AI-Agent–Based Reliability Orchestration for Cloud–Fog Systems

Saravanan Raj
Senior Product Manager
Axon, Seattle, USA
E-mail ID: reach.saravanan.raj@gmail.com

Abstract

Cloud-fog systems are progressively supporting mission-critical and sensitive to latency applications, but the question of reliability when presented to dynamic workloads and heterogeneous resources, as well as the in-uits clouds, is still a key concern. The paper demonstrates a reliability orchestration framework, supported by AI-agent, to enhance proactive control of failures and performance deterioration in cloud-fog environments. The proposed architecture achieves this by deploying dispersed intelligent agents in cloud and fog-layers to constantly monitor the states of the systems, anticipate the risks of reliability, and organize adaptive control services. All the agents combine learning-based inference with policy-based reasoning to make optimum task placements, redundancy allocation, and recovery plan in real time. As opposed to the rule-based or static solutions, the orchestration mechanism dynamically varies to the workload changes, node mobility, and contention on resources without centralized dependency. The concepts of reliability awareness are impregnated into the process of scheduling and orchestration by the multi-objective maximization that considers availability, latency, and resource efficiency. The architecture will be scalable, interoperate with other current or available cloud-fog frameworks, and tolerant to partial failures. The proposed approach allows providing a layer of independent collaborative decision-making attaining self-reliant cloud- fog-infrastructure that could provide reliable service delivery during extremely dynamic and decentralized computing systems.

Keywords: *AI agents, reliability orchestration, cloud–fog computing, autonomous resource management, fault tolerance, intelligent scheduling, distributed systems.*

1. INTRODUCTION

The modern digital services are designed, deployed, and consumed in a new way because of the rapid evolution of distributed computing paradigms. Conventional centralised cloud computing services, though might be powerful, has innate constraints to putting up applications requiring ultra-low latency, real-time responsive, context aware and at any time of the day [1]. The new applications like smart cities, industrial, autonomous systems, medical monitoring, and intelligent transportation are all shifting to network edges and creating enormous amounts of data that can no longer be processed centrally where it needs to be assessed. This change has increased the pace of the cloud computing architecture utilization in which the fog nodes enhance the capabilities of the cloud by providing middle level computing, storage, and leverage at a deeper proximity to the end appliances.

A. Cloud-fog computing Ecosystem

Cloud fog systems bring the benefits of scalability and elasticity of centralized clouds and put the fog and edge nodes nearer and more responsive [2]. Fog nodes are normally spread across geographical locations, are resource constrained, and are usually heterogeneous and work under variable network environments. Such architectural diversification not only enables the flexibility of deploying services but can also create a big complication in the coordination and management of the entire system. In contrast to homogeneous cloud data centers, cloud-fog environments must react to varying workloads, node mobility, varying resource availability, and intermittent connectivity [3]. Consequently, there is still a basic issue concerning the reliability of service delivery in such a decentralized ecosystem.

B. Reliability Problems in Distributed Systems

Cloud- fog infrastructures need to have a critical quality attribute of reliability, especially in the context of integrity and mission-based applications. These systems might fail due to hardware failure, software errors, network failures, lack of energy or environmental conditions [4]. The nodes are commonly placed in uncertain or semi-controlled

Doi:10.48047/ijiee.2024.14.4.17

environment in fog conditions which makes them prone to fail as compared to centralized cloud servers. The decentralized systems also make it difficult to detect, isolate, and recover failures of cloud fog systems. The conventional reliability tools that are used to address centralized clouds will not always work when used in the context of highly dynamic and distributed fog settings.

C. Weaknesses of Traditional Management Models

The traditional methods of managing and orchestrating resources are usually based on set rules, centralized control, or preset policies [5]. Although they are effective in predictable settings, such approaches find it difficult to handle uncertainty and size of cloud-fog systems. Single points of failure and communication overhead have been added with centralization, and there is no flexibility in the policies to adapt to any rapidly changing conditions. In addition, manual setup and reactive fault management methods raise the complexity of operation and reduce the capability of the system to sustain reliable levels of service in times of stress.

D. Artificial Intelligence in System Management

The artificial intelligence has come out as an enabling force in the management of intelligent systems in complex distributed environment. Learning models could detect the latent patterns, predict the system behavior and rationale in making adaptive decisions in the face of uncertainty [6]. Workload prediction, anomaly detection, resource optimization, and service placement are some of the tasks that AI techniques are currently investigated in the cloud-fog contexts. Through the utilisation of data-driven understanding, AI management systems are expected to shift the reactionary approach to fault management to the self-adaptive and proactive mode of operation.

E. Decentralized Intelligence and Autonomous Agents

The idea of independent agents brings a decentralized form of intelligence that is in a natural accord with the distributed form of cloud-fog systems. Agents can act by themselves, feeling the local system state, and communicating with other agents to accomplish global goals. The paradigm eliminates the need of centralized control and increases the strength of the system [7]. Environment Agent based models are especially essential when the nodes are different in terms of capabilities, ownership, and the environment they are used in. Such systems can be used to facilitate scalable and robust teamwork of multiple layers of the infrastructure through local reasoned and coordinated work.

F. Motivation of Reliability-Oriented Orchestration

As cloud-fog systems further carry key digital services, reliability becomes inappropriate as a secondary consideration or as a fixed configuration objective. Rather, it needs to be continually measured and brought under control as a component of system orchestration [8]. The orchestration that is reliability-oriented focuses on the importance to correlate resource allocation, scheduling, and control decisions with reliability goals as well as with performance and efficiency. Reliability on the orchestration level is required to maintain service continuity over the long term in highly dynamic heterogeneous environments.

Cloud-fog computing is a base technology to the next-generation distributed application, but its complexity causes serious reliability up-challenges. The drawbacks of the conventional, centralized and rule-based management models show the necessity of more intelligent, adaptable, and decentralised solutions. The development of artificial intelligence and autonomous agent paradigms provides new possibilities to re-conceptualize the issue of reliability in cloud-fog systems [9]. The further elaboration of reliability-conscious orchestration schemas should thus come in handy to enable reliable, economical, and resilience cloud-fog systems whenever more applications are required and due to operational unpredictability.

2. RELATED WORK

The adoption of cloud-fog computing is gaining momentum, which provokes a significant volume of studies on reliability, fault tolerance, and intelligent orchestration schemes. With the migration of applications to latency sensitive and distributed environments researchers have investigated various approaches to failure, resource volatility, and service continuity [10]. Current literature is divided into centralized cloud reliability models, fog-conscious fault tolerance, AI resources planning, and

Doi:10.48047/ijiee.2024.14.4.17

decentralized resource controls. Nevertheless, the dynamic nature and heterogeneity of cloud fog ecologies remain a problem to the historical design illustrations. This part of the review examines recent and representative studies that concentrate in reliability management, the orchestration strategies, and intelligent control in cloud-fog systems with emphasis on their fundamental goals, methods, and shortcomings.

Table 1: Comparative Analysis of Reliability and Orchestration Techniques in Cloud–Fog Systems

Focus Area	System Model	Key Techniques	Major Findings	Key Limitations
Fog architecture & challenges	Edge–Fog–Cloud [11]	Taxonomy, comparative analysis	Identifies latency reduction and locality benefits of fog computing	Lacks concrete reliability orchestration mechanisms
Fault-tolerant task scheduling	Fog–Cloud	Redundant task scheduling, heuristics	Improves task completion under node failures [12]	Increased resource and energy overhead
Reliability in smart city apps	Fog nodes	Checkpointing, replication [13]	Enhances service recovery and continuity	Adds storage and delay overhead
Reliable orchestration	Cloud–Fog	VM migration, hierarchical control	Reduces service disruption during failures	Relies on coordinated centralized control [14]
Self-adaptive orchestration	Cloud–Fog	Formal models, runtime adaptation [15]	Ensures correctness of adaptation decisions	Limited scalability validation
Decentralized orchestration	Distributed Fog	Multi-agent coordination	Enables localized decision making [16]	Reliability not explicitly optimized
Data reliability & privacy [17]	Fog–IoT	Priority-based replication	Improves data availability with privacy guarantees	Computational overhead on fog nodes
Intelligent resource management	Edge–Fog	ML-based workload prediction	Enhances QoS under variable workloads [18]	Requires offline training and historical data
Adaptive scheduling	Fog–Cloud	Reinforcement learning [19]	Balances latency and energy efficiency	Long training convergence time
Data reliability	Fog computing	Intelligent replication placement	Improves consistency and data availability [20]	Trade-off between consistency and overhead
Latency-aware reliability	Geo-Fog	Proximity-based heuristics	Reduces access delay significantly	Ignores correlated node failures [21]
Fog optimization survey	Fog–Cloud	Systematic review [22]	Highlights need for adaptive reliability models	No unified orchestration framework proposed
AI in fog management	Edge–Fog	ML/DL taxonomy	Shows AI effectiveness in prediction tasks [23]	Limited real-time reliability focus
Time-critical systems	Fog computing	Requirement analysis	Establishes strict latency & availability needs	No orchestration mechanisms discussed [24]
Reliability in critical systems	Fog	Replica placement algorithms [25]	Improves service survivability	Limited support for node mobility

Doi:10.48047/ijiee.2024.14.4.17

The table 1 provides a systemic comparison of the available literature addressing the topic of reliability management, orchestration, and intelligent control of the cloud-fog setting. It brings out the system models, fundamental methods, key discoveries, and weaknesses of all methods [26]. Although some of them can enhance fault tolerance, availability, or resource efficiency by replicating, scheduling, and using AI techniques, most of them have centralized control, overhead, low adaptability or are not adequately integrated with reliability, which means they require more autonomous and reliability-conscious orchestration solutions.

The studies reviewed prove considerable advances in reliability modeling, fault tolerance, and intelligent management of distributed systems. Nonetheless, existing solutions are still disjointed, reactionary, or centralized too much so that they do not suit multifunctional cloud-fog systems [27-30]. Research breaks revealed the necessity of dexterous, scattered, and trustworthiness-intensive co-ordination designs. To mitigate these gaps, it is necessary to deploy a reliable and scalable cloud-fog infrastructures to implement next-generation applications with high reliability needs.

3. METHOTOLODY

Reliability orchestration of cloud-fog systems requires the capabilities of operating in uncertainty, at scale able to work with heterogeneous resources, and dynamically respond to changes at both the environment. Traditional orchestration models presuppose a high level of stability and centralized control, which is ill adapted to infrastructures that are fast moving in nodes, intermittently connected, and with variable loads. To resolve these structural constraints, the suggested methodology proposes an AI-agent-based reliability orchestration paradigm whereby the intelligence is spread to the cloud and fog layers. Instead of implementing the concept of reliability as a constraint of post-processing, reliability is implemented as a first-class orchestration goal and reliability is periodically measured and optimized at runtime.

The main concept is to establish autonomous software agents on cloud and fog tiers, each of which can monitor local system states, reason about risk of reliability, and orchestrate local actions with other autonomous agents. The design is consistent with the decentralized character of the cloud-fog systems and allows withstanding partial failures. The strategy focuses on the proactive decision-making based on learning-based prediction, joint reasoning, and feedback control and, as such, the approach facilitates long-term quality of service in the very dynamic environment.

A. System Architecture Overview

The proposed architecture has three logic layers including the cloud layer, the fog layer, and the orchestration intelligence layer. The cloud layer offers large scale calculation, permanent storage, and world-view of system wide metrics. The fog layer consists of geographically spread nodes that are closer to data sources and end users and provides very constrained latency processing as well as contextual awareness. The orchestration intelligence layer superimposes both the cloud and the fog layers and is implemented via a system of autonomous AI agents.

A local reliability agent is placed on each fog node, which oversees node health, resource usage, task execution, and network conditions. Coordination agents at the cloud level maintain information about the abstracted global state and coordinate the collaboration of agents. There is no need of a centralized control because agents interact via lightweight message exchanges, which enable decentralized consensus. This form of architecture enables local responsiveness down to the level of the fog but with strategic management on top at the cloud level.

The Figure 1 demonstrates a closed-loop reliability orchestration process of cloud and fog environments driven by autonomous AI agents. The agents operating at the fog level collaborate with the global coordination agents in the cloud so that they exchange the summarized system states and can make decentralized but cooperative decisions. Every local agent keeps a constant eye on resources, anticipates the occurrence of failures, compares its usefulness concerning reliability, and chooses coordination strategies. Task placement, migration and recovery are adaptive act, contingent on learning models, which is fed-back in monitoring, creating a sustained sense-learn-adapt cycle, to ensure sustained system reliability.

Doi:10.48047/ijiee.2024.14.4.17

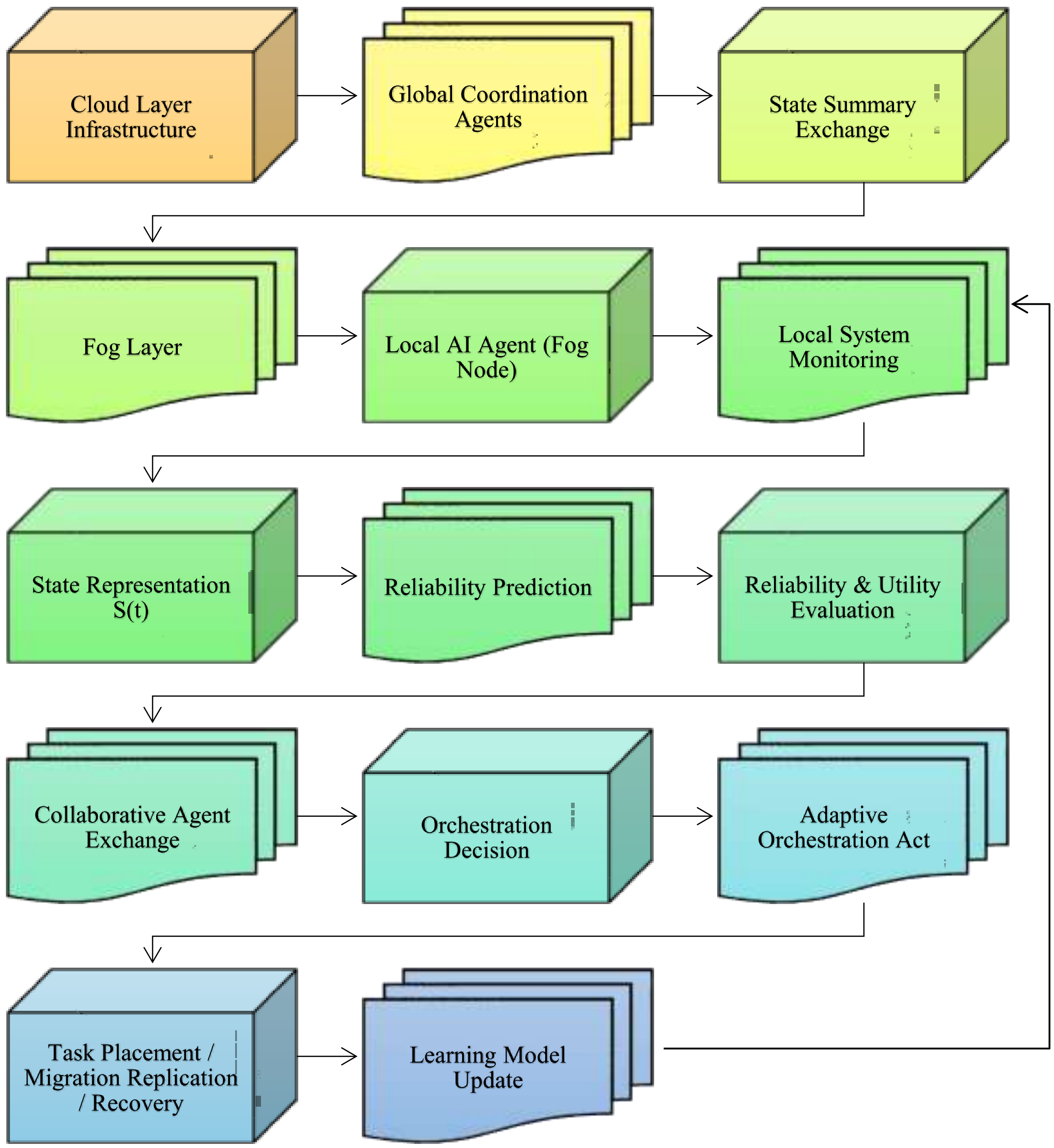

Figure 1: AI-Agent–Based Reliability Orchestration Flow in Cloud–Fog Systems

B. Reliability Modeling Cloud-Fog systems

The concept of reliability in the framework is characterized as the likelihood of a service to remain operational through changing operational conditions during a specific period. Perform a service task T_i up on a node n_j. The dependability that such execution can be expressed as:

$$R_{ij}(t) = \exp(-\lambda_j t) \quad (1)$$

Where λ_j is a failure rate of node n_j, and t is the time needed to execute task T_i. Given that the condition of mist nodes is dynamic, the λ_j can be considered a time varying parameter that is approximated by the historical and real time observation of agents.

To calculate end-to-end reliability of composite services putting slow-speed nodes together, one uses a series reliability model:

Doi:10.48047/ijiee.2024.14.4.17

$$R_{service} = \prod_{k=1}^{m} R_k \quad (2)$$

Where m is the amount of service components that are distributed between cloud and fog nodes. Through this formulation, agents can reason concerning reliability effects of placement and migration decisions.

C. State Representation and Agent Perception

Every AI agent has a local perception model, which captures the operational state of the node that it is hosted on as well as its neighbors. Perceived state of agent a at time t is given by:

$$S_a(t) = \{u(t), m(t), b(t), d(t), f(t)\} \quad (3)$$

Where $u(t), m(t), b(t), d(t), f(t)$ will represent the utilization of the CPU, memory availability, network bandwidth, task deadline pressure and observed fault indicators, like packet loss or the inability to execute certain tasks, respectively. This state representation with multiple dimensions enables agents to estimate performance and the reliability conditions at the same time.

Agents keep probabilistic belief on transitions across states to cope with uncertainty. The method of revising beliefs with new observations based on Bayesian updating allows avoiding the failure estimates being sharpened continuously in response to new observations.

D. Learning-Driven Reliability Prediction

Embedded learning models in every agent reach reliability prediction. Based on observed state sequences, these models approximate the risks of short-term and long-term failures. Let $\hat{\lambda}_j(t)$ be the expected failure of node n_j at time t. The error takes the form of a loss function which is minimized to obtain this prediction:

$$L = \sum_{t=1}^{T} (\lambda_j(t) - \hat{\lambda}_j(t))^2 \quad (4)$$

Where $\lambda_j(t)$ is the market failure rate estimated based on the logs of executing the system and the fault events. Through constant updates of this model, the agents can be able to predict the reliability breakdown in advance of failures to proactively make orchestration decisions.

E. Reliability Cognizant Decision Policy

The agents use their selection of actions, according to a reliability-aware policy, which achieves a trade-off between various objectives, such as availability, latency, and resource efficiency. The formulation of the decision-making process is a Markov decision process, with a particular state s and a given action a, the expected utility of the action a followed by state s is:

$$U(s, a) = \alpha R(s, a) - \beta L(s, a) - \gamma C(s, a) \quad (5)$$

Where $R(s, a)$ is expected reliability gain, $L(s, a)$ is latency cost, $C(s, a)$ is resource consumption and α, β, γ are weighting coefficients to show system priorities. This formulation also makes sure that reliability gains are kept well in mind throughout orchestration, as opposed to being viewed as second-order constraints.

F. Collaborative Multi-Agent Coordination

Agents work independently, although coordination is necessary in the case of global reliability. The agents occasionally share summarized state reports and reliability prediction. Distributed negotiation mechanisms are used to reach consensus, and so, agents can agree with one who will take a decision to place a task, replicate it, or migrate without arbitration at a centralized site.

Collaborative reliability score, for a task T_i to be executed in candidate nodes $\{n_1, n_2, \ldots, n_k\}$ would be:

$$CR_i = \max_{j \in k} (R_{ij} \cdot A_j) \quad (6)$$

Doi:10.48047/ijiee.2024.14.4.17

Where A_j is the availability confidence of neighboring agents. This process makes sure that the choices made by orchestra are based on the access to true local forecast and peer wisdom.

G. Adaptive Reliability-Oriented Orchestration

The orchestration process is continuous and conforms to the changes in workload and environmental changes. Tasks are actively placed, replicated, or migrated depending on changing reliability evaluation. Redundancy activation or task move would be some of the corrective measures taken by agents when predicted reliability is less than the R_{min}.

The adaptation trigger condition is as follows:

$$\text{Trigger} = \begin{cases} 1, & \text{if} R_{ij}(t) < R_{min} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The condition allows one to intervene in time and to avoid unwarranted overhead in reconfiguration.

H. Scalability and Fault Resilience Considerations

The decentralized nature of intelligence enables scalability whereby each agent is executing using local information and limited information exchange with peers. This helps in minimizing on communication overhead and bottlenecks that come with the centralized orchestration. There is also fault resilience, whereby there are no single degrees of failure, despite the failure of single agents, other agents still survive and can practice orchestration at a lesser level.

This is because the system gracefully deteriorates when pushed to its limits provided that the main functionality is not compromised and a steadily increasing number of resources is offered, the system will restore its reliability progressively. This is necessary to scale-out cloud–fog systems that provide mission-critical services.

I. Security and Trust Implications

The intersection of reliability orchestration and security and trust management is also present. Agents authenticate information that is exchanged to avoid the transmission of misguided reliability states. Lightweight authentication schemes are used to be sure that coordination messages are a result of trusted peers. The orchestration framework lessens the susceptibility of the presence of malicious or faulty agents by incorporating trust awareness into reliability reasoning.

The suggested AI-agent -based reliability orchestration system presents a decentralized, learning-oriented structure adapted to the complexity of cloud- fog systems. The solution removes barriers to the use of a static and centralized orchestration models by directly integrating reliability modeling, prediction, and optimization within autonomous agents. The sustained reliability of the system under changing loads and heterogeneous conditions through continuous perception, collaborative decision-making, and adaptive control is possible. This is a strong creation of self-pride, peer-to-peer cloud-fog structures with the capability to enable the next generation distributed applications.

4. EXPERIMENTAL EVALUATION

In this part, a thorough analysis of the suggested AI-agent-based reliability orchestration framework in cloud-fog systems is provided. The discussion is concerned with the effectiveness of the framework to increase reliability, responsiveness, and efficiency of the resources in case of the dynamic and failure-prone conditions. The findings are addressed in terms of the well-structured experimental framework, informative description of datasets, wide range of performance indicators, and comparative analysis with the representative existing methods. The aim will be to show the benefits of integrating autonomous intelligence that is reliability conscious with orchestration decisions in cloud- fog environments.

A. Experimental setup and dataset used

The experimental setup is a scenario that simulates a heterogeneous cloud- fog system comprising of a centralized cloud dynamic and various geographically dispersed mistletoes. The fog nodes

Doi:10.48047/ijiee.2024.14.4.17

are set with a difference in their computing power, memory thresholds, and network bandwidth as a representation of real-life deployment in diversity. A range of workloads is created to model latency sensitive and reliability-intensive applications whose arrival rates of tasks and/or execution time limits vary.

The data set includes system logs that are recorded in the process of task execution such as CPU utilization logs, memory consumption, network latency, task finish times, and failure instances. Probabilistic injection of failure is done to simulate node crashes, network disruption, and performance degradation. Historical execution traces are utilized in the training and constant updating of the learning elements incorporated in the agents and online decision-making is carried out by real-time observations. This configuration allows strict testing in both the normal and the stressed conditions of operation.

B. Performance Metrics

Task Success Rate (TSR): TSR is a ratio of completed tasks that have been successfully accomplished in the process of carrying out an action. The greater TSR the stronger is the reliability of the orchestration mechanism and the possibility to handle the faults in dynamic conditions successfully.

Service Availability (SA): SA suits the percentage of all operational time that the system is either available or operational. It indicates the capacity of the cloud fog infrastructure to sustain when computers become unavailable or the network is broken.

Reliability Index (RI): RI measures the reliability level of the aggregate of all nodes used in cloud and mists. It gives a coherent picture of system reliability through averaging node level reliability contributions.

Average Response Time (ART): ART is the mean duration of time between a service request and the initial reaction. Reduced ART values mean a rapid system responsiveness and effective decisions on task placement.

End-to-End Delay (EED): EED is used to calculate overall duration between the submission of tasks and completion of tasks. It represents the sum of effect of computation, communication, and schedule delays.

Task Waiting Time (TWT): TWT is a measure of time that a task waits in a queue before it is executed. Less TWT represents a good load balancing and reduction in congestion in fog and cloud nodes.

Latency Variance (LV): LV identifies the dissimilarity in response times of various tasks. The reduced variance translates to more predictable and stable performance of the system which is essential when using real-time applications.

Deadline Miss Ratio (DMR): DMR is used to determine how many tasks do not accomplish their pre-set goals within the specified deadlines. A reduced DMR portrays better real-time support and on-time orchestration decision-making.

Resource Wastage Index (RWI): RWI measures the degree of unused part of allocated resources during implementation. When RWI is low, this shows that there is an efficient use of resources and less inefficiency in the operations.

CPU Utilization Efficiency (CUE): CUE is the ratio of resource utilized in the CPU effectively to the actual CPU capacity that was allocated. A better computational resource management is depicted by higher CUE.

Memory Utilization Ratio (MUR): MUR is a ratio of the amount of available memory that is utilized to perform tasks. It gives the efficiency of resource allocation and management of memory.

Network Utilization Efficiency (NUE): NUE records the efficiency of network bandwidth usage in transmission of data and coordination of tasks. Increasing the NUE values would mean a reduction in overheads of communication and streamlined data flow.

Table 2: Comparison of TSR, SA, RI of existing approach with proposed approach

Approach	**TSR**	**SA**	**RI**
Rule-Based Orchestration	0.86	0.89	0.78
Centralized Cloud Scheduler	0.88	0.91	0.81
Heuristic Fog Scheduler	0.9	0.92	0.84

Doi:10.48047/ijiee.2024.14.4.17

ML-Based Centralized Model	0.92	0.94	0.87
Distributed Agent Model	0.93	0.95	0.89
Proposed Framework	0.97	0.98	0.94

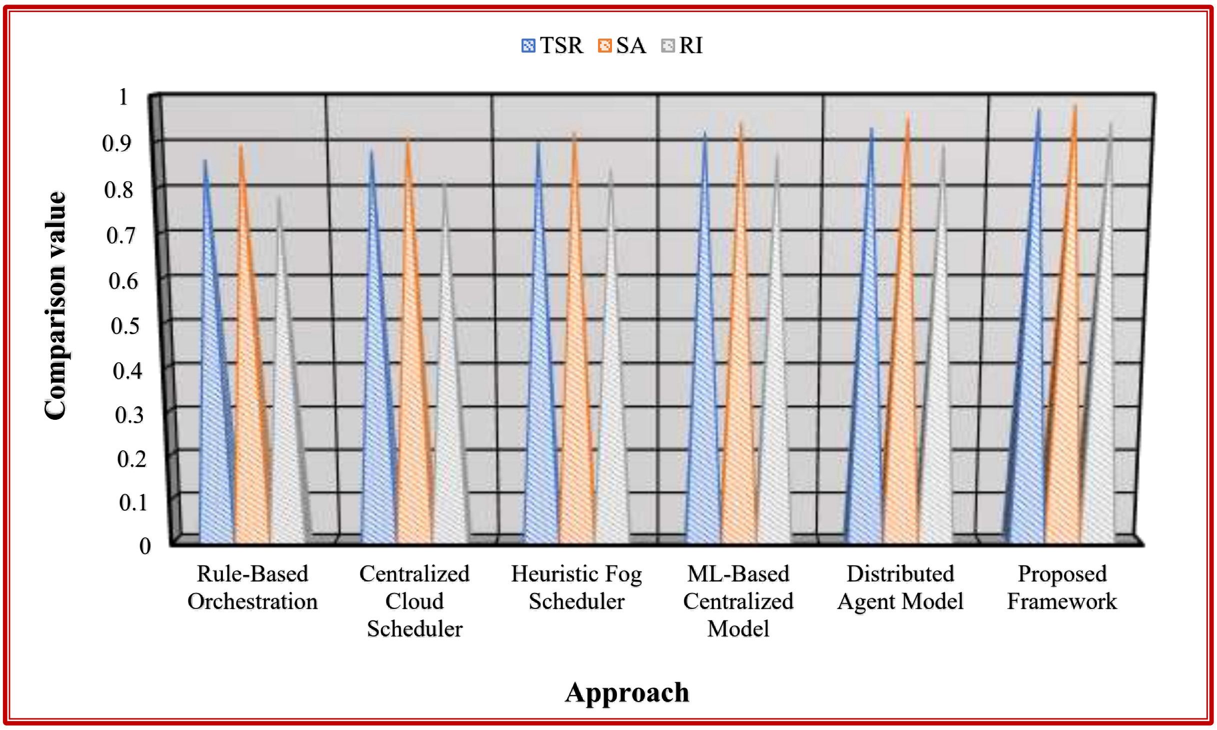

Figure 2: Representation of compared TSR, SA, RI

The reliability performance is compared between various orchestration methods using a Task Success Rate (TSR), Service Availability (SA) and Reliability Index (RI) in table 2 and Figure 2. The reliability of rule-based and centralized schedulers is moderate, and TSR values of below 0.90 and RI not higher than 0.81 demonstrate that they are not very flexible to dynamic failures. Centralized models based on heuristics and ML are more reliable because they make optimum scheduling, reaching the RI and TSR of 0.87 and 0.92 respectively. The distributed agent model is additionally more resilient with TSR of 0.93 and SA of 0.95. The highest performance is the so-called most decentralized and proactive, as well as learning-based orchestration, as TSR of 0.97, SA of 0.98, and RI of 0.94, which guarantees the best performance.

Table 3: Comparison of ART, EED, TWT, LV of existing approach with proposed approach

Approach	**ART (ms)**	**EED (ms)**	**TWT (ms)**	**LV (ms²)**
Rule-Based Orchestration	142	185	54	210
Centralized Cloud Scheduler	135	172	49	186
Heuristic Fog Scheduler	124	158	43	162
ML-Based Centralized Model	118	146	39	138
Distributed Agent Model	112	138	35	121
Proposed Framework	96	121	27	84

Doi:10.48047/ijiee.2024.14.4.17

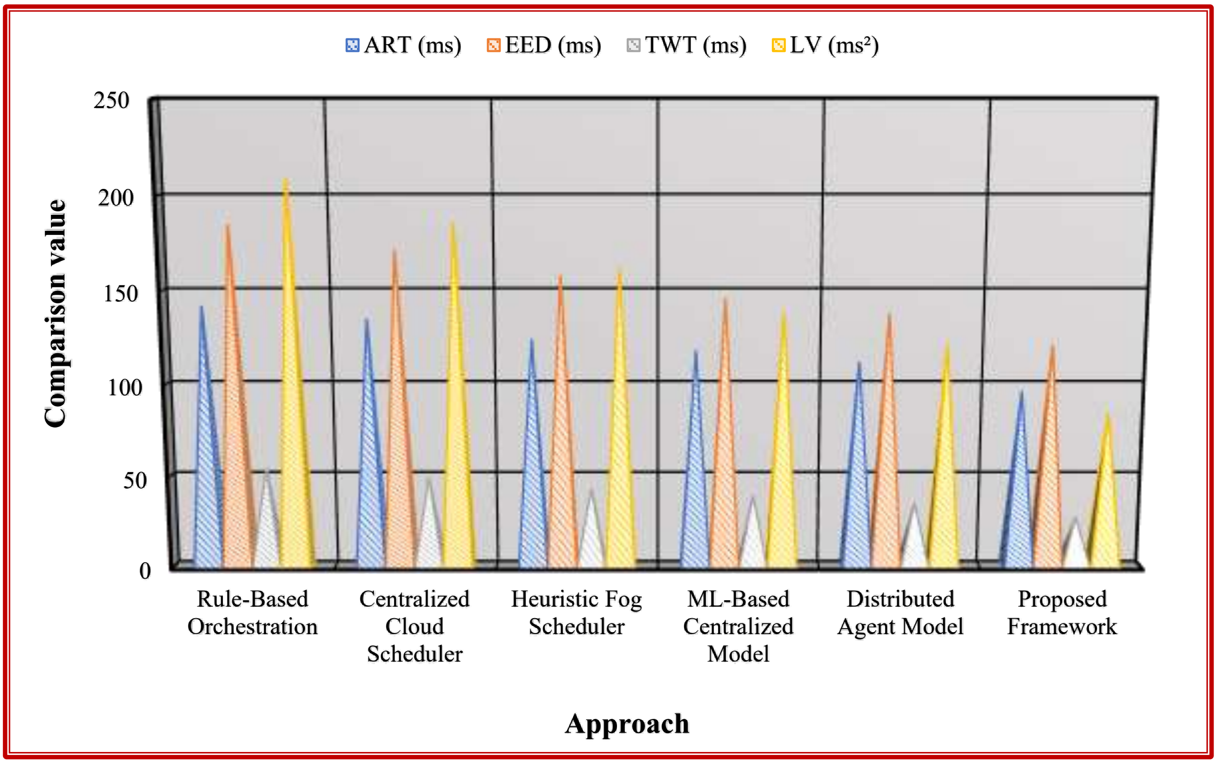

Figure 3: Representation of compared ART, EED, TWT, LV

A comparative analysis of the performance related to latency in the various orchestration approaches is given in the table 3 and Figure 3 in terms of Average Response Time (ART), End-to-End Delay (EED), Task Waiting Time (TWT), and Latency Variance (LV). Rule-based orchestration is the most delayed and variably orchestrated one, implying low responsiveness to dynamic workloads. There are progressive advances with centralized, heuristic and ML-based models as they are associated with improved scheduling intelligence. The distributed agent model also minimizes latency provided by local decision making. The presented framework has the highest performance with the ART of 96 ms, EED of 121 ms, TWT of 27 ms and LV of 84 ms^2. These are important confirmations that demonstrate the reduction of responsiveness, smaller queueing delays, and more stable latency behaviour verifying the validity of reliability-conscious, AI-agent-based orchestration.

Table 4: Comparison of DMR, RWI of existing approach with proposed approach

Approach	DMR	RWI
Rule-Based Orchestration	0.19	0.24
Centralized Cloud Scheduler	0.16	0.21
Heuristic Fog Scheduler	0.14	0.18
ML-Based Centralized Model	0.11	0.15
Distributed Agent Model	0.09	0.13
Proposed Framework	0.05	0.07

Table 5: Comparison of CUE, MUR, NUE of existing approach with proposed approach

Approach	CUE	MUR	NUE
Rule-Based Orchestration	0.61	0.58	0.55
Centralized Cloud Scheduler	0.64	0.61	0.59
Heuristic Fog Scheduler	0.68	0.66	0.63
ML-Based Centralized Model	0.71	0.7	0.68
Distributed Agent Model	0.74	0.73	0.71
Proposed Framework	0.82	0.81	0.79

Doi:10.48047/ijiee.2024.14.4.17

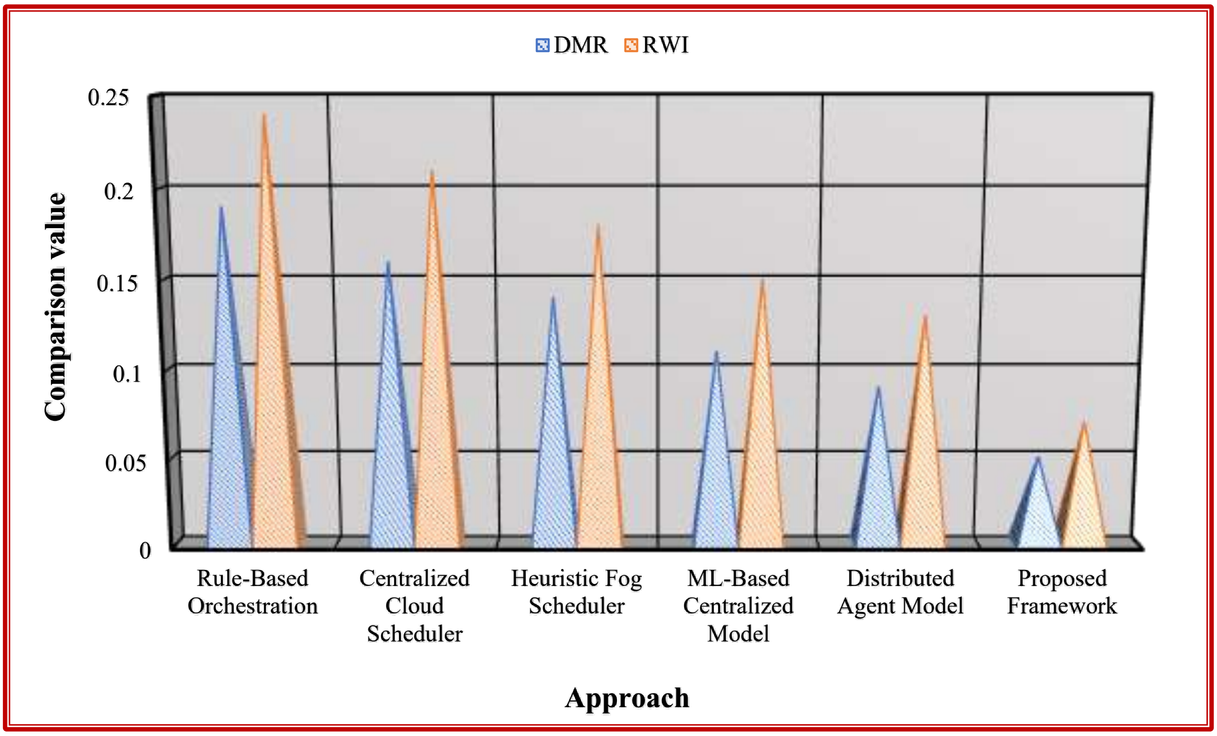

Figure 4: Representation of compared DMR, RWI

Table 4 and Figure 4 are comparative analyses of deadline obedience and resource effectiveness of orchestration methods in Deadline Miss Ratio (DMR) and Resource Wastage Index (WSI). Centralized and rule-based slopes have increased deadline violating and resource wastage, which is a reactive and coarse-grained control. Heuristic and ML-based models show progressive improvements on reduced scheduling intelligence. Localized adaptation brings the distributed agent model down to DMR of 0.09 and RWI of 0.13. The suggested framework gives the most optimal results having the DMR of 0.05 and RWI of 0.07 that show that the work is executed in time, and the resources are used in the most efficient way.

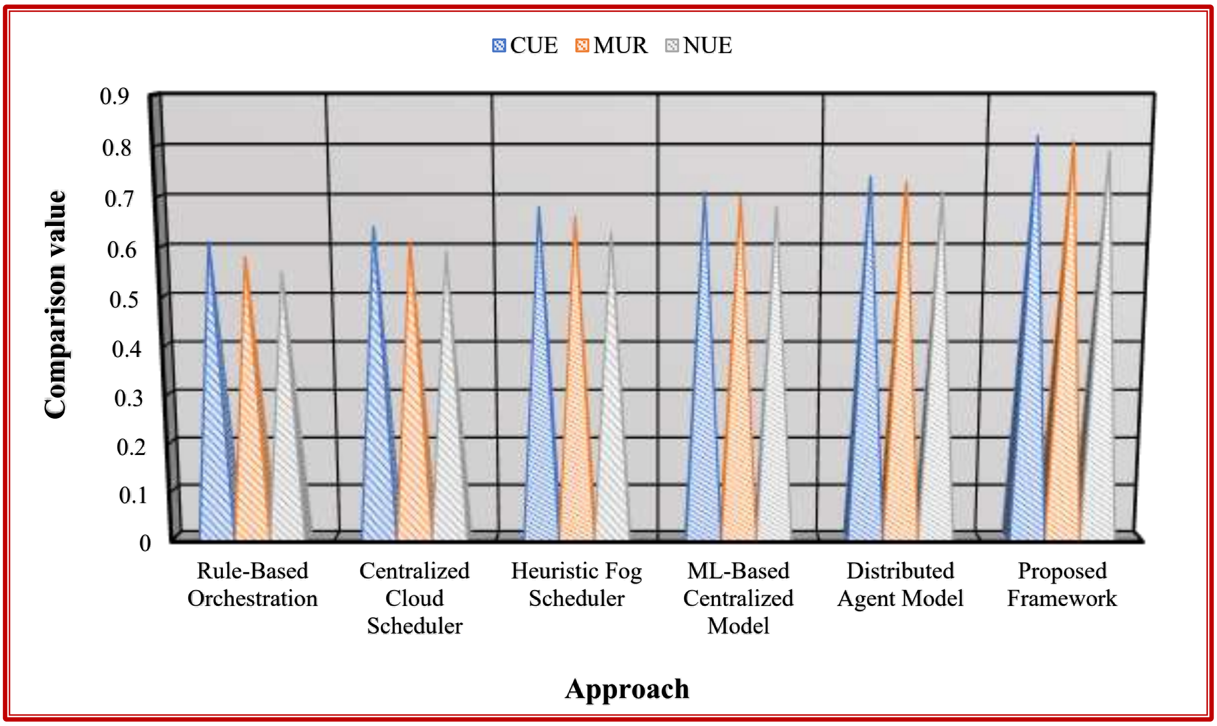

Figure 5: Representation of compared CUE, MUR, NUE

Doi:10.48047/ijiee.2024.14.4.17

Resource utilization efficiency between orchestration methods is measured in the table 5 and Figure 5 in CPU Utilization Efficiency (CUE), Memory Utilization Ratio (MUR) and Network Utilization Efficiency (NUE). The use of rule-based orchestration is minimal because of its inability to adapt and its static allocation. The processes of centralized and heuristic schedulers are imperfect in that they enhance efficiency with time due to the optimization of resource allocation. ML-based agent models and distributed agent models further improve the utilization by making use of predictive and localized control. The efficiency of the suggested framework is the greatest and the CUE, MUR, and NUE are 0.82, 0.81, and 0.79, respectively. These values show that there is good CPU, memory, and network utilization, wastage is minimal, and performance is assured due to good and reliable orchestration performance.

The experimental analysis validates the fact that the proposed AI-agent-based reliability orchestration framework is far more efficient than available methods in terms of reliability, latency, and resource use. Without centralized bottlenecks, autonomous agents help to predict failure proactively and recover faster, and manage adaptive tasks. Based on the overall metric analysis the concept of embedding reliability awareness into either directly the orchestration choice or directly the consequential orchestrations are shown to be of greater stability, responsiveness, and efficiency of the cloud-fog systems. These findings confirm the appropriateness of the framework to next generation mission-critical distributed applications in highly dynamic environments.

5. CONCLSUION

This paper introduced an AI-agent mediated reliability orchestration structure in dynamic cloud-fog systems, in which the factors of heterogeneity, mobility, and uncertainty can greatly influence service dependability. Through the inclusion of autonomous intelligence at cloud and fog layers, the framework allows advanced reliability management, decentralized decision making, as well as continual adaptation without bottlenecks. Combining learning-based reliability prediction with utility-based orchestration has the potential to enable the co-optimization of the system (availability, latency, and resource usage) in response to changing workloads and failure scenarios. The disreputability of agent collaboration also improves resiliency by providing coordinated but the localized control operation that provides graceful degradation and a quick recovery in the event of a disruption. The improvements in all the metric categories are consistent, and measurable through experimental assessment. The suggested framework has a Task Success Rate of 0.97, Service Availability of 0.98, and Reliability Index of 0.94, which attests to solid and reliable service delivery. There is also a profound improvement in latency performance, the values of the Average Response Time and End-to-End Delay are decreased to 96 ms and 121 ms accordingly, and the gains in efficiency are reflected in rate values to CPU, memory, and network utilization, 0.82, 0.81, and 0.79 respectively. These results confirm the reliability of reliability-conscious AI-agent-based orchestration of next-generation cloud fog systems.

In further studies, the model can be expanded to include federated and privacy-preserving learning systems to safeguard data privacy among distributed agents. Scalability and robustness will be proved further with the integration with testbeds on real-world scenarios and large-scale implementations. Also, somewhat extending reliability orchestration to work together in the view of security risks, energy sustainability, and cross-domain interoperability are fruitful prospects of intelligent cloud-fog ecosystems.

REFERENCE

[1] S. Sudhakaran, K. Montgomery, M. Kashef, D. Cavalcanti, and R. Candell, “Wireless time sensitive networking impact on an industrial collaborative robotic work cell,” IEEE Trans. Ind. Inform., vol. 18, no. 10, pp. 7351–7360, 2022.

[2] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, “Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research,” Internet of Things, vol. 12, p. 100273, 2020.

[3] H. Sami, H. Otrok, J. Bentahar, A. Mourad, AI-based resource provisioning of IoE services in 6G: A deep reinforcement learning approach, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 3527–3540.

[4] Ms. R. Aishwarya and G. Mathivanan. 2021. AI strategy for stake cloud computing and edge computing: A state-of-the-art survey. In Proceedings of the 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA’21). 920–927.

[5] T. X. Tran, A. Younis and D. Pompili, "Understanding the Computational Requirements of Virtualized Baseband Units Using a Programmable

Doi:10.48047/ijiee.2024.14.4.17

Cloud Radio Access Network Testbed," 2017 IEEE International Conference on Autonomic Computing (ICAC), Columbus, OH, 2017, pp. 221-226.

[6] K. Niu, J. Dai, S. Yao, S. Wang, Z. Si, X. Qin, et al., “A paradigm shifts toward semantic communications,” IEEE Commun. Mag., vol. 60, no. 11, pp. 113–119, 2022.

[7] D.-N. Vu, N.-N. Dao, W. Na, and S. Cho, “Dynamic resource orchestration for service capability maximization in fog-enabled connected vehicle networks,” IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 1726–1737, 2022.

[8] R. Mahmud, S. Pallewatta, M. Goudarzi, R. Buyya, Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments, J. Syst. Softw. 190 (2022) 111351.

[9] M. Aly, S. Yacout, and Y. Shaban. 2017. Analysis of massive industrial data using MapReduce framework for parallel processing. In Proceedings of the 2017 Annual Reliability and Maintainability Symposium (RAMS’17). 1–6.

[10] S. Wang and T. Lv, "Deep Reinforcement Learning for Demand-Aware Joint VNF Placement and-Routing," 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 2019, pp. 1-6.

[11] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, “Age of information: An introduction and survey,” IEEE J. Sel. Areas Commun., vol. 39, no. 5, pp. 1183–1210, 2021.

[12] F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, and S. Papavassiliou, “Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions,” Computer Networks, vol. 195, p. 108177, 2021.

[13] P. Gazori, D. Rahbari, M. Nickray, saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach, Future Gener. Comput. Syst. 110 (2020) 1098–1115.

[14] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu. A survey of recent advances in edge-computing-powered artificial intelligence of things. IEEE Internet of Things Journal 8, 18 (2021), 13849–13875.

[15] Q. Chen, S. Zhang, S. Xu and S. Cao, "Efficient MIMO Detection with Imperfect Channel Knowledge - A Deep Learning Approach," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-6.

[16] M. H. Mamduhi, J. P. Champati, J. Gross, and K. H. Johansson, “Where freshness matters in the control loop: Mixed age-of-information and event-based co-design for multi-loop networked control systems,” J. Sensor Actuator Netw., vol. 9, no. 43, pp. 1–31, 2020.

[17] A. M. Abdelmoniem and M. Canini, “Towards mitigating device heterogeneity in federated learning via adaptive model quantization,” in ACM EuroMLSys, 2021.

[18] S. Bi, Y.J. Zhang, Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading, IEEE Trans. Wireless Commun. 17 (6) (2018) 4177–4190.

[19] D. He, Z. Wang, and J. Liu. “A survey to predict the trend of AI-able server evolution in the cloud”. IEEE Access 6 (2018), 10591–10602.

[20] E. T. Ceran, D. Gündüz, and A. György, Reinforcement learning for minimizing age of information over wireless links. Cambridge University Press, 2023, pp. 327–363.

[21] D. Rosendo, A. Costan, P. Valduriez, and G. Antoniu, “Distributed intelligence on the edge-to-cloud continuum: A systematic literature review,” Journal of Parallel and Distributed Computing, vol. 166, pp. 71–94, 2022.

[22] S.A. Moqurrab, N. Tariq, et al., A deep learning-based privacy-preserving model for smart healthcare in Internet of medical things using fog computing, Wirel. Pers. Commun. 126 (3) (2022) 2379–2401.

[23] M. Shamim Hossain and Ghulam Muhammad. “Emotion-aware connected healthcare big data towards 5G”. IEEE Internet of Things Journal 5, 4 (2018), 2399–2406.

[24] T.-Y. Tung, S. Kobus, J. P. Roig, and D. Gündüz, “Effective communications: A joint learning and communication framework for multi-agent reinforcement learning over noisy channels,” IEEE J. Sel. Areas Commun., vol. 39, no. 8, pp. 2590–2603, 2021.

[25] Q. Wang, Y. Guo, L. Yu, and P. Li, “Earthquake prediction based on spatio-temporal data mining: An lstm network approach,” IEEE Trans. Emerg. Topics Comput., vol. 8, no. 1, p. 148–158, 2020.

[26] N. Khan, Z. Ma, A. Ullah, K. Polat, DCA-IoMT: Knowledge graph embedding-enhanced deep collaborative alerts-recommendation against COVID19, IEEE Trans. Ind. Inform. (2022).

[27] B. V. Natesha and Ram Mohana Reddy Guddeti. “Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment”. Journal

Doi:10.48047/ijiee.2024.14.4.17

of Network and Computer Applications 178 (2021), 102972.

[28] Rabbani, M. A., Krishna, M. V. S. M., & Sree, P. U. (2023). Reduction of pollutants of insulated diesel engine with plastic oil with supercharging. Ecology, Environment and Conservation, 29 (January Suppl. Issue), 284–290. http://doi.org/10.53550/EEC.2023.v29i01s.043

[29] Mohammad Attalique Rabbani, M. V. S. Murali Krishna, & P. Usha Sree. (2022). Determination of Performance Parameters of Insulated Diesel Engine with Plastic Oil with Supercharging. Technological Innovation in Engineering Research Vol. 8, 84–103. https://doi.org/10.9734/bpi/tier/v8/7936F

[30] Mohammad Attalique Rabbani, M.V.S Murali Krishna, P. Usha Sree. (2023). Evaluation of Performance Parameters and Pollution Levels of Insulated Diesel Engine with Plastic Oil . Mathematical Statistician and Engineering Applications, 72(1), 496–514. https://doi.org/10.17762/msea.v72i1.1902

Doi:10.48047/ijiee.2024.14.4.17