International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

Beyond Reactive Scaling: A Review of AI-Driven Proactive and Context-Aware Auto-Scaling in Cloud-Edge Environments

Hemasree Koganti

Independent Researcher Fremont, California, USA

ABSTRACT: The rapid nature of cloud-edge computing requires advanced resource scaling techniques. Though the popular reactive scaling is plagued by inefficiency and delay, this review presents AI-based proactive and context-aware auto-scaling methodologies that forecast workload changes and scale resources in advance. By reviewing the combination of machine learning and deep learning algorithms with cloud-native orchestration tools, this paper points out their scope in reducing latency, saving costs, and improving performance. In addition, we provide an overview of current methodologies, present context-aware decision-making mechanisms, and list open research problems. The aim of this paper is to direct future research into managing sustainable and intelligent cloud-edge infrastructures.

KEYWORDS: AI-driven auto-scaling, proactive scaling, context-aware computing, cloud-edge environments, machine learning, Kubernetes, workload prediction, edge computing, resource optimization, deep learning.

I. INTRODUCTION

A. Background and Motivation

The accelerated growth of cloud-edge infrastructure driven by digital transformation across finance, healthcare, and IoT industries has required the creation of very elastic and effective resource allocation strategies. Traditional auto-scaling methods, particularly reactive ones, have limitations because they are based on threshold-based criteria such as CPU or memory usage [8]. Such methods lead to over-provisioning or under-provisioning of resources, leading to inefficient SLA compliance and increased operational costs.

In contrast, anticipatory and context-sensitive auto-scaling based on artificial intelligence is a novel approach that employs ML and DL algorithms to forecast demand and allocate resources efficiently in advance to avoid performance loss. The problem is especially severe in cloud-edge architecture, where latency-sensitive applications like AR/VR, autonomous driving platforms, and stock trading systems demand real-time response, and the limitations of reactive scaling are more evident [36].

Application of contextual data—network latency, user activity, and service-level objectives (SLOs)—in scaling decision-making greatly enhances responsiveness and resource utilization. Continuous development of edge computing and container orchestration technologies, such as Kubernetes, has made it feasible to employ intelligent auto-scaling agents in distributed systems, which can offer fine-grained control over performance and expenditure [16].

B. Objectives and Scope

This review investigates AI-based auto-scaling methods that transcend traditional reactive paradigms. The specific objectives include:

- Exploring predictive ML/DL techniques for intelligent resource forecasting.
- Investigating context-aware decision-making in distributed edge-cloud systems.
- Comparing the effectiveness, limitations, and trade-offs of proactive versus reactive and context-agnostic approaches.
- Identifying current gaps and suggesting future directions for AI-augmented auto-scaling.

The scope of this paper includes time-series forecasting using models like LSTM and Transformers, reinforcement learning-based policy optimization, federated learning for decentralized environments, and integration strategies using Kubernetes and other cloud-native tools.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

C. Article Organization

The paper is organized as follows:

- Section 2 presents the fundamentals of auto-scaling in cloud-edge architectures.
- Section 3 reviews AI-driven proactive auto-scaling methods using ML and DL models.
- Section 4 delves into context-aware strategies and their role in dynamic resource management.
- Section 5 compares and evaluates various approaches based on key metrics.
- Section 6 outlines open research challenges and prospective advancements.
- Section 7 concludes with a summary of insights and future directions for intelligent auto-scaling systems.

II. FUNDAMENTALS OF AUTO-SCALING IN CLOUD-EDGE ENVIRONMENTS

A. Overview of Auto-Scaling

Auto-scaling is the process of automatically scaling computational resources—like virtual machines (VMs), containers, or edge nodes—in anticipation of varied workload needs. Horizontal scaling, which adds or removes instances, and vertical scaling, which ups or downs the capacity (e.g., CPU, memory) of a given instance, are the two major types [1]. Horizontal scaling is typically favored in microservices and container environments because it is flexible with minimal downtime.

In cloud-edge infrastructures where workloads extend across centralized cloud servers to geographically dispersed edge nodes, real-time responsiveness and cost efficiency become rather daunting challenges. Auto-scaling processes need to consider both computational capacity and network topologies, since latency and bandwidth conditions significantly impact end-user experience [3].

B. Reactive vs. Proactive Scaling

Historically, reactive auto-scaling is based on pre-configured metrics and thresholds—e.g., triggering scaling actions when CPU usage passes the 80% or memory usage crosses a threshold. While easy and useful for static or deterministic workloads, this has limitations in response latency, usually triggering scaling after the performance issue is already discovered [8].

In contrast, proactive auto-scaling attempts to anticipate workload variations ahead of time through statistical analysis, machine learning algorithms, or blended methods. This foreknowledge enables avoidance of service degradation, SLA breaches, and wastage of resources and cost [39]. For instance, a future traffic spike predicted by a time-series model would pre-emptively scale resources, reducing queue latency and processing slowdowns.

TABLE 1
BELOW OUTLINES KEY DIFFERENCES BETWEEN REACTIVE AND PROACTIVE SCALING MECHANISMS

Feature	Reactive Scaling	Proactive Scaling
Trigger Condition	Threshold-based (e.g., CPU > 80%)	Forecast-based (e.g., predicted traffic spike)
Response Time	Post-event (after metric breach)	Pre-event (based on anticipated trends)
Implementation Complexity	Low	Medium to High
Resource Utilization	Risk of over/under-provisioning	Optimized for expected demand
Suitability	Static/Slow-changing workloads	Dynamic/Real-time workloads (e.g., financial apps)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

C. Challenges in Cloud-Edge Auto-Scaling

Auto-scaling in hybrid cloud-edge environments introduces a unique set of challenges that differ from traditional cloud-only setups.

- Heterogeneity of Resources: Edge environments comprise diverse hardware configurations, including low-power devices with limited processing capacity. This heterogeneity complicates uniform scaling policies [4].
- Latency Sensitivity: Applications such as augmented reality (AR), real-time analytics, and financial trading systems are extremely sensitive to network latency. Even minor delays in scaling can degrade quality of service (QoS) [16].
- Workload Unpredictability: Cloud-edge systems must handle bursty and volatile workloads driven by user behavior, event-driven traffic, or sensor data. Designing scaling mechanisms that accommodate these patterns requires advanced prediction models and context-aware policies [15].
- Limited Observability at the Edge: Unlike cloud platforms with robust monitoring tools, edge nodes often lack comprehensive observability, making it difficult to gather accurate metrics for informed scaling decisions [2].
- Security and Data Privacy: Data offloading to the cloud for scaling decisions may raise privacy concerns, particularly in applications like healthcare and finance. Federated learning and on-device intelligence offer potential solutions [4].

Therefore, intelligent, distributed, and low-latency solutions that can adjust to infrastructure limitations and application-level performance objectives are necessary for efficient auto-scaling in such environments.

III. AI-DRIVEN PROACTIVE AUTO-SCALING TECHNIQUES

In cloud-edge environments, proactive auto-scaling has been made possible in large part by artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL). By using past data, present metrics, and contextual information to predict future demand, these intelligent systems enable resource provisioning before workload spikes actually occur. AI turns auto-scaling into a dynamic, data-driven process that offers more accuracy and responsiveness by substituting adaptive models for static rules (Garcia Lopez et al., 2022; Wang et al., 2020).

A. Machine Learning-Based Predictive Scaling

ML-based predictive auto-scaling typically utilizes time-series forecasting techniques to estimate workload patterns over time. Popular algorithms include:

- Autoregressive Integrated Moving Average (ARIMA): A traditional statistical model that performs well on linear, stationary data but struggles with non-linear or seasonal workload patterns [8][34].
- Support Vector Regression (SVR): A supervised learning approach effective for small datasets with limited noise, though it lacks scalability and adaptability to high-dimensional data.
- Long Short-Term Memory (LSTM) networks: LSTMs are a type of recurrent neural network (RNN) particularly suited for capturing long-term temporal dependencies in workload sequences. LSTM-based models have demonstrated superior performance in predicting resource usage in cloud applications [34].
- Reinforcement Learning (RL): RL methods enable an agent to learn optimal scaling policies through continuous interaction with the environment. Over time, the system adapts to traffic changes and makes intelligent scale-up or scale-down decisions to balance cost and performance [16].

A prominent implementation of ML-based auto-scaling is Google Cloud's Predictive Autoscaler, which utilizes historical metrics and seasonal trends to adjust compute instances before demand spikes occur.

B. Deep Learning for Workload Prediction

DL models, with their hierarchical architectures and non-linear capabilities, outperform traditional ML techniques in handling complex, high-dimensional data. In workload forecasting, the following DL models are widely explored:

- CNN-LSTM Hybrids: These models combine Convolutional Neural Networks (CNNs) for extracting spatial features and LSTMs for capturing temporal dynamics, effectively modeling spatiotemporal patterns in distributed workloads [34].
- Transformer Networks: Originally designed for Natural Language Processing (NLP), transformers leverage self-attention mechanisms to model long-range dependencies. They have shown remarkable performance in workload forecasting by capturing intricate relationships between input features across time [20].
- Autoencoders for Anomaly Detection: DL-based autoencoders are also used to detect anomalies in resource utilization patterns, prompting pre-emptive scaling when unusual behavior is detected (Chen et al., 2021).
- Graph Neural Networks (GNNs): In edge computing environments with graph-like topologies, GNNs have emerged as powerful tools to predict workload flows across nodes, aiding decentralized scaling [15].

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

Deep learning models can be trained offline using historical traces and then deployed within Kubernetes Horizontal Pod Autoscaler (HPA) or Vertical Pod Autoscaler (VPA) extensions to enable predictive elasticity in production environments.

C. Proactive Scaling in Edge Environments

Proactive scaling at the edge introduces challenges due to resource constraints, privacy concerns, and network variability. To address these, researchers are exploring several innovative solutions:

- Federated Learning (FL): FL enables collaborative model training across multiple edge nodes without sharing raw data, preserving privacy while still allowing effective workload prediction. This decentralized intelligence supports local scaling decisions while maintaining global coordination [4].
- Edge-Aware Reinforcement Learning: Customized RL algorithms are adapted for edge conditions, optimizing for latency, bandwidth, and energy consumption. These models dynamically prioritize local vs. cloud scaling based on workload criticality and node availability [15].
- Collaborative Edge Clusters: Nodes in proximity form clusters and share workload predictions to distribute the scaling burden intelligently. Clustering improves model robustness and ensures fault tolerance by avoiding single-point bottlenecks [15].
- Lightweight AI Models: Edge devices often lack GPU acceleration. Hence, researchers develop lightweight DL architectures such as MobileNet or TinyLSTM for on-device inference, facilitating low-latency scaling decisions.

In conclusion, proactive auto-scaling using AI models is revolutionizing how latency-sensitive edge systems adjust to changing demands in addition to revolutionizing cloud-native applications.

IV. CONTEXT-AWARE AUTO-SCALING STRATEGIES

Proactive auto-scaling relies heavily on predictive models, but context-aware approaches go one step further by adding more real-time data about the system and its surroundings. Context describes dynamic parameters that can have a big impact on scaling decisions, like user behavior, network status, application-level QoS requirements, and location-specific factors. Systems can make more intelligent, situation-specific adjustments by integrating such context, which is particularly important in distributed and heterogeneous cloud-edge environments [15][4].

A. Defining Context in Auto-Scaling

Context in auto-scaling can be broadly categorized into the following dimensions:

- User-Centric Context: The user's location, device type, session length, and frequency of access are all included in the user-centric context. For instance, auto-scaling during periods of high usage can assist in keeping latency low for apps that stream media or conduct banking [8].
- Network Context: When deciding whether to process a task at the edge or offload it to the cloud, factors like packet loss, network latency, and bandwidth fluctuations are important considerations [2].
- Application-Level QoS Requirements: These comprise application-specific metrics like throughput, jitter, and response time. Compared to workloads involving batch processing, real-time services such as video conferencing require distinct scaling strategies [15].
- Infrastructure Context: Making wise scaling decisions also requires consideration of energy limitations, device heterogeneity, and the availability of computational resources at both local and distant nodes.

The system can attain granular control by utilizing this complex context, which enables scaling policies to be in line with both anticipated demand and current operational constraints.

B. AI Techniques for Context Awareness

To extract and utilize context effectively, a variety of AI techniques have been explored:

- Clustering Algorithms (e.g., K-Means, DBSCAN): such as K-Means and DBSCAN, are unsupervised learning techniques that group related user behaviors or workload patterns and enable customized scaling rules for each cluster. IoT workloads from comparable sensor types, for instance, could be grouped together and scaled consistently [39].
- Bayesian Networks: Based on ambiguous or insufficient data, these probabilistic graphical models can forecast future system states and capture dependencies among different contextual variables. They are especially helpful when several contextual factors interact to affect performance, such as network speed and user load [4].
- Decision Trees and Random Forests: These models use context-feature inputs to produce interpretable policies that offer transparency and explainability in scaling decisions, which is crucial in regulated sectors like healthcare and finance [20].
- Online Learning Models: Online learning ensures that scaling decisions stay current even in volatile situations by gradually adapting models in highly dynamic environments as new context data becomes available.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

These AI methods can be implemented as external controllers integrated with Prometheus and KEDA (Kubernetes Event-driven Autoscaler) or integrated into orchestration platforms such as Kubernetes through Custom Resource Definitions (CRDs).

C. Dynamic Policy Adaptation

A key advantage of context-aware auto-scaling is the ability to define dynamic and self-adaptive policies rather than relying on hard-coded thresholds.

- Rule-Based Systems: Scale out if latency exceeds 100ms during business hours" is one of the manually defined rules used by these systems. Despite being simple, they necessitate domain knowledge and frequently don't generalize to different situations [3].
- Self-Learning Policies: AI-driven systems are able to adjust policies over time by learning from past trends. For instance, it is possible to effectively balance cost and performance by training reinforcement learning models to dynamically modify thresholds in response to shifting traffic profiles [16].
- Hybrid Approaches: Some systems offer a semi-automated solution where basic logic is improved through learning by combining AI feedback loops with rule-based templates. This method works well in production environments where auditable control is necessary but adaptive optimization is desired [15].

TABLE 2
SUMMARIZES KEY CONTEXT-AWARE AI TECHNIQUES AND THEIR APPLICATIONS

Technique	Purpose	Example Application
K-Means Clustering	Group similar workloads	IoT sensor clusters in smart cities
Bayesian Networks	Model probabilistic dependencies	QoS-aware scaling for healthcare applications
Reinforcement Learning	Learn dynamic, long-term policies	Financial service workload orchestration
Decision Trees	Generate interpretable policy rules	Edge resource allocation in latency-sensitive apps

V. COMPARATIVE ANALYSIS OF AI-DRIVEN AUTO-SCALING APPROACHES

The performance, effectiveness, and operational trade-offs of various AI-driven auto-scaling techniques are assessed and contrasted in this section. We can better understand their applicability in various cloud-edge deployment scenarios by looking at metrics like scalability, cost-effectiveness, energy consumption, and adaptability. Comparative frameworks are essential for comprehending the theoretical differences between these approaches as well as their practical performance under various workloads.

A. Performance Metrics

To assess the effectiveness of auto-scaling strategies, the following core performance metrics are commonly used [2][20]:

- Scalability: Refers to the system's ability to seamlessly adapt to increasing workloads without performance degradation.
- Cost Efficiency: Measures how well the auto-scaling strategy optimizes resource usage to avoid both over-provisioning (leading to excess costs) and under-provisioning (causing SLA violations).
- Latency/QoS Adherence: Indicates how effectively the method maintains application responsiveness, especially under bursty traffic.
- Energy Consumption: Relevant in edge settings where devices are resource-constrained and energy efficiency is crucial.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

- Decision Overhead: Reflects computational complexity and time taken by the scaling controller to decide on actions.

Each scaling strategy offers trade-offs across these metrics. For example, while deep learning offers accuracy, it may introduce high decision latency compared to rule-based methods.

B. Strengths and Limitations

The following table summarizes key strengths and limitations of various auto-scaling paradigms:

Approach	Strengths	Limitations
Reactive Scaling	Simple to implement; low computational cost	Latency-prone; responds post-demand spike; limited in dynamic workloads
ML-Based Proactive	Learns from historical data; cost-effective; improves SLA compliance	Requires extensive data; retraining needed with workload shifts
DL-Based Forecasting	Captures complex, non-linear patterns; suitable for long-term predictions	High computational cost; interpretability challenges
Context-Aware Scaling	Real-time adaptive; integrates environmental factors	Policy design complexity; requires multi-modal data streams
RL-Based Decisioning	Self-adaptive; long-term performance optimization	Convergence time; safety constraints in mission-critical systems
Federated Learning	Data privacy; decentralized control	Communication overhead; model heterogeneity

This comparative understanding is crucial for platform architects choosing appropriate scaling strategies in environments like distributed finance, smart cities, or 5G-based applications.

C. Case Studies and Real-World Applications

Several notable implementations of AI-driven auto-scaling illustrate the strengths of these approaches:

- AWS Predictive Auto Scaling: AWS uses machine learning models to forecast EC2 instance needs based on past usage and seasonality. It proactively scales resources to minimize cost and maintain SLA targets. The approach, however, is primarily cloud-centric and less suited for edge-heavy deployments (AWS, 2023).
- Google Cloud Prophet Models: Google Cloud leverages Facebook's Prophet forecasting tool for resource prediction. Prophet works well for time-series data with strong seasonal effects, though its accuracy diminishes with high variability [3].
- Kubernetes with RL Agents: Research prototypes integrate reinforcement learning agents into Kubernetes to dynamically adjust pod replicas based on response times and request rates [16]. These implementations show promising cost savings and performance boosts but require safety policies to avoid erratic behaviors during training.
- Edge-Aware FL Platforms: Emerging solutions combine federated learning with edge orchestrators to perform privacy-preserving, real-time scaling. These are particularly impactful in smart healthcare and connected vehicle systems, where local conditions must be factored into scaling decisions [4].

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

- Open-source Solutions (e.g., KEDA): Kubernetes Event-driven Autoscaler (KEDA) supports scaling based on event sources like message queues and HTTP requests. While rule-based, KEDA is often used with AI plugins for dynamic scaling in production [20].

These implementations collectively demonstrate the practical viability of AI-enhanced scaling strategies while highlighting the need for context sensitivity, explainability, and cross-layer optimization.

VI. OPEN CHALLENGES AND FUTURE DIRECTIONS

Even though AI-driven auto-scaling has advanced significantly, there are still many implementation challenges, trust concerns, and standardization gaps in the field, especially when it comes to the cloud-edge continuum. Addressing the following unresolved issues becomes essential as applications develop to require ultra-reliability, low latency, and scalability across distributed infrastructures.

A. Explainability and Trust in AI Models

The inability of complex models, particularly deep learning and reinforcement learning, to be explained is one of the main obstacles to the broad use of AI-based auto-scaling [20]. In mission-critical industries like finance or healthcare, where user trust, auditability, and regulatory compliance are crucial, these "black-box" systems frequently fall short in providing justification for their scaling choices.

- Future Work: Integrating explainable AI (XAI) techniques such as SHAP, LIME, or attention visualization into auto-scaling frameworks can enhance transparency and decision accountability. Hybrid systems combining rule-based logic and AI are also a promising direction [4].

B. Cross-Layer Optimization

Existing AI-based auto-scaling systems frequently work in isolation, concentrating only on the infrastructure or application layer. However, cross-layer coordination is necessary to achieve true efficiency in distributed environments, from application-level QoS policies to network and resource provisioning [3].

- Future Work: Research should explore holistic models that unify control across layers. For example, scaling decisions at the application level could be informed by real-time telemetry from the network layer, or power profiles from the hardware layer in edge devices.

C. Integration with 5G and IoT Ecosystems

As 5G, IoT, and network slicing become the backbone of edge computing, existing auto-scaling models must adapt to new constraints such as ultra-low latency, slice-specific QoS, and device mobility [15].

- Future Work: AI-driven scaling strategies must be integrated with 5G orchestration tools, enabling service-level isolation and real-time responsiveness. Collaborative models combining edge intelligence and network-aware adaptation are critical to supporting latency-critical applications like autonomous driving or AR/VR.

D. Lack of Benchmarking Standards

Unlike traditional cloud benchmarks (e.g., SPEC Cloud, TPC), AI-driven scaling lacks standardized testbeds and metrics to evaluate performance uniformly [2]. This inconsistency hinders reproducibility and comparative studies across academic and industrial research.

- Future Work: The community must develop benchmarking suites for auto-scaling strategies—incorporating diverse workloads (e.g., web apps, stream processing, IoT), network conditions, and evaluation metrics (e.g., SLA violations, carbon footprint, decision latency). Open-source platforms like EdgeBench or KubeEdge may serve as foundational testbeds for such efforts.

E. Model Robustness and Adaptability

Workload behavior can change drastically due to external events like seasonal surges, user trends, or cyber incidents. Rigid models trained on historical data often fail to generalize to such variations [39].

- Future Work: Research must shift toward continual learning, meta-learning, and zero-shot adaptation techniques that enable models to evolve in real-time without complete retraining. Federated learning combined with transfer learning can facilitate cross-domain adaptability across similar deployment contexts.

F. Ethical and Environmental Considerations

As AI models become more complex, their energy consumption grows, which may contradict the objective of resource efficiency in cloud-edge infrastructures [8]. Furthermore, data privacy remains a concern when training models across sensitive domains.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

- Future Work: Future scaling systems should embed green AI practices, such as model pruning or quantization, and uphold ethical principles by preserving data sovereignty using federated or differential privacy-preserving learning techniques.

When taken as a whole, these difficulties offer prospects for future studies in reliable, flexible, and consistent AI-driven scaling. The future of autonomous digital infrastructure management will be determined by the capacity to anticipate, rationalize, and adjust to workload dynamics as cloud-edge ecosystems emerge as the operational foundation of new technologies.

VII. CONCLUSION

The greater complexity and dynamism in cloud-edge environments require intelligent, adaptive, and cost-conscious resource management solutions. Simple reactive auto-scaling methods in conventional methods are inadequate in addressing the high demands of today's applications like real-time analytics, financial transactions, and immersive media. This review has looked at how proactive and context-aware auto-scaling based on AI can overcome these shortcomings by forecasting workload patterns, factoring in environmental context, and responding dynamically.

By a thorough examination of machine learning, deep learning, reinforcement learning, and federated learning methods, we emphasized their capacity to enhance performance, scalability, and resource utilization. In addition, we examined the significance of context-awareness in facilitating scaling based not merely on usage metrics, but on user activity, QoS demands, and network conditions. Comparative observations across several approaches demonstrated both their applicability and natural trade-offs.

Even with such advancements, there are challenges. The absence of explainability, cross-layer coordination, benchmarking standards, and adaptability under changing workloads needs to be explored further. Support for new technologies such as 5G, IoT, and edge intelligence, and a move towards green and ethical AI will be the next frontier for auto-scaling systems. In conclusion, this review highlights the potential for AI-accelerated auto-scaling as a core capability in the future generation of autonomous cloud-edge infrastructure. Ongoing interdisciplinary study and industry collaboration will be important to achieving its full potential.

REFERENCES

[1] Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., & Merle, P. (2018). Elasticity in cloud computing: State of the art and research challenges. IEEE Transactions on Services Computing, 11(2), 430–447.
[2] Bermbach, D., Pröstl-Andrade, L., & Tai, S. (2020). Benchmarking auto-scaling strategies for cloud applications. Journal of Systems and Software, 168, 110638.
[3] Buyya, R., Srirama, S. N., Casale, G., & Varghese, B. (2021). Next-generation cloud computing: New trends and research directions. Future Generation Computer Systems, 115, 619–632.
[4] Chen, S., Wang, Z., Liu, W., & Zhang, Y. (2021). Federated learning for distributed auto-scaling in edge networks. IEEE Transactions on Services Computing, 14(6), 2105–2118.
[5] Chen, Y., Hu, Q., Liu, X., & Zhang, Y. (2020). Context-aware resource allocation in edge-cloud systems using Bayesian models. Computer Networks, 178, 107341.
[6] Dang, T. T., Truong, H. L., & Dustdar, S. (2019). AI-based resource auto-scaling for containerized applications. Proceedings of the IEEE International Conference on Cloud Computing, 192–199.
[7] Fang, Z., Wu, H., Ren, H., & Wang, G. (2023). Resource-aware deep reinforcement learning for edge-cloud orchestration. Journal of Parallel and Distributed Computing, 172, 1–12.
[8] Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., ... & Felber, P. (2022). AI-based predictive auto-scaling in cloud computing: A survey. Future Generation Computer Systems, 128, 205–220.
[9] Ghobaei-Arani, M., Souri, A., & Rahmanian, A. (2021). A hybrid metaheuristic algorithm for QoS-aware auto-scaling of cloud resources using container-based architecture. Concurrency and Computation: Practice and Experience, 33(6), e6067.
[10] Guo, Y., Zhang, Y., Lin, W., & Jiang, C. (2022). Transformer-based forecasting for auto-scaling microservices in multi-cloud environments. IEEE Access, 10, 88370–88384.
[11] Hasan, M. K., Islam, M. R., & Almogren, A. (2020). Anomaly-aware deep learning-based resource prediction for elastic cloud computing. Sensors, 20(14), 4025.
[12] He, Q., & Jin, H. (2022). Graph neural networks for resource allocation in edge computing. Computer Communications, 180, 140–152.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 6, Issue 3, May- June 2024 |

DOI:10.15662/IJEETR.2024.0603009

[13] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. ETSI White Paper, 11(11), 1–16.
[14] Islam, M. R., Hasan, M. K., & Gumaei, A. (2021). Deep LSTM with embedded feature selection for cloud workload prediction. Information Sciences, 562, 136–151.
[15] Kumar, J., Gupta, H., & Buyya, R. (2023). Context-aware auto-scaling for IoT-edge applications. IEEE Internet of Things Journal, 10(5), 4321–4335.
[16] Li, H., Wu, J., Huang, Y., & Hu, X. (2021). Reinforcement learning for dynamic resource provisioning in edge computing. IEEE Transactions on Cloud Computing, 9(3), 1125–1139.
[17] Li, Z., Zhang, Y., & Xu, Y. (2020). Energy-aware container placement in microservice-based edge computing using deep reinforcement learning. IEEE Transactions on Green Communications and Networking, 4(3), 828–842.
[18] Lin, H., & Wang, X. (2021). Multi-objective auto-scaling of containers in edge-cloud systems. Information Systems Frontiers, 23, 1085–1102.
[19] Liu, Z., & Zhang, Q. (2019). A hybrid deep learning framework for workload prediction in cloud systems. Neurocomputing, 361, 177–186.
[20] Mishra, A., Kumar, P., & Verma, S. (2022). Explainable AI for autonomous cloud resource management. ACM Computing Surveys, 55(8), 1–32.
[21] Mohan, N., Ehsan, M., & Wang, Q. (2021). A context-aware policy engine for auto-scaling edge applications. Future Internet, 13(8), 204.
[22] Nguyen, T. T., Nguyen, D. C., & Bhargava, B. (2021). Federated learning for edge-cloud orchestration in IIoT systems. Computer Networks, 194, 108117.
[23] Papageorgiou, E., Loukas, A., & Mylonas, G. (2020). Adaptive auto-scaling using online machine learning. Future Internet, 12(8), 134.
[24] Peng, W., & Wang, Y. (2019). Cost-aware resource management using reinforcement learning in cloud platforms. Future Generation Computer Systems, 95, 70–79.
[25] Qiu, T., Zhao, J., & Yang, S. (2020). Edge computing-enabled intelligent IoT: Performance optimization and adaptive learning. IEEE Network, 34(6), 136–143.
[26] Rahman, M., & Gao, J. (2022). Lightweight deep learning for real-time edge auto-scaling. Computer Communications, 183, 83–92.
[27] Raza, S., & Lee, S. (2021). A comparative study of auto-scaling mechanisms for microservice applications. Sustainable Computing: Informatics and Systems, 31, 100567.
[28] Sharma, A., & Sood, S. K. (2020). Predictive resource provisioning for cloud applications using LSTM networks. Computing, 102(10), 2391–2409.
[29] Singh, R., & Rajan, A. (2023). Online learning-based dynamic scaling for video streaming on edge servers. Multimedia Tools and Applications, 82, 13495–13514.
[30] Sun, Y., & Dong, X. (2020). A survey on adaptive and intelligent auto-scaling strategies. IEEE Transactions on Network and Service Management, 17(4), 2346–2364.
[31] Tang, F., McLaughlin, K., & Sezer, S. (2021). Cloud-native autoscaling with Prometheus and Kubernetes. Software: Practice and Experience, 51(6), 1367–1383.
[32] Thung, G., & Hoon, H. T. (2018). Towards decentralized auto-scaling in fog computing using reinforcement learning. Proceedings of the IEEE Fog World Congress, 1–6.
[33] Torres, J., & Breitbart, Y. (2022). Workload-aware adaptive scaling using online learning. Journal of Cloud Computing, 11, 45.
[34] Wang, L., Li, J., & Xu, H. (2020). Deep learning for workload forecasting in cloud data centers. Journal of Parallel and Distributed Computing, 144, 25–38.
[35] Wang, W., Chen, F., & Zhang, Z. (2021). A survey on federated learning for edge intelligence. ACM Transactions on Internet Technology, 21(2), 1–34.
[36] Wu, J., & Deng, R. H. (2020). AI-powered smart resource provisioning in fog-cloud environments. IEEE Internet Computing, 24(3), 34–42.
[37] Xie, X., & Ding, S. (2021). Contextual deep learning for adaptive scaling in serverless platforms. IEEE Transactions on Cloud Computing, 9(4), 1470–1483.
[38] Yang, J., Li, Z., & Huang, Y. (2019). Real-time scaling in microservices using reinforcement learning. Proceedings of the IEEE International Conference on Cloud Engineering, 140–147.
[39] Zhang, Y., Liu, M., & Tan, Y. (2019). Proactive elasticity in cloud computing: A time-series forecasting approach. Cluster Computing, 22(4), 1459–1474.
[40] Zhang, Z., Xu, X., & Zhou, Y. (2020). AI-driven resource orchestration for service-centric 5G networks. IEEE Wireless Communications, 27(4), 90–97.