International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 04(April),2026
Received: 18-02-2026 Accepted: 23-03-2026 Published: 01-04-2026

A Pareto-Front-Based Neural Decision Framework for Power-Efficient IoT Devices

Jaffar Ali S
Dept. of Electronics and Communication Engineering Annamacharya Institute of Technology and Sciences Kadapa, India.
aliaitk.ece@gmail.com

Suchithra C
Dept. of Electronics and Communication Engineering Annamacharya Institute of Technology and Sciences Kadapa, India.
chukkalurusuchithra@gmail.com

Varalakshmi D
Dept. of Electronics and Communication Engineering Annamacharya Institute of Technology and Sciences Kadapa, India.
varalakshmiduttuluri@gmail.com

Yasaswini B
Dept. of Electronics and Communication Engineering Annamacharya Institute of Technology and Sciences Kadapa, India.
bantuboina@gmail.com

Pavan Kumar C
Dept. of Electronics and Communication Engineering Annamacharya Institute of Technology and Sciences Kadapa, India.
chapagallapavankumar@gmail.com

***Abstract*— IoT-based fog computing environments have limited energy resources, changing workloads, and diverse devices, effective power management is a crucial concern. In order to overcome this difficulty, the current approach introduces a framework for adaptive resource and power management decision making based on neural networks. In order to record the real-time load and energy condition of the infrastructure, the system continuously checks important operational data, such as CPU utilization, power consumption, and the number of active containers on each fog node. To produce insightful depictions of system states, these tracked measurements go through a feature extraction procedure. A Multilayer Perceptron (MLP) model is trained using the retrived characteristics to understand the intricate connection between resource usage patterns and the best migration choices. The neural network dynamically assesses if container movement is required based on the learnt behaviour and chooses a suitable destination, like the cloud or another fog node, to balance workloads. This clever migration approach seeks to prolong the battery life of IoT devices, avoid node overcrowding, and minimize excessive power usage. The learning - driven framework provides better adaptability to varying workloads and shifting network conditions as compared to coventional static and rule - based techniques. The approach is appropriate for scalable and sustainable Internet of Things applications since it improves energy efficiency and resource usage in fog computing settings.**

***Keywords*—Internet of Things; Energy Efficiency; Pareto Optimization; Neural Network Decision Making; Fog–Cloud Computing.**

I. INTRODUCTION

A multitude gizmos may now communicate and exchange data via the internet courtesy of the proliferation of the Internet of Things (IoT). By analyzing this data, Organizations can boost their services, but doing so presents privacy and data security vulnerabilities. The demand spanning the amount of digital data has expanded in recent years due to the use of computers, mobile devices, sensors, and smart connected systems [1]. Fast data processing and quick responses are essential for many Internet of Things applications. These requirements might not always be met by traditional cloud solutions, particularly in extensive IoT networks. By processing data closer to the devices, fog computing simplifies in resolving this concern. This bolsters IoT system performance and streamlines a scenario [2]. Many hip IoT applications entail fast data mining. It facilitates this by decoding information more committed [3]. The assortment of IoT devices to escalate swiftly, generating a lot of data that
needs to be escalade Fog computing speeds up system performance and lessens bottlenecks by processing data closer to devices[4]. The expansion of IoT devices generates a lot of data that requires good processing. By processing this data close to the network edge, fog computing lowers latency and accelerates system performance [5].Large time-series information gathered from IoT devices may have odd trends. Maintaining system performance necessitate these irregularities [6]. Fog computing uses Pareto optimization to place tasks efficiently by balancing cost, delay, and energy. It gives better results than normal methods, especially in changing IoT environments. IoT applications need efficient deployment in fog computing to handle several objectives like cost, delay, and system reliability. Multi-objective optimization helps in finding better solutions by balancing these factors effectively [14-15-21].

II LITERATURE REVIEW

The authors devised a mobility-aware job delegation and migration strategy for fog computing networks. The technique offloads tasks to fog nodes while accounting for device mobility in order to reduce latency and improve service continuity. The results show superior system performance when compared to traditional offloading tactics [9]. The algorithm's complex optimization processes may increase processing costs even while placement efficiency is improved [10]. Researchers have also zeroed on striking multiple objectives such as cost, latency, and energy consumption. Multi-objective optimization algorithms have been developed to improve reliability, fault tolerance, and energy efficiency in fog computing systems. To promote fog computing systems' resiliency, fault tolerance, and energy efficiency, multi-objective optimization protocols have been conceived [14–16]. In order to optimize resource utilization in IoT environments, recent studies have also looked into load balancing tricks and energy-aware scheduling techniques. The suggested strategy increases energy efficiency, adaptability, and decision accuracy in unpredictable fog-cloud environments [18-21].

III EXISTING SYSTEM

Resource management and container migration are typically managed using conventional techniques in Internet of Things (IoT) devices. To decide when a container needs to be moved from one device to another, these techniques keep

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 04(April),2026
Received: 18-02-2026 Accepted: 23-03-2026 Published: 01-04-2026

an eye on metrics like CPU utilization, battery life, and memory usage [9–13]. Another limitation is that many existing approaches focus on only a single optimization objective, such as task completion time or load balancing [14]. Container migration usually occurs when specific threshold conditions are satisfied. Migration might happen, for instance, when memory usage becomes excessive, battery power falls below 20%, or CPU utilization surpasses 80%.

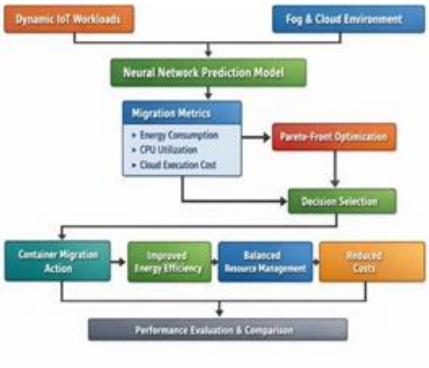


Fig. 1. Existing system

These straight forward rule-based techniques can function well in some systems and are simple to implement [18]. Conventional container migration techniques may lead to drawbacks like unnecessary migrations, excess energy usage, and poor resource use [20]. We know that IoT environments are dynamic and complex traditional static methods, they are unable to effectively adjust to shifting system conditions. Consider multiple factors like energy usage, CPU utilization, migration overhead, and cloud execution cost [21-22]. Hence, IoT systems want adaptive decision-making techniques that can efficiently optimize several goals and dynamically react to system changes [23]. We use advanced methods such as Neural Network-Based Decision Making combined with Pareto-Front Optimization to increase power consumption and resource efficiency in IoT devices [24].

IV PROPOSED SYSTEM

Energy-efficient workload management is crucial in fog-based IoT systems to extend battery-powered devices operational lifetimes while preserving acceptable performance. At present, neural network-based decision-making techniques usually use feature-designed rules or predefined thresholds, like fixed CPU or power usage restrictions, to initiate container transfer. These methods may result in poor choices under dynamic workloads, varied device capabilities, or changing clous cost, in spite of successful in controlled circumstances. The systems' flexibility may be limited by static decision limits and single-objective optimization, especially when several alternative goals like reducing power consumption while managing execution costs need to be taken into account immediately.

Each possible migration stage is evaluated across various goals, such as power consumption, CPU utilization, and cloud execution cost, rather than extracting features or using heuristic decision methods. When no objective can be enhanced without degrading another, Pareto-front optimization is used to find non-dominated migration options. This formulation removes the requirement for manually stated thresholds or weighted collection functions and allows the system to directly record trade-offs among competing objectives. The decision-making flexible and more suitable to the dynamic nature of IoT fog-cloud environments.

Instead of making migration decisions directly, the extended framework uses a neural network model to evaluate the expected results for possible migration acts. The neural network decision making model predicts the effects of moving containers to various locations, such as additional fog nodes or cloud providers, based on current system parameters, such as CPU utilization, power consumption, and the number of active containers. The Pareto-Front analysis, which identifies the set of ideal trade-off solutions, uses these expected outcomes as inputs. A final migration decision is chosen from this Pareto-optimal set employing a policy that strikes a compromise between operational cost and energy efficiency, such as enforcing economic limitations or selecting the lowest-cost solution. The division of prediction and optimization improves the decision process's robustness and explainability.

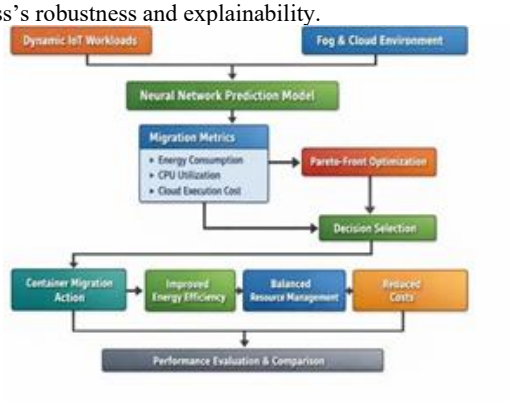


Fig. 2. Proposed system.

The proposed strategy increases adaptability and decision quality in energy-constrained IoT situations by combining Pareto-Front Optimization with Neural Network Decision Making based prediction. The overall approach assures that migration choices take into consideration several goals at once, resulting in better balanced power consumption and less unnecessary cloud utilization. These research results illustrate that Pareto-Optimized decision making offers a more sensible and successful approach to controlling energy usage in fog-based IoT systems.

4.1 Working of Pareto-Front

In this multi-objective space, Pareto-front analysis determines which proposed actions represent non-dominated solutions. If there is no other solution that promotes a minimum number of objectives without negatively impacting any other goal, the solution is said to be non-dominated or Pareto-optimal. In real life, if migrating a container to a cloud provider. Both approaches may be Pareto-optimal, but neither strictly dominates the other because each has advantages in different dimensions. For example, provider A would cut power usage compared to transferring to provider B, but provider A also costs more and has more CPU overhead. On the other hand, provider C is dominated by provider A and can be ignored if switching to provider C results in poorer power consumption, higher costs, and worse CPU use than provider A across every single parameter. By using Pareto Front Optimization with Neural Network decision making decrease the cloud cost, execution time, CPU utilization, and number of active containers. Instead of threshold using the trade-offs in pareto front optimization.

The Pareto-front consists of all non-dominated solutions, forming a frontier in objective space that represents the boundary of achievable performance. Points along this

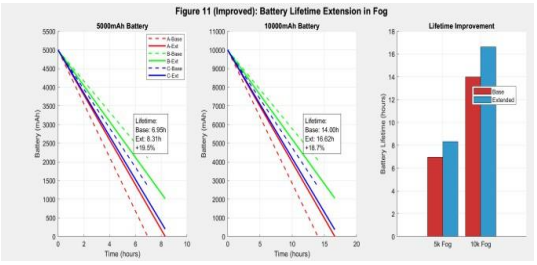
2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 04(April),2026
Received: 18-02-2026 Accepted: 23-03-2026 Published: 01-04-2026

frontier characterizes the fundamental trade-offs imposed by physical constraints, workload characteristics, and infrastructure capabilities. Moving from one point on the Pareto front to another necessarily involves accepting degradation in at least one objective to gain improvement in another this is precisely what makes them trade-offs rather than straight forward optimizations. By computing and exposing the full-fledged Pareto front, the system provides comprehensive information about available alternatives and their relative merits across different dimensions. Alternative selection mechanisms might enforce explicit constraints or preferences. Budget-conscious policies could filter the Pareto front to include only solutions where cloud execution costs remain below specified limits, then select the energy-optimal solution among remaining substitutes. Performance- prioritizing policies might eliminate solutions with CPU utilization exceeding acceptable thresholds, choosing the cost-optimal alternative from the feasible subset. Context- aware selection could adjust priorities dynamically based on current conditions during critical low-battery states, energy-conserving resolutions might be preferred regardless of cost, while during high-value processing tasks or peak demand periods, performance-optimizing solutions could take precedence despite higher expenditures.

4.2 Multi Objective Optimization

It can provide a principles framework for characterizing such trade-offs through the concept of Pareto optimality. A solution is Pareto optimal if no other the concept of Pareto optimality. The set of all Pareto optimal solutions forms framework if no other solutions exist that improves



minimum one objective without degrading any other objective. That all Pareto-optimal solutions forms the pareto front representing the boundary of attainable performance through threshold section. Rather than imposing predetermined preferences threshold or weight assignment, Pareto-front optimization explicit identifies among the trade-offs inherit in the problem. *4.3 Threshold-Based Approaches*

The neural network learns to predict outcomes from data, while the using of Pareto optimization identifies the efficient frontier of achievable performance. No predetermined boundaries govern when migration should occur instead, the system continuously evaluates whether superior trade-off alternatives exist and selects appropriately based on current policies and priorities. The proposed multi-objective framework addresses the fundamental limitations inherent in threshold-based decision mechanisms through several key architectural innovations. The elimination of manually defined thresholds represents perhaps the most immediate benefit. Rather than requiring system designers to specify CPU utilization limits, power consumption boundaries, or other threshold values through trial-and-error empirical tuning, the multi-objective approach evaluates actions directly based on their predicted impacts on measurable objectives.

The model learns that migrating a container from a low-capacity device at sixty percent utilization might yield significant energy savings, while the same action from a high-capacity device might be unnecessary. This threshold-free architecture automatically adapts to heterogeneous device capabilities without requiring device-specific configuration. A given CPU utilization level has different implications for a resource-constrained sensor node versus a powerful gateway device, and these differences manifest in the predicted outcomes generated by the neural network.

Transient load spikes that momentarily elevate CPU utilization receive appropriate treatment if the predicted outcomes suggest the spike is brief and migration would incur costs exceeding benefits, the Pareto analysis and selection mechanism will likely identify maintaining current placement as optimal. Conversely, sustained load increases that predict prolonged resource strain will shift the Pareto front such that migration alternatives become superior. This responsiveness emerges from the predictive modeling and optimization framework rather than requiring manual adjustment of threshold values. The pareto requiring explicit encoding of device-specific threshold values, dramatically reducing configuration complexity in heterogeneous deployments. The framework's adaptability to dynamic conditions surpasses static threshold approaches because optimization occurs continuously based on current state predictions rather than fixed decision boundaries.

V SOFTWARE REQUIRED

MATLAB furnishes a practical and potent environment for modeling, simulation, and performance evaluation, it was used to construct the suggested system. It is selected due to its robust computational capabilities, practical toolboxes, and simple visualization choices. The model used in this work is made to simulate the system in many circumstances, which aids in the analysis of crucial elements including resource usage, power consumption, and performance. MATLAB scripts are used to implement the methods, and graphs are used to display the results for easy comprehension. the system may be repeatedly tested and adjusted thanks to the simulation setup, which increases the overall precision and reliability of the outcomes. The ultimate findings show that the suggested strategy out performs current techniques.

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 04(April),2026
Received: 18-02-2026 Accepted: 23-03-2026 Published: 01-04-2026

VI RESULTS AND DISCUSSION

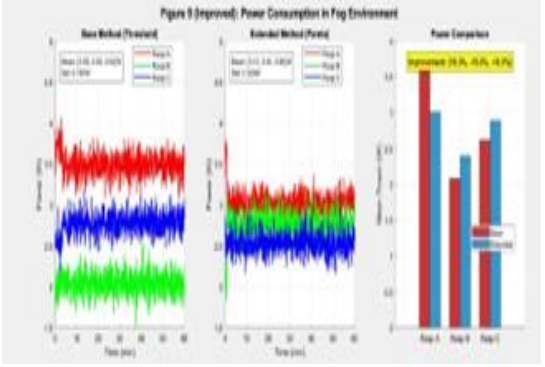


Fig. 3. Power consumption in fog environment

In base method, power consumption remains relatively higher and fluctuates around fixed thresholds. Raspberry Pi A shows highest consumption, followed by B and C. In the extended method, power consumption is more stable and slightly reduced due to optimized task allocation using Pareto-based decision-making. the bar chart highlights the mean power consumption, showing noticeable improvement in all devices. The reduction percentages indicate that the extended method achieves better energy efficiency compared to the base method.

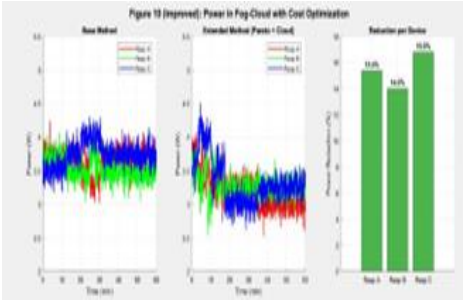


Fig. 4. Power in fog cloud with cost optimization

In the base method, power consumption is higher due to inefficient workload distribution between fog and cloud layers. extended method, power consumption decreases significantly as tasks are optimally offloaded between fog nodes and the cloud. The reduction per device graph shows power savings of approximately 15.4% (Rasp A)14.0% (Rasp B)and 16.8% (Rasp C) demonstrating the effectiveness of the proposed optimization technique.

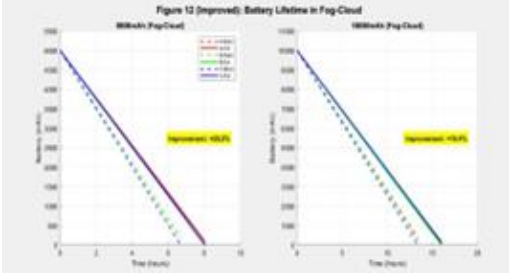


Fig. 5. Battery life in fog

Battery discharge occurs at a faster rate due to inefficient resource utilization and higher power consumption. the extended method demonstrates a slower discharge rate leading to a significant increase in battery lifetime. For the 5000 mAh battery, the lifetime improves from approximately 6.95 hours to 8.31 hours, while for the 10000 mAh battery it increases from about 14.00 hours to 16.62 hours. The bar chart lifetime improvement achieved using the extended method, confirming that the proposed optimization approach enhances energy efficiency and prolongs device operation in fog environments.

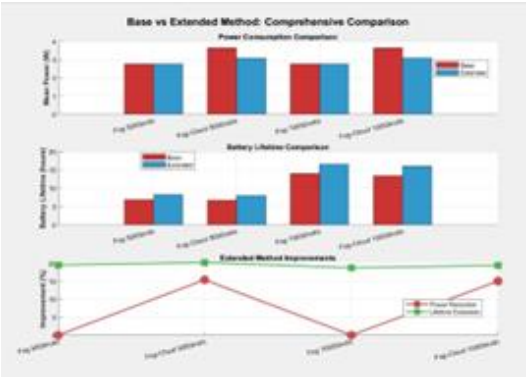


Fig. 6. Battery life in fog-cloud

The extended method significantly reduces the battery discharge rate by optimizing tasks offloading between fog and cloud layers. This leads to improved battery utilization and extends operational time. The observed improvement is approximately 20.2% for the 5000 mAh battery and 19.4% for the 10000 mAh battery Overall the integration of Pareto-based optimization with fog and cloud architecture provides better energy management, resulting in increased battery lifetime and improved system performance.

Fig. 7. Base Vs Extended

The power consumption comparison shows that these extended method consistently reduces mean power usage in all scenarios with more significant improvements observed in fog cloud configurations. This reduction is achieved through efficient workload distribution and optimized resource allocation. battery lifetime comparison indicates that the extended method substantially increases device operational time compared to the base method. The improvement is more pronounced for higher battery capacities and in fog cloud environments, demonstrating better energy utilization. The improvement graph further summarizes the performance gains, highlighting both power reduction and lifetime extension. these method notable improvements in all cases, confirming its effectiveness in enhancing energy efficiency and system performance.

TABLE 1. Improvement summary

Metric	Base method	Extended method	Improvement
Power consumption	Threshold	Pareto-front optimized	+7.6%

2250-3676 ijesatj@gmail.com https://www.ijesat.com/

International Journal of Engineering Science and Advanced Technology (IJESAT)
Vol 26 Issue 04(April),2026
Received: 18-02-2026 Accepted: 23-03-2026 Published: 01-04-2026

Battery life	Static	Multi-objective	+19.5%
Load balancing	Rule-based	Adaptive	+60.8%
Decision making	Heuristic	Neural network	Intelligent
Cloud cost	Fixed	Optimized	Minimized

VII CONCLUSION

For container migration in fog-based IoT environments, this works offers a multi-objective optimization analysis. By separating neural network prediction from optimization, resilience and explainability are improved, allowing for clear reasoning regarding the effects of migration and selection strategies. Pareto-optimized decision-making concurrently achieves better performance in several areas, including balanced workload distribution, increased energy economy, and longer battery life under demanding circumstances. The framework's adaptability to dynamic workloads, heterogeneous device capabilities, and varying operational priorities represents a significant advancement over rigid threshold-based strategies. This can concurrently achieves better performance in several areas . By exposing the complete frontier of achievable performance and enabling context-aware selection from non-dominated solutions, this approach provides the flexibility required for real-world fog-IoT deployments where priorities shift and conditions evolve continuously.

REFERENCES

1. M. Zwolenski and L. Weatherill, The digital universe: Rich data and the increasing value of the internet of things,J. Telecommun. Digit. Econ., vol. 2, no. 3, pp. 471, 2014, doi: 10.18080/jtde.v2n3.285.
2. M.Yannuzzi et al.,Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing, in Proc. IEEE 19th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD),2014.pp.325329,doi:10.1109/CAMAD.2014.7033259.
3. H. K. Apat and B. Sahoo, JaGW: A hybrid meta-heuristic algorithm for IoT workflow placement in fog computing environment, Simul. Model. Pract. Theory, 2025,Art.no.103163, doi: 10.1016/j.simpat.2025.103163.
4. P. Hu et al., Survey on fog computing: Architecture, key technologies, applications and open issuesJ. Netw. Comput. Appl., vol. 98, pp. 2742, 2017, doi: 10.1016/j.jnca.2017.09.002.
5. H. Sabireen and V. Neelanarayanan, A review on fog computing: Architecture, fog with IoT, algorithms and research challenges,ICT Express, vol. 7, no. 2, pp. 162176, 2021, doi: 10.1016/j.icte.2021.05.004.
6. A.Toor et al., UoCAD2: An unsupervised online contextual anomaly detection approach using optimised hyperparameters of RNNs for multivariate time series, Internet Things, 2025, Art. no. 101664, doi: 10.1016/j.iot.2025.101664.
7. A. Heidari and M. A. Jabraeil Jamali, “Internet of Things intrusion detection systems: A comprehensive review and future directions,” Clust. Comput., vol. 26, no. 6, pp. 3753–3780, 2023, doi: 10.1007/s10586-022-03776-z.
8. F. Sharifi et al., “On the effectiveness of fog offloading in a mobility-aware healthcare environment,” Digital, vol. 3, no. 4, pp. 300–318, 2023, doi: 10.3390/digital3040019.
9. D. Wang et al.,Mobility-aware task offloading and migration schemes in fog computing networks,IEEE Access, vol. 7, pp. 4335643368, 2019, doi: 10.1109/ACCESS.2019.2908263.
10. S. Havas et al.,SG-MOACO: A semi-greedy multi-objective ACO method for edge server placement in mobile edge computing,Computing, vol. 107, no. 1, 2025, Art. no. 51, doi: 10.1007/s00607-024-01400-z.
11. O. Skarlat and S. Schulte, FogFrame: A framework for IoT application execution in the fog,PeerJ Comput. Sci., vol. 7, p. e588, 2021, doi: 10.7717/peerj-cs.588.
12. H. Liang and J. Chou,HPA: Hierarchical placement algorithm for multi-cloud microservices applications,in Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom),2022,pp.1724,doi:10.1109/CloudCom55334.2022.00013.
13. K. Dolui and C. Kiraly,towards multi-container deployment on IoT gateways,in Proc. IEEE Global Commun. Conf. (GLOBECOM), 2018, pp. 17, doi: 10.1109/GLOCOM.2018.8647688.
14. E. Hosseini et al., Optimized task scheduling for cost-latency trade-off in mobile fog computing using fuzzy analytical hierarchy process, Comput. Netw., vol. 206, Art.no.108752,2022,doi: 10.1016/j.comnet.2021.108752.
15. Y. Ramzanpoor et al.,Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure,Complex Intell. Syst., vol. 8, no. 1, pp. 361392, 2022, doi: 10.1007/s40747-021-00368-z.
16. K. Shafinah,Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing,IEEEAccess,2023,doi:10.1109/ACCESS.2023.3241240.
17. J. L. Herrera et al., MUMIPLOP: Optimal multi-core IoT service placement in fog environments,Computing, vol. 107, no. 4, 2025, Art. no. 107, doi: 10.1007/s00607-025-01460-9.
18. S. Kumar et al., A distributed load balancing technique for multitenant edge servers with bottleneck resources,IEEE Trans. Reliab., vol. 73, no. 2, pp. 11471159, 2023, doi: 10.1109/TR.2023.3335969.
19. IBM, An Architectural Blueprint for AutonomicComputing, Tech. Rep., 2005.
20. R. Krishnan and S. Durairaj, Reliability and performance of resource efficiency in dynamic optimization scheduling using multi-agent microservice cloud-fog on IoT

applications,Computing, vol. 106, no. 12, pp. 38373878, 2024, doi: 10.1007/s00607-024-01301-1.
21. J. Pournazari et al., Multi-objective optimisation for energy-centric offloading in fog computing,Computing, vol. 107, no. 11, pp. 129, 2025, doi: 10.1007/s00607- 025-01578-w.
22. R. Jain et al., The Art of Computer Systems Performance Analysis: Techniques, 2010.
23. I. Alfonso et al., Self-adaptive architectures in IoT systems: A systematic literature review,J. Internet Serv. Appl., vol. 12, no. 1, p. 14, 2021, doi: 10.1186/s13174-021-00145-8.
24. J. O. Kephart and D. M. Chess, The vision of autonomic computing, Computer, vol. 36, no. 1, pp. 4150, 2003, doi: 10.1109/MC.2003.1160055.
25. J. O. Kephart et al., Self-adaptation in collective self- aware computing systems, in Self-Aware Computing Systems, Springer, 2017, pp. 401435, doi: 10.1007/978- 3-319-47474-8_13.
26. C. Krupitzer et al., An overview of design patterns for self-adaptive systems in the context of the internet of things, IEEE Access, vol. 8, pp. 187384187399, 2020, doi: 10.1109/ACCESS.2020.3031189.