_{RM_l}(rm) =$ $\{\max(\min(1, \frac{30-rm}{10}), 0)\}$ $\mu_{RM_m}(rm) =$ $\{\max(\min(\frac{rm-20}{10}, 1, \frac{60-rm}{10}), 0)\}$ $\mu_{RM_h}(rm) =$ $\{\max(\min(\frac{rm-50}{10}, 1), 0)\}$	
IM $=\mu_{IM}(im)$	$\mu_{IM_l}(im)$ $= \{\max(\min(1, \frac{55-im}{15}), 0)\}$ $\mu_{IM_m}(im) =$ $\{\max(\min(\frac{im-40}{15}, 1, \frac{85-im}{15}), 0)\}$ $\mu_{IM_h}(im) = \{\max(\min(\frac{im-70}{15}, 1), 0)\}$	
M=$\mu_M(m)$	$\mu_{M_n}(m) = \{\max(\min(1, \frac{55-m}{10}), 0)\}$ $\mu_{M_y}(m) = \{\max(\min(\frac{m-45}{10}, 1), 0)\}$	

Table 2: MFs of CM

Input/Output	**MFs**	**Graphical Depiction of MF**
DC= $\boldsymbol{\mu_{DC}(dc)}$	$\boldsymbol{\mu_{DC_n}(dc) = \{\max(\min(1, \frac{50-dc}{20}), 0)\}}$ $\boldsymbol{\mu_{DC_y}(dc) =}$ $\boldsymbol{\{\max(\min(\frac{dc-30}{20}, 1), 0)\}}$	
SC=$\boldsymbol{\mu_{SC}(sc)}$	$\boldsymbol{\mu_{SC_n}(sc) = \{\max(\min(1, \frac{65-sc}{20}), 0)\}}$ $\boldsymbol{\mu_{SC_y}(sc) = \{\max(\min(\frac{sc-45}{20}, 1), 0)\}}$	

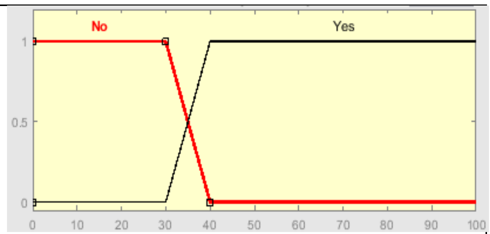
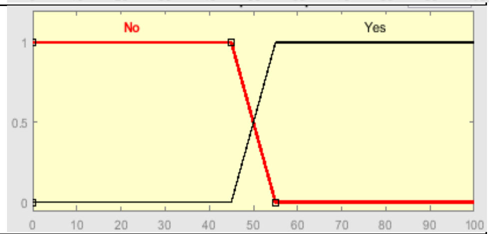
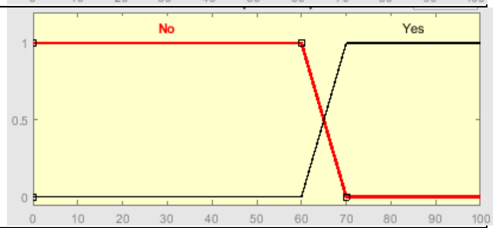
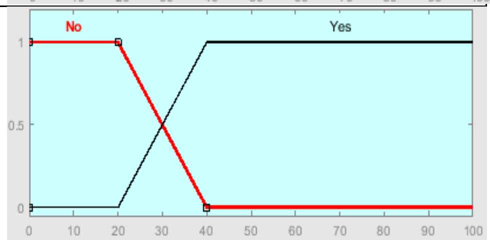
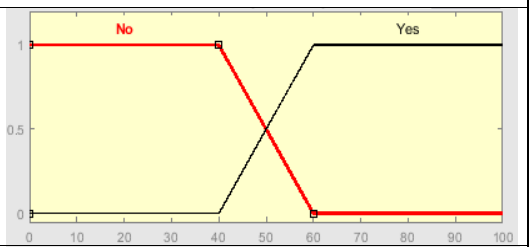
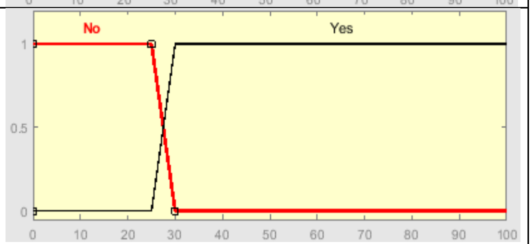
Properties (P)= $\mu_P(p)$	$\mu_{P_n}(p) = \{\max(\min\left(1, \frac{40-p}{10}\right), 0)\}$ $\mu_{P_y}(p) = \{\max(\min\left(\frac{p-30}{10}, 1\right), 0)\}$	
NC= $\mu_{NC}(nc)$	$\mu_{CN_n}(nc) = \{\max(\min\left(1, \frac{55-nc}{10}\right), 0)\}$ $\mu_{CN_y}(nc) = \{\max(\min\left(\frac{nc-45}{10}, 1\right), 0)\}$	
Security (S)= $\mu_S(s)$	$\mu_{S_n}(s) = \{\max(\min\left(1, \frac{70-s}{10}\right), 0)\}$ $\mu_{S_y}(s) = \{\max(\min\left(\frac{s-60}{10}, 1\right), 0)\}$	
CM= $\mu_{CM}(cm)$	$\mu_{CM_n}(cm) = \{\max(\min\left(1, \frac{40-cm}{20}\right), 0)\}$ $\mu_{CM_y}(cm) = \{\max(\min\left(\frac{cm-20}{20}, 1\right), 0)\}$	

Table 3: MFs of the DC

Input/Output	MFs	Graphical Depiction of MF
Tracing (T)= $\mu_T(t)$	$\mu_{T_n}(t) = \{\max(\min\left(1, \frac{60-t}{20}\right), 0)\}$ $\mu_{T_y}(t) = \{\max(\min\left(\frac{t-40}{20}, 1\right), 0)\}$	
Monitoring (M)= $\mu_M(m)$	$\mu_{M_n}(m) = \{\max(\min\left(1, \frac{30-m}{5}\right), 0)\}$ $\mu_{M_y}(m) = \{\max(\min\left(\frac{m-25}{5}, 1\right), 0)\}$	

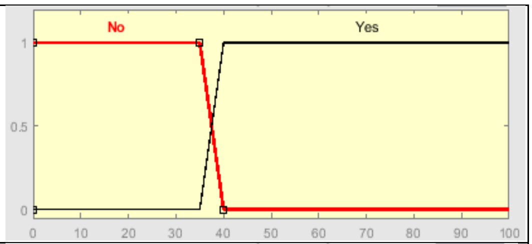
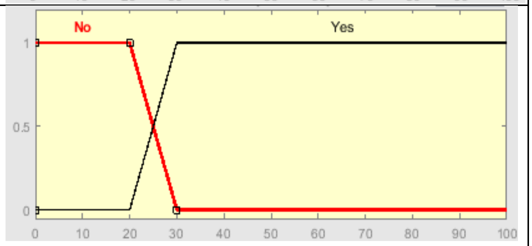
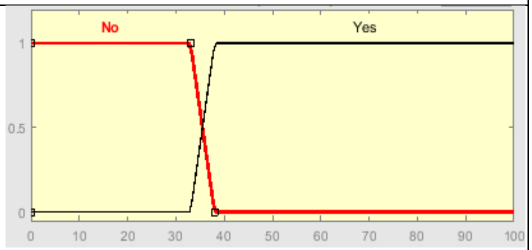
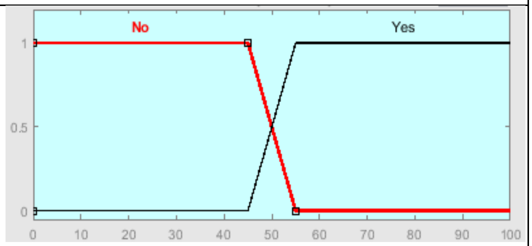
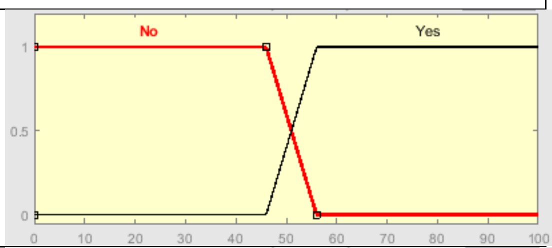
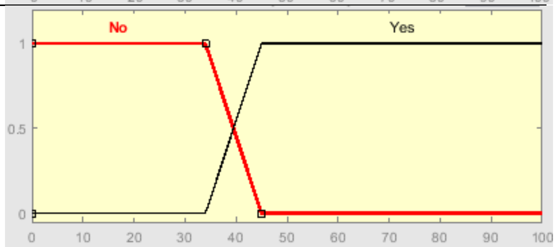
Categorization (C)=$\mu_C(c)$	$\mu_{C_n}(c)=\{\max\left(\min\left(1,\frac{40-c}{5}\right),0\right)\}$ $\mu_{C_y}(c)=\{\max\left(\min\left(\frac{c-35}{5},1\right),0\right)\}$	
M=$\mu_M(m)$	$\mu_{M_n}(m)$ $=\{\max\left(\min\left(1,\frac{30-m}{10}\right),0\right)\}$ $\mu_{M_y}(m)=\{\max\left(\min\left(\frac{m-20}{10},1\right),0\right)\}$	
CM=$\mu_{CM}(cm)$	$\mu_{CM_n}(cm)$ $=\{\max\left(\min\left(1,\frac{40-cm}{5}\right),0\right)\}$ $\mu_{CM_y}(cm)=$ $\{\max\left(\min\left(\frac{cm-30}{5},1\right),0\right)\}$	
DC=$\mu_{DC}(dc)$	$\mu_{DC_n}(dc)$ $=\{\max\left(\min\left(1,\frac{55-dc}{10}\right),0\right)\}$ $\mu_{DC_y}(dc)=$ $\{\max\left(\min\left(\frac{dc-45}{10},1\right),0\right)\}$	

Table 4: MFs of SCo

Input/Output	**MFs**	**Graphical Depiction of MF**
M= $\mu_M(m)$	$\mu_{M_n}(m)=\{\max(\min\left(1,\frac{55-m}{10}\right),0)\}$ $\mu_{M_y}(m)=\{\max(\min\left(\frac{m-45}{10},1\right),0)\}$	
CM= $\mu_{CM}(cm)$	$\mu_{CM_n}(cm)$ $=\{\max\left(\min\left(1,\frac{45-cm}{10}\right),0\right)\}$ $\mu_{CM_y}(cm)=$ $\{\max\left(\min\left(\frac{cm-35}{10},1\right),0\right)\}$	

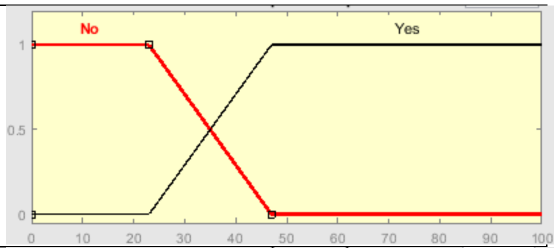
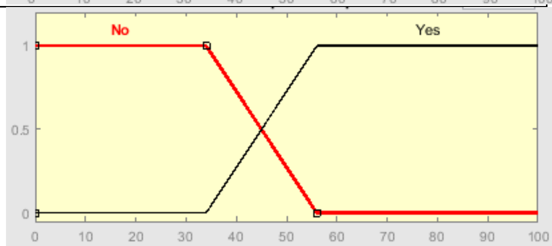
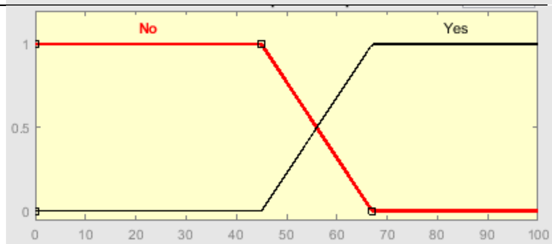
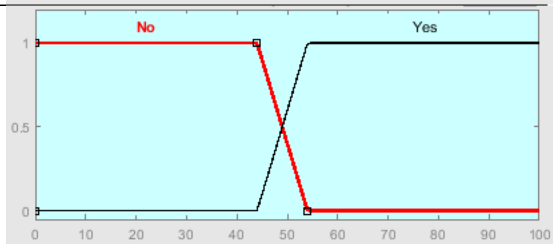
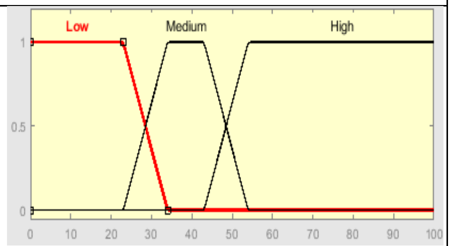
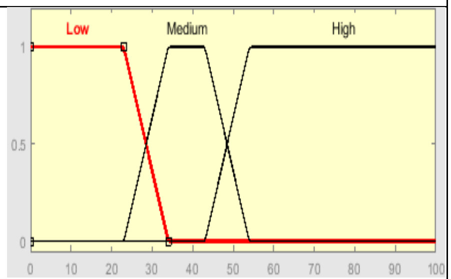
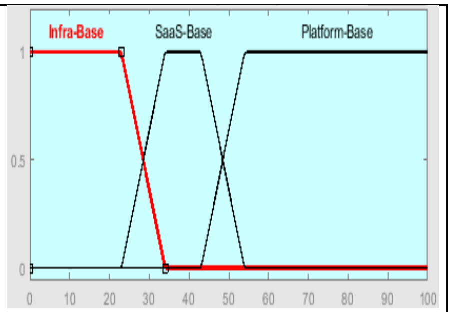
MS=$\boldsymbol{\mu_{MS}(ms)}$	$\mu_{MS_n}(ms) = \{\max(\min\left(1,\frac{50-ms}{30}\right),0)\}$ $\mu_{MS_y}(ms) = \{\max(\min\left(\frac{ms-20}{30},1\right),0)\}$	
IS=$\boldsymbol{\mu_{IS}(is}$	$\mu_{IS_n}(is) = \{\max(\min\left(1,\frac{55-is}{20}\right),0)\}$ $\mu_{IS_y}(is) = \{\max(\min\left(\frac{is-35}{20},1\right),0)\}$	
SF(SF)= $\boldsymbol{\mu_{SF}(sf)}$	$\mu_{SF_n}(sf) = \{\max(\min\left(1,\frac{70-sf}{20}\right),0)\}$ $\mu_{SF_y}(sf) = \{\max(\min\left(\frac{sf-45}{20},1\right),0)\}$	
SCo=$\boldsymbol{\mu_{SC}(sc)}$	$\mu_{SCo_n}(sco) = \{\max(\min\left(1,\frac{55-sco}{10}\right),0)\}$ $\mu_{SCo_y}(sco) = \{\max(\min\left(\frac{sco-45}{10},1\right),0)\}$	

Table 5: MFs of CR

Input/Output	MFs	Graphical Depiction of MF
DC= $\mu_{DC}(dc)$	$\mu_{DC_l}(dc) = \{\max(\min\left(1,\frac{35-dc}{15}\right),0)\}$ $\mu_{DC_m}(dc) = \{\max(\min\left(\frac{dc-25}{15},1,\frac{55-dc}{15}\right),0)\}$ $\mu_{DC_h}(dc) = \{\max(\min\left(\frac{dc-45}{15},1\right),0)\}$	
SCo= $\mu_{SC}(sco)$	$\mu_{SC_l}(sc) = \{\max(\min\left(1,\frac{35-sc}{10}\right),0)\}$ $\mu_{SC_m}(sc) = \{\max(\min\left(\frac{sc-25}{10},1,\frac{55-sc}{10}\right),0)\}$ $\mu_{SC_h}(sc) = \{\max(\min\left(\frac{sc-45}{10},1\right),0)\}$	

CR=$\mu_{CR}(cr)$	$\mu_{CR_l}(cr) = \{\max\left(\min\left(1, \frac{35-cr}{10}\right), 0\right)\}$ $\mu_{CR_m}(cr) = \{\max\left(\min\left(\frac{cr-25}{10}, 1, \frac{55-cr}{10}\right), 0\right)\}$ $\mu_{CR_h}(cr) = \{\max\left(\min\left(\frac{cr-45}{10}, 1\right), 0\right)\}$	

I. Mapping

To build a fuzzy rule base, conditional statements of fuzzy logic known as if-then statements are utilized. Such statements are crucial in constructing a map. Below are some rules for the fuzzy inference procedure:

1. If RM is Low in addition, IM is Low, M is No
2. If RM is Low in addition, IM is Medium, M is No
3. If RM is Low in addition IM is High, M is No
4. If RM is High in addition IM is High, M is Yes

II. Cluster Module

1. If DC is Yes, SC is Yes, Properties is Yes, NC is Yes as well as Security is No, CM is Yes
2. If DC is Yes, SC is Yes, Properties is Yes, NC is Yes as well as Security is Yes, CM is Yes
3. If DC is No, SC is No; Properties is Low, NC is Yes, as well as Security is Yes, CM is No

.

.

.

32.If DC is Yes and SC is Yes, Properties is Yes, NC is No, as well as Security is No, CM is Yes

III. Device Controller

1. If there is No Tracing, No Monitoring, No Categorization, M is No, as well as CM is No, DC is No
2. If there is No Tracing, Monitoring is Yes; Categorization is Yes, M is Yes, as well as CM is Yes, DC is Yes
3. If there is Tracing, No Monitoring, No Categorization, M is No, as well as CM is No, DC is No

.

.

.

32.If there is Tracing, there is Monitoring, No Categorization, M is No, as well as CM is Yes, DC is Yes

IV. Service Controller

1. If M is No, CM is Yes, MS is No, IS is Yes, SF is No, SCo is No
2. If M is No, CM is Yes, MS is No, IS is Yes, as well as SF is Yes, SCo is Yes
3. If M is No, CM is Yes, MS is Yes, IS is Yes, as well as SF is Yes, SCo is Yes

.

.

.

32. If M is Yes, CM is No, MS is No, IS is Yes, as well as SF is No, SCo is No

V. Cloud Ranking

1. If DC is Low, SC is Low, CR is Infra-Base
2. If DC is Medium, SC is Medium, CR is SaaS-Base
3. If DC is Medium, SC is High, CR is Platform-Base

.

.

.

9. If DC is High, SC is High, CR is Platform-Base

Defuzzification refers to converting a fuzzy set, which involves degrees of membership, into a computable outcome in Crisp logic. This involves comparing membership degrees and transforming the fuzzy set into a new set. In fuzzy control frameworks, it is expected to be representative. De-Fuzzifiers are available in a wide range of shapes and sizes. In this model, a centroid form of a De-Fuzzifier is utilized. Figures 2 to 6 depict the graphical representation of the De-Fuzzifier for the FIS of M, CM, DC, SCo, and CR.

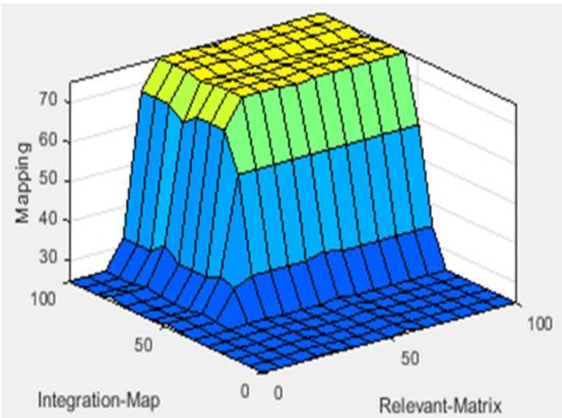


Figure 2(a): Mapping rule surface centered on IM as well as RM

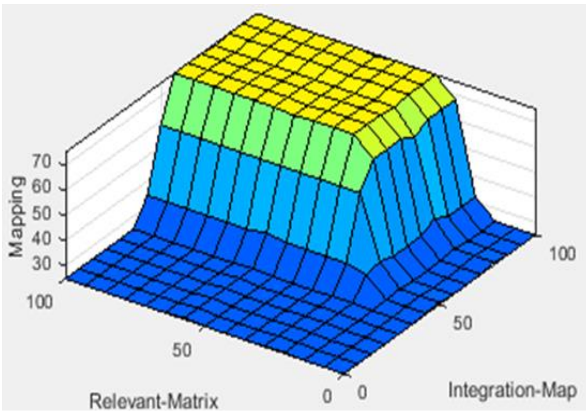


Figure 2(b): Mapping rule surface centered on RM as well as IM

Figure 2: Rule surface of mapping

Fig. 2 (a) illustrates that if the IM lies between 0-50 and the RMbetween 30-80, the mapping is Good (Yellowish). If the IM lies between 50-90 and the RM between 30-80, the mapping is Satisfactory (Greenish). If the IM lies between 0-100 and the RM between 0-100, the mapping is Bad (Bluish).

Fig. 2 (b) illustrates that if the RM lies between 20-80 and the IM between 50-80, the mapping is Good (Yellowish). If the RM lies between 25-80 and the IM among 60-80, the mapping is Satisfactory

(Greenish). If the RM lies among 0-100 and to the IM among 0-100, at that point mapping is Bad (Bluish).

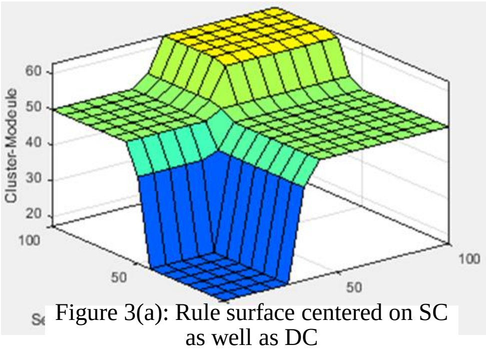


Figure 3(a): Rule surface centered on SC as well as DC

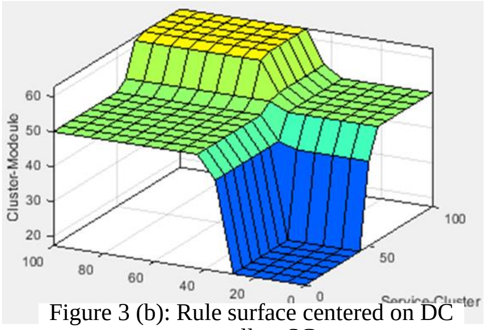


Figure 3 (b): Rule surface centered on DC as well as SC

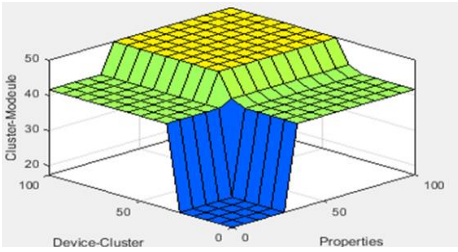


Figure 3(c): Rule surface centered on DC as well as properties

Figure 3: Rule surface of CM

Fig. 3 (a) illustrates that if the SC lies among 0-50 in addition the DC among 0-50, the CM is Good (Yellowish). If the SC lies amongst 50-100 in addition the DC among 50-100, the module is Satisfactory (Greenish). If the SC lies among 0-40 in addition the DC among 0-30, then the CM is Bad (Bluish).

Fig. 3 (b) illustrates that if the DC lies among 50-100 and the SC among 80-100, the CM is Good (Yellowish). If the DC lies among 40-100 and the SC among 50-100, the CM is Satisfactory (Greenish). If the DC lies between 0-30 plus the DC among 0-40, the CM is Bad (Bluish).

Fig. 3 (c) illustrates that if the DC lies among 40-100 and properties among 50-100, the CM is Good (Yellowish). If the DC lies among 50-100 and properties among 50-100, then the CM is Satisfactory (Greenish). If the DC lies among 0-30 in addition to properties among 0-30, the CM is Bad (Bluish).

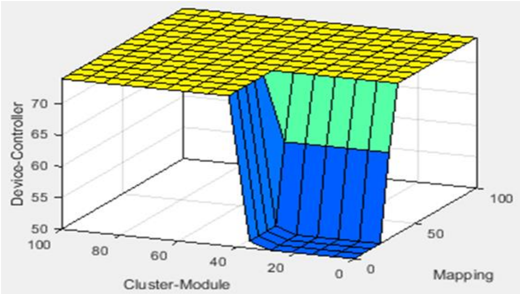


Figure 4(a): Rule surface centered on SC as well as DC

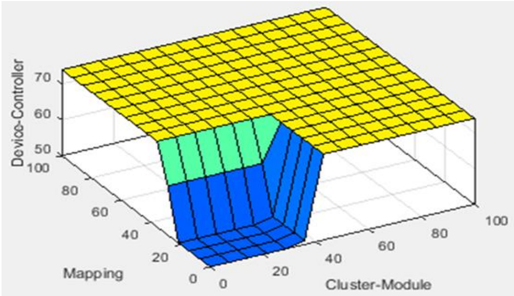


Figure 4(b): Rule surface centered on DC as well as SC

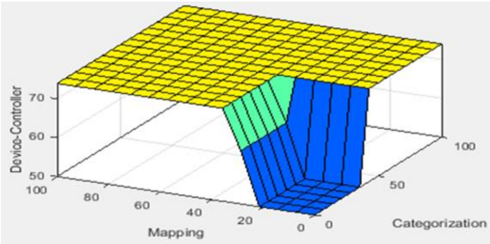


Figure 4(c): Rule surface centered on DC as well as properties

Figure 4: Rule surface of the DC

Fig. 4 (a) illustrates that if the CM lies among 40-100 and mapping among 50-100, the DC is Good (Yellowish). If the CM lies among 0-20 and mapping among 40-50, then the DC is Satisfactory (Greenish). If the CM lies among 0-45 and mapping among 0-20, then the DC is Bad (Bluish).

Fig. 4 (b) illustrates that if mapping lies among 40-100 and the CM among 50-100, the DC is Good (Yellowish). If mapping lies among 20-40 and CM among 0-20, the DC is Satisfactory (Greenish). If mapping lies among 0-20 and CM among 0-40, the DC is Bad (Bluish).

Fig. 4 (c) illustrates that if mapping lies between 40-100 and categorization among 50-100, the DC is Good (Yellowish). If mapping lies among 30-40 and categorization among 0-20, then the DC is Satisfactory (Greenish). If mapping lies among 0-25 and categorization between 0-40, then the DC is Bad (Bluish).

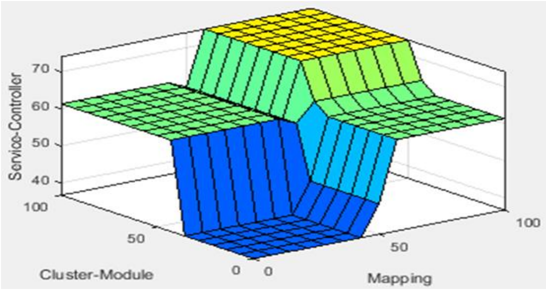


Figure 5(a): Rule surface centered on SC as well as DC

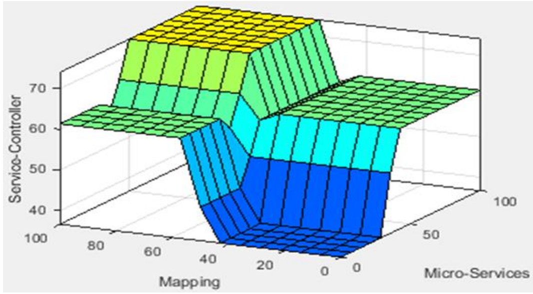


Figure 5(b): Rule surface centered on DC as well as SC

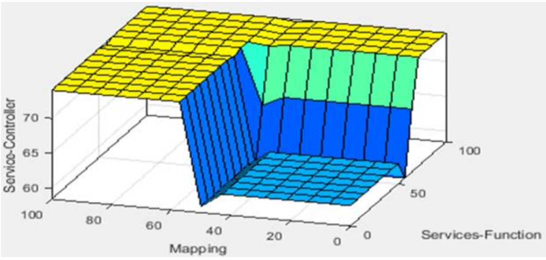


Figure 5(c): Rule surface centered on DC as well as properties

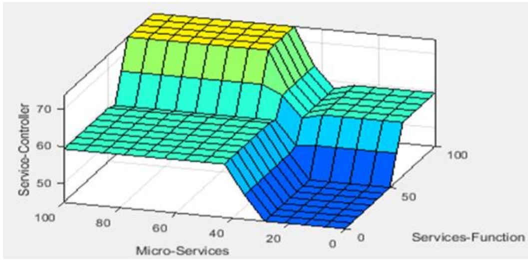


Figure 5(d): Rule surface centered on MS as well as SF

Figure 5: Rule surface of the DC

Fig. 5 (a) illustrates that if the CM lies between 50-100 and mapping among 50-100, the SCo is Good (Yellowish). If the CM lies among 50-100 and mapping among 50-100, the SCo is Satisfactory (Greenish). If the CM lies among 0-30 and mapping among 0-50, the SCo is Bad (Bluish).

Fig. 5 (b) illustrates that if mapping lies among 60-100 and MS among 60-100, the SCo is Good (Yellowish). If mapping lies among 50-100 and MS among 50-100, the SCo is Satisfactory (Greenish). If the CM lies among 0-40 and MS among 0-30, the SCo is Bad (Bluish).

Fig. 5 (c) illustrates that if mapping lies among 60-100 and SF among 80-100, the SCo is Good (Yellowish). If mapping lies among 0-40 and SF among 60-80, the SCo is Satisfactory (Greenish). If the CM lies among 0-50 and SF among 0-50, the SCo is Bad (Bluish).

Fig. 5 (d) illustrates that if MS lie among 50-100 and SF among 80-100, the SCo is Good (Yellowish). If mapping lies among 40-100 and SF among 60-100, the SCo is Satisfactory (Greenish). If the CM lies among 0-40 and SF among 0-50, the SCo is Bad (Bluish).

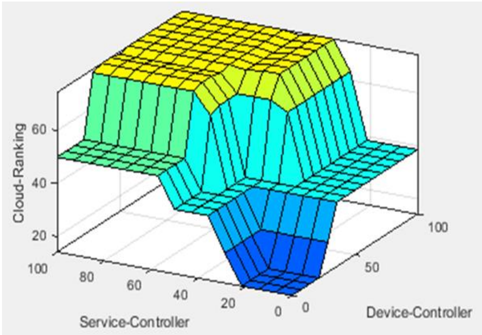


Figure 6(a): CR rule surface centered on SCo as well as DC

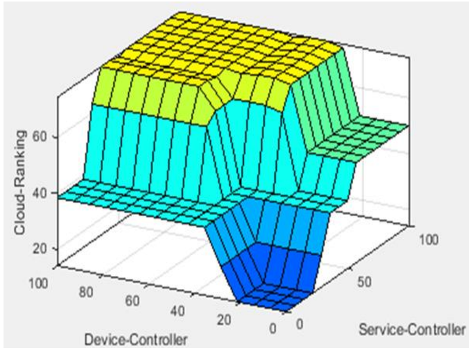


Figure 2(b): CR rule rule surface centered on device controller as well as SC

Figure 6: Rule surface of CR

Fig. 6 (a) illustrates that CR is good if the SCo lies among 60-100 and the DC among 60-100 (Yellowish). If the SCo lies among 30-60 and the DC among 60-100, the CR is Satisfactory (Greenish). If the SCo lies among 0-50 and the DC among 0-40, the CR is Bad (Bluish).

Fig. 6 (a) illustrates that CR is good if the DC lies among 50-100 and SCo among 60-100 (Yellowish). The CR is satisfactory if the DC lies among 40-100 and the SCo among 30-60 (Greenish). If the DC lies among 0-40 and the SCo among 0-40, the CR is Bad (Bluish).

4 Simulation Results of Proposed Cloud Automation

In fuzzy logic, Boolean logic deals with partly true or false values. Boolean values or membership values in fuzzy sets are characterized in fuzzy logic by a number on the scale [0, 1], with 0 indicating absolute Falseness also 1 indicating whole truth. The fuzzy system is simulated on MATLAB for simulation results, and the simulated graphs are introduced in (Figs. 7-11). MATLAB is also used in modelling, simulation, algorithm development, prototyping, and other domains. Five data sources and one performance factor produce the reproduction results. The proposed cloud automation performance evaluation is demonstrated in this research from the system forecasts.

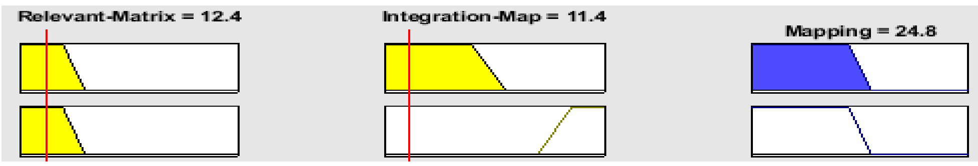


Figure 7(a): Lookup graph of mapping (No)

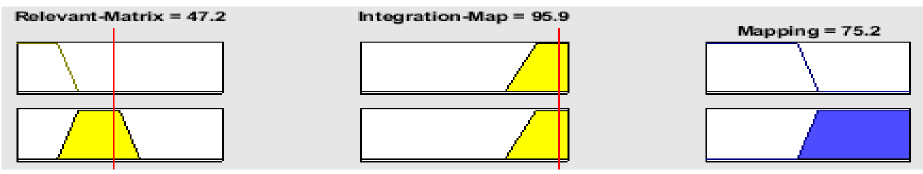


Figure 7(b): Lookup graph of mapping (Yes)

The information presented in Figure 7(a) suggests that when the RM values are Low, the IM is also Low, resulting in a mapping of No. Similarly, Figure 7(b) indicates that if the RM ranges are Medium, the IM becomes High, and the mapping becomes Yes.

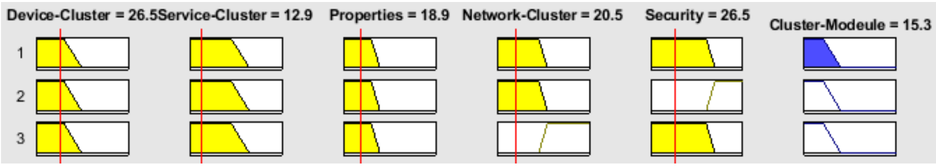


Figure 8(a): Lookup graph of mapping (No)

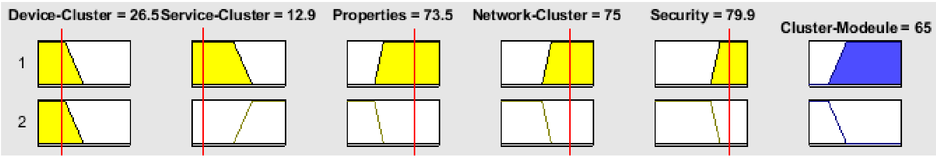


Figure 8(b): Lookup graph of mapping (Yes)

In Fig. 8(a), the absence of values in the DC, SC, properties, NC, and security results in the absence of the CM. In contrast, Fig. 8(b) demonstrates that the absence of values in the DC and SC, coupled with the presence of values in properties, NC, and security, leads to the presence of the CM.

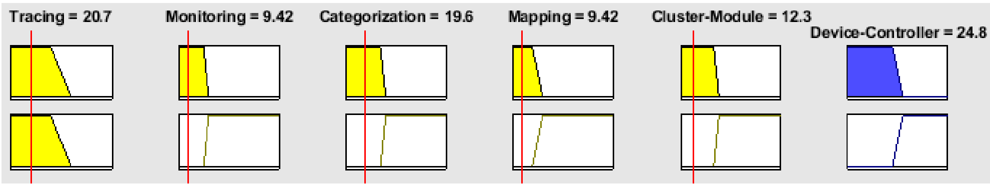


Figure 9(a): Lookup graph of DC (No)

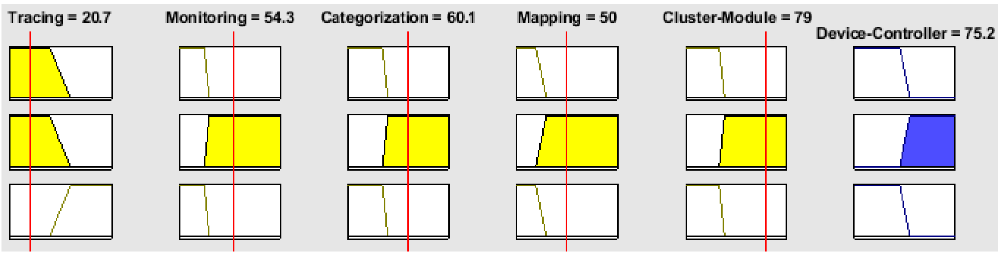


Figure 9(b): Lookup graph of DC (Yes)

In Fig. 9(a), when tracing, monitoring, categorization, mapping, and CM are set to "No", the DC is also "No". On the other hand, Fig. 9(b) indicates that when tracing is set to "No" and monitoring, categorization, mapping, and CM are all set to "Yes", the DC is "Yes".

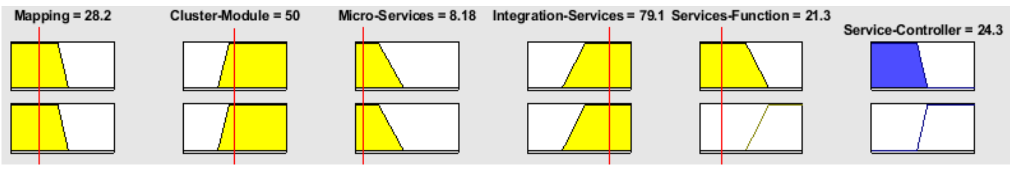


Figure 10(a): Lookup graph of SCo (No)

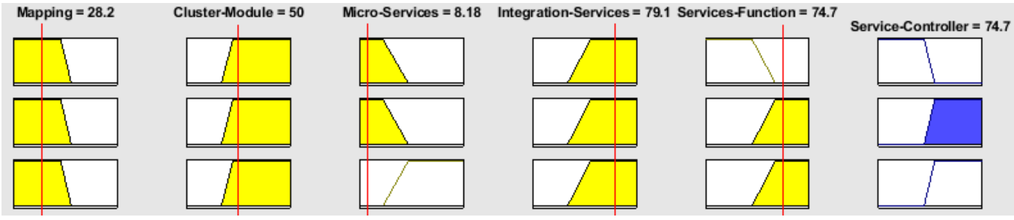


Figure 10(b): Lookup graph of SCo (Yes)

According to Fig. 10(a), if the mapping ranges are set to "No" and the CM is "Yes", MS is "No", IS is "Yes", and SF is "No", then the DC is "No". Fig. 10(b), however, indicates that when mapping ranges are "No", CM is "Yes", MS is "No", IS is "Yes", and SF is "Yes", the DC is "Yes".

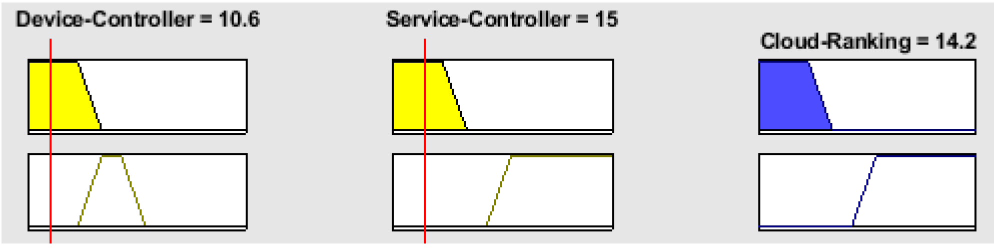


Figure 11(a): Lookup graph of CR (Infra Base)

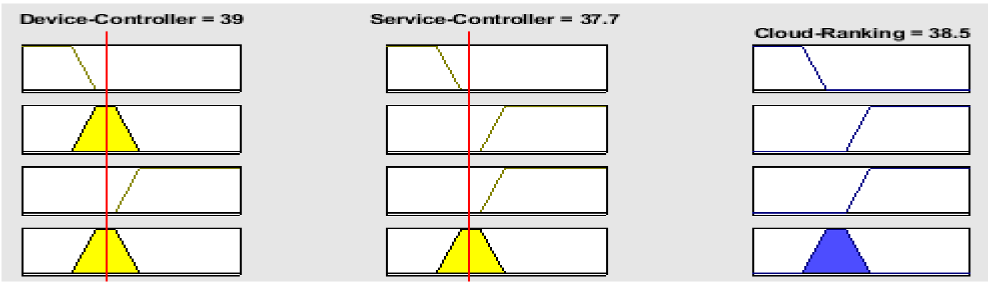


Figure 11(b): Lookup graph of CR (SaaS Base)

In Fig. 11(a), when the ranges of the DC and the SCo are both set to "Low", the CR is "Infra Base". Fig. 11(b) indicates that when the ranges of the DC and the SCo are both set to "Medium", the CR is "SaaS Base".

5 Conclusion

This study highlights the potential benefits of utilizing the extensive internet-accessible cloud infrastructure resources to enhance automation systems. Cloud automation is a comprehensive concept involving processes and tools to streamline cloud computing operations, leading to the optimal system and service output. Organizations can benefit from long-term cost savings by investing in a Cloud Automation Service (CAS) and customizing it as necessary. Implementing cloud automation to manage cloud resources can provide a uniform set of predictable processes and policies that can quickly adapt to changing customer demands, increasing system flexibility and versatility. Furthermore, cloud automation can prevent security vulnerabilities when teams rely heavily on error-prone manual workflows.

This study proposes a hierarchical Mamdani fuzzy expert system that can effectively handle quantitative and qualitative decision criteria in a business context. The system can effectively address uncertainties associated with multi-criteria decision-making through fuzzy logic. The proposed system has demonstrated that fuzzy expert systems can be used to handle complex decision-making situations where data uncertainty is present effectively.

References

[1] M. Armbrust, A. Fox, D. Patterson, N. Lanham, B. Trushkowsky J., and Oh, H. "Scads: scale-independent storage for social computing applications," *arXiv preprint arXiv, vol.9,* no. 3, pp. 909-919, 2011.

[2] P. Bodík, R. Griffith, C. Sutton, A. Fox, M. I. Jordan M. I., and Patterson, D. A., "Statistical machine learning makes automatic control practical for internet datacenters," *HotCloud,* vol. 9, pp. 12-18, 2013.

[3] X. Dutreilh, S. Kirgizov, O. Melekhova, J. Malenfant, N. Rivierre and Truck, I. "Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow," *The Seventh International Conference on Autonomic and Autonomous Systems,* USA, pp. 67-74, *2011.*

[4] Z. Fang, "Resource management on cloud systems with machine learning," MS Dissertation, Universität Politècnica de Catalunya, 20112.

[5] N. Jain, I. Menache and O. Shamir, "On-demand or spot learning-based resource allocation for delay-tolerant batch computing," *Microsoft Research,* vol. 10, no. 5, pp. 1-14, 2012.

[6] E. Kalyvianaki, T. Charalambous and S. Hand, "Self-adaptive and self-configured cpu resource provisioning for virtualized servers using kalman filters," In *Proceedings of the 6th International Conference on Autonomic Computing*, New York, United States, pp. 117-126, 2012.

[7] J. Rao, X. Bu, C. Z. Xu, L. Wang and G. Yin, "Vconf: a reinforcement learning approach to virtual machines auto-configuration," In *Proceedings of the 6th International Conference on Autonomic Computing*, New York, United States, pp. 137-146, 2013.

[8] P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu C., and Hacigümüş, H., "Intelligent management of virtualized resources for database systems in cloud environment," In *IEEE 27th International Conference on Data Engineering*, DC, United States, pp. 87-98, *2011.*

[9] Khan, A. I., Kazmi, S. A. R., Atta, A., Mushtaq, M. F., Idrees, M., Fakir, I., ... & Qasim, A. (2021). Intelligent cloud-based load balancing system empowered with fuzzy logic. Computers, Materials and Continua, 67(1), 519-528.

[10] Tariq, M. I., Tayyaba, S., Ali Mian, N., Sarfraz, M. S., Hussain, A., Imran, M., ... & Paraschiv, N. (2020). An analysis of the application of fuzzy logic in cloud computing. Journal of Intelligent & Fuzzy Systems, 38(5), 5933-5947..

[11] Khan, W. A., Abbas, S., Khan, M. A., Qazi, W. M., & Khan, M. S. (2020). Intelligent task planner for cloud robotics using level of attention empowered with fuzzy system. SN Applied Sciences, 2, 1-13..

[12] S. M. Muneer, M. B. Alvi and M. A. Rasool, "Genetic algorithm based intelligent system for estate value estimation, *International Journal of Computational and Innovative Sciences*, vol. 1, no. 1, pp. 25-38, 2020.

[13] Mohiyuddin, A., Javed, A. R., Chakraborty, C., Rizwan, M., Shabbir, M., & Nebhen, J. (2022). Secure cloud storage for medical IoT data using adaptive neuro-fuzzy inference system. International Journal of Fuzzy Systems, 24(2), 1203-1215.

[14] Anand, A., & Singh, A. K. (2021). Cloud based secure watermarking using IWT-Schur-RSVD with fuzzy inference system for smart healthcare applications. Sustainable Cities and Society, 75, 103398.

[15] Bhardwaj, K. K., Banyal, S., Sharma, D. K., & Al-Numay, W. (2022). Internet of things based smart city design using fog computing and fuzzy logic. Sustainable Cities and Society, 79, 103712.

[16] Chopra, S., Dhiman, G., Sharma, A., Shabaz, M., Shukla, P., & Arora, M. (2021). Taxonomy of adaptive neuro-fuzzy inference system in modern engineering sciences. Computational Intelligence and Neuroscience, 2021.

[17] Li, X., Li, H., Sun, B., & Wang, F. (2018). Assessing information security risk for an evolving smart city based on fuzzy and grey FMEA. Journal of Intelligent & Fuzzy Systems, 34(4), 2491-2501.

[18] Moayedi, H., & Le Van, B. (2022). Feasibility of harris hawks optimization in combination with fuzzy inference system predicting heating load energy inside buildings. Energies, 15(23), 9187.

[19] Khan, M. A., Abbas, S., Atta, A., Ditta, A., Alquhayz, H., Khan, M. F., & Naqvi, R. A. (2020). Intelligent Cloud Based Heart Disease Prediction System Empowered with Supervised Machine Learning. Computers, Materials & Continua, 65(1).