https://doi.org/10.7840/kics.2024.49.8.1170

A Design and Implementation of Energy-Aware Resilience Architecture for Mobile Edge Cloud

JangWon Lee[◆], YoungHan Kim[°], Dooho Keum[*], Gyu-min Lee[*], Suil Kim[**], Myoung-hun Han[**]

ABSTRACT

In In edge-cloud environments, mobile nodes face significant challenges due to their mobility and the distributed nature of the environment. The unstable communication links between mobile nodes and the cloud often lead to frequent disruptions in connectivity, posing obstacles to seamless operation and service delivery. Effective energy management strategies are crucial to address these challenges and ensure the long-term viability of mobile nodes. In this paper, we propose an architecture for energy-aware resilience in edge-cloud environments for a standalone mobile node in an edge-cloud environment that can operate seamlessly in the connection disruption from the cloud. Our architecture leverages machine learning-based energy consumption prediction techniques to forecast energy consumption patterns while considering dynamic network conditions. In addition, we propose a threshold-based control policy for autonomous node resilience, enabling mobile nodes to adaptively adjust their operations in response to fluctuating energy levels and network conditions of edge environments. Through proactive energy management strategies, such as workload autoscaling with energy awareness, we aim to minimize energy consumption and maximize node survival time, particularly under constrained conditions. Experimental evaluations demonstrate the efficiency of our proposed approach in extending node longevity and ensuring reliable operation in dynamic and resource-constrained edge-cloud environments.

Key Words : Mobile Edge Computing, Resiliency, Cloud Computing, Scaling

Ⅰ. Introduction

Mobile computing has become ubiquitous in today's digital landscape, with an increasing number of devices and applications requiring seamless connectivity and on-the-go access to resources[1,2]. This proliferation of mobile devices, coupled with the advent of edge computing paradigms, has paved the way for a new era of computing where data processing and storage capabilities are distributed closer to the point of generation, enabling low-latency and context-aware services[3]. However, the effectiveness of edge-cloud systems heavily depends on the resilience and efficiency of mobile nodes operating within these environments[4,5].

In traditional edge computing scenarios, edge nodes are typically stationary entities deployed at fixed locations within a network. However, the emergence of mobile edge computing (MEC) introduces a paradigm shift by allowing edge nodes to exhibit mobility[6,7]. This enables dynamic computing resource reconfiguring based on changing environmental conditions, user demands, and network topology. This mobility concept adds a layer of complexity and opportunity, pre-

※ This work was supported by the Agency for Defense Development by the Korean Government(UC210025VD).
◆ First Author : Soongsil University Department of Electronic Engineering, jangwon.lee@dcn.ssu.ac.kr, 학생회원
° Corresponding Author : Soongsil University Department of Electronic Engineering, younghak@ssu.ac.kr, 종신회원
* LIG Nex1, dooho.keum@lignex1.com, 정회원; gyumin.lee@lignex1.com
** Agency for Defense Development(ADD), sikim777@add.re.kr; mengddor@add.re.kr, 정회원
논문번호 : 202405-096-C-RN, Received May 13, 2024; Revised June 14, 2024; Accepted June 17, 2024

senting new challenges and possibilities for designing resilient and energy-efficient systems.

Energy-aware resilience in mobile edge computing becomes particularly pertinent in dynamic environments where edge nodes may relocate frequently by moving[8-10]. Energy consumption is a critical concern for mobile devices, as limited battery life constrains their operational capabilities and longevity. Moreover, unstable communication links between mobile nodes and the cloud hinder seamless connectivity, leading to frequent disruption[11].

Implementing mobile edge nodes in dynamic environments necessitates a new architectural framework capable of accommodating node mobility while meeting the requirements of energy-aware resilience.

Despite the growing interest in mobile edge computing and energy-aware resilience, several limitations persist in current research efforts. In mobility management, numerous studies have explored techniques for enhancing the mobility of nodes within edge-cloud environments[12,13]. These approaches have focused on mobility prediction algorithms to anticipate the movement patterns of mobile nodes and optimize resource and service allocation accordingly. Similarly, in energy optimization, numerous strategies have been proposed to mitigate energy consumption[14,15] and prolong the battery life of mobile nodes[16]. However, these studies lack connection status consideration, which is the main characteristic of a mobile node in an edge-cloud environment. It can interrupt the connection process between mobile nodes, reducing the node's life and changing the energy consumption process.

The motivation behind this research stems from the pressing need to develop comprehensive solutions that address the challenges posed by mobility, energy consumption, and resilience in the mobile edge-cloud environment. By advancing the state-of-the-art energy-aware resilience strategy and architectural design for mobile edge nodes, we aim to contribute to designing an architecture for supporting emerging applications in areas such as the military, where mobile nodes operate under stringent energy constraints, necessitating effective energy management strategies. The connection between these mobile nodes and cloud computing servers is usually disrupted due to unstable network conditions at edge environments.

In this paper, to address the energy consumption problem from network conditions of edge environment, we propose leveraging machine learning techniques to enhance the adaptability and efficiency of energy consumption prediction in mobile edge computing environments. In addition, the mobile edge node is provided with the self-management concept for resilience that can autonomously perform recovery actions independently without a direct connection from the cloud server. We also propose a threshold-based control policy mechanism for node resilience that can control the node's operation under different remaining energy levels to minimize the node's energy consumption and survival time. Self-management mobile edge nodes empower to adaptively manage energy resources and optimize performance in dynamic and resource-constrained edge environments.

Through experimentation and validation, we seek to demonstrate the effectiveness of our proposed approach in prolonging node longevity, minimizing energy consumption, and maximizing system resilience.

In summary, our contributions to this paper are as follows:

- Propose an energy-aware resilience architecture for mobile edge nodes, leveraging machine learning to predict energy consumption and considering network connection status in unstable edge environments.
- Propose a threshold-based energy control policy at the self-management mobile edge node, which can be configurable to node energy characteristics and automatically adjust the node's operation by executing resilience actions such as workload autoscaling to minimize the node's energy consumption and increase its survival time.

The remainder of this article is organized as follows: Section II presents our proposed architecture and main components, followed by the contribution and significance of this work. Section III provides our setup for stimulation and experiment setup to evaluate the proposal, along with our experiment results, then followed by a conclusion and future work in Section IV.

Ⅱ. Design of Energy-Aware Resilience for Mobile Node in Edge-Cloud Environment

This section presents components of the proposed architecture with energy-aware resilience based on machine learning for mobile edge nodes. We also illustrate design choices, requirements for the model machine learning and energy consumption/remaining prediction mechanism, and an auto recovery strategy, which aims to enable the adaptive resilience of mobile nodes.

2.1 Overview of System Architecture

Figure 1 depicts the proposed architecture. The architecture for energy-aware resilience in mobile nodes, integrating cloud and edge components, is structured to optimize energy usage and ensure system resilience in dynamic mobile computing environments.

Several modules are deployed at the edge to monitor and manage mobile node operations efficiently. *Monitoring Agents:* the edge infrastructure hosts monitoring agents, including the Node Energy Collector, Node Resource Collector, and Node Connection Status Collector. These agents continuously gather real-time data on energy consumption, resource utilization, and network connectivity status, providing crucial insights into the health and performance of mobile nodes.

Energy-Aware Policy Agent: Configurable from the cloud, the Energy-Aware Policy Agent is responsible for determining decision actions based on mobile nodes' health status and energy levels. By analyzing collected data from monitoring agents and applying predefined policies or machine learning models, this agent dynamically adjusts node configurations and behaviors to optimize energy efficiency and maintain resilience. In addition, these policies and trained models are updated regularly after the training process on the cloud with the newest collected data to ensure accuracy and minimize errors affecting prediction results.

Workload Autoscaler: The Workload Autoscaler module dynamically scales the computational workload of mobile nodes based on demand and resource availability. Leveraging insights from monitoring agents and energy-aware policies, this module adjusts the allocation of computational resources to match changing workload requirements, ensuring efficient resource utilization and minimizing energy wastage.

The edge components interact with the cloud part, which serves as the central management and deci-

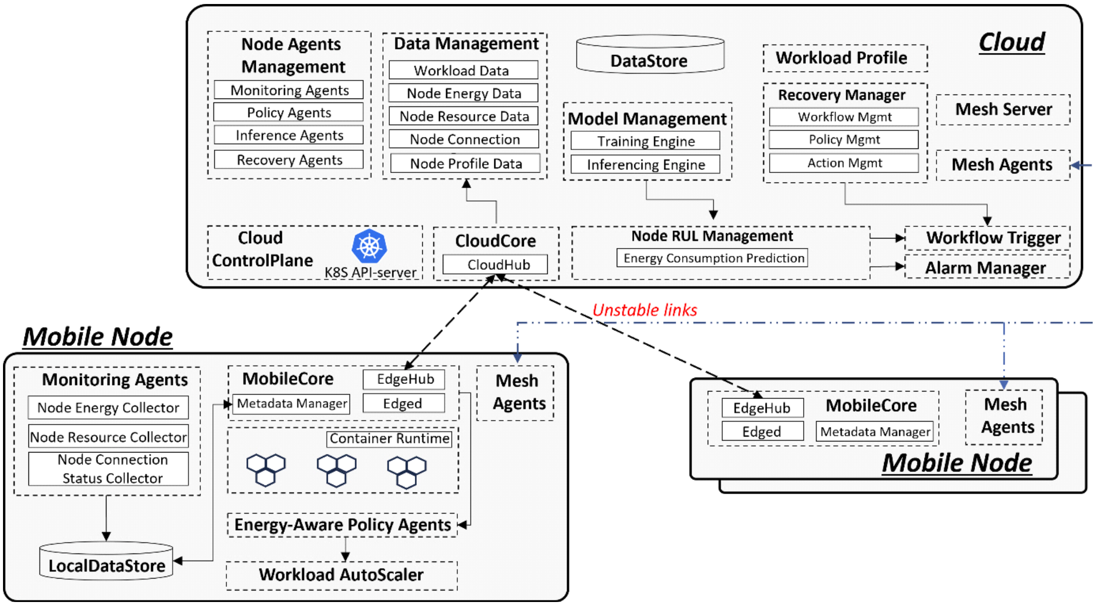

Fig. 1. Energy-aware Resilience for Mobile Node in Edge-Cloud Environments.

sion-making hub.

Mobile Code and CloudCore Connection Mobile nodes connect with the cloud through the Mobile Code and CloudCore interface. This connection facilitates communication, data exchange, and coordination between mobile nodes and the cloud-based infrastructure, enabling seamless integration and collaboration across distributed environments.

Energy-Aware Operation Support Modules: These modules support energy-aware operations by managing node agents and data. Node Agent Management oversees the deployment and configuration of monitoring agents, while Node Data Management handles energy, resource, connection status data, and workload profiles.

Node Remaining Usage Life Management: A critical module in the cloud part, Node Remaining Usage Life Management predicts the remaining energy of nodes and determines suitable recovery policies. This module triggers appropriate actions to ensure continued operation and resilience by analyzing node data and applying recovery policies.

In this paper, with the implementation, we emphasize scaling in action, particularly scaling workload on current edge nodes. The architecture's ability to dynamically adjust workload allocation based on real-time data and predictive analytics enables efficient resource utilization and energy management. In summary, the architecture for energy-aware resilience in mobile nodes leverages machine learning and dynamic resource management to optimize energy consumption, ensure system resilience, and facilitate efficient workload scaling.

2.2 Data Monitoring

In our proposed architecture for energy-aware resilience in mobile nodes, efficient data monitoring at the mobile node is crucial for gathering real-time insights into energy consumption, resource utilization, and network connectivity status. This description outlines the mechanisms, cycle, and collection/storage mechanism employed for data monitoring, especially in scenarios where the connection is disconnected from the anchor node. The data monitoring mechanism at the mobile node operates through specialized monitoring agents deployed within the node's software infrastructure. These agents are responsible for continuously collecting, processing, and transmitting relevant data metrics to the cloud-based management layer for analysis and decision-making. The monitoring agents include the Node Energy Collector, which monitors the energy consumption of the mobile node, capturing fluctuations in power usage over time. Node Resource Collector: Tracks resource utilization metrics such as CPU usage, memory usage, and storage capacity, providing insights into the node's computational workload. Node Connection Status Collector: Monitors the mobile node's connectivity status and network conditions.

When the mobile node becomes disconnected from the anchor node or experiences intermittent connectivity issues, a robust collection and storage mechanism is employed to prevent data loss and ensure data integrity. The mobile node utilizes local storage resources to buffer collected data during periods of disconnection, using techniques such as circular buffers or temporary caches to manage storage efficiently. Once the connection is restored, the buffered data is transmitted to the cloud-based management layer, ensuring the continuity of the data monitoring and analysis process.

2.3 Energy-aware Scheme for Mobile Node

This section presents the proposed energy consumption prediction techniques based on the Long-Short Term Memory[17] model to predict the remaining energy at the edge node when it connects to the anchor node to check its health.

2.3.1 Node Energy Consumption Consideration

Many studies have been conducted on building a node's energy consumption model for this issue[15,16,18,19]. Therefore, in this study, we propose to consider the simplest power consumption model with energy factors affecting the operation and lifespan of mobile nodes while operating in an edge environment with dynamic network conditions.

At any given time t, the remaining energy of the mobile node $E_{rm}^{(t)}$ is determined by the following:

$$E_{rm}^{(t)} = E_{rm}^{(t-1)} - E_{con}^{(t)} \quad with\ t \geq 1$$
$$E_{rm}^{0} = C\ (Maximum\ Energy\ Capacity\ of\ Node) \tag{1}$$

The energy consumption $E_{con}^{(t)}$ can be calculated by:

$$E_{con}^{(t)} = E_{rx}^{(t)} + E_{tx}^{(t)} + E_{proc}^{(t)} + E_{idle}^{(t)} + E_{listen}^{(t)} + E_{mobility}^{(t)} \tag{2}$$

In which, $E_{rx}^{(t)}$ and $E_{tx}^{(t)}$ represent the energy consumed for data transmission. $E_{proc}^{(t)}$ signifies the energy used for processing tasks, $E_{idle}^{(t)}$ denotes the energy required to maintain an idle state of the node, $E_{listen}^{(t)}$ accounts for connection establishment or re-establishment energy, and $E_{mobility}^{(t)}$ represents the energy expended due to node moving.

In the dynamic landscape of edge-cloud environments, understanding the intricacies of mobile node behavior and workload characteristics is crucial for devising effective energy-aware schemes. By comprehensively characterizing mobile nodes and the workloads they undertake, we can meticulously manage energy consumption associated with data transmission, workload processing, and idle states. However, the unpredictable nature of network conditions in unstable edge environments introduces significant challenges. Frequent disconnections and reconnections between mobile nodes and anchor nodes escalate energy consumption, while node mobility further compounds the energy overhead. These factors emerge as critical determinants in crafting resilient energy-aware architectures for mobile nodes in edge-cloud environments. Mitigating network instability's adverse effects and node mobility is paramount in optimizing energy utilization and enhancing system resilience. Therefore, in this paper, we propose utilizing machine learning techniques to predict mobile nodes' energy consumption (node's energy remaining).

2.3.2 Machine Learning-based energy consumption prediction

In this paper, we propose using the Long-Short-Term Memory model (LSTM) for energy consumption prediction. LSTM, a recurrent neural network[21](RNN), is well-suited for sequential data modeling, making it ideal for capturing temporal dependencies in energy consumption patterns. Unlike traditional RNNs, LSTM networks can effectively address the vanishing gradient problem and retain long-term dependencies through their gated architecture, including forgetting, input, and output gates, as shown in Figure 2.

The LSTM model's ability to capture short-term fluctuations and long-term trends in energy consumption dynamics aligns well with the requirements of edge-cloud environments characterized by intermittent network connectivity status and varying workload demands.

In this paper, the LSTM model integrates two crucial metrics as input: *node connection status data (N)*, which is the number of times that the mobile node disconnected from the cloud nodes in the monitoring time interval, and the *node energy remaining data* at a time $(t-1)$. Including these metrics is essential for capturing the dynamic interplay between network connectivity and energy utilization, significantly impacting mobile nodes' energy dynamics in edge-cloud environments. *Node connection status data* serves as a proxy for network stability and reliability, influenced by factors such as energy consumption for connection establishment ($E_{listen}^{(t)}$ This metric provides valuable insights into the temporal variability of network conditions, enabling the LSTM model to adjust its predictions adaptively based on evolving connectivity patterns. Additionally, incorporating the node energy consumption data at time (t−1) allows the LSTM model to leverage historical energy usage information to forecast future energy levels. By considering the energy remaining in the previous time

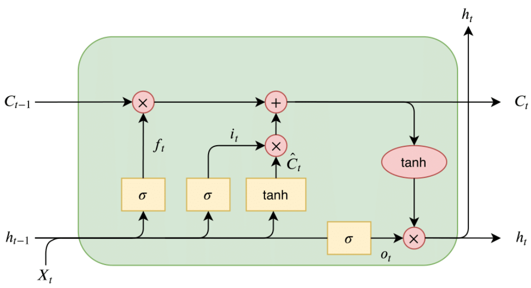

Fig. 2. Structure of LSTM model[20]

step, the model can capture temporal dependencies and trends in energy consumption dynamics, thus enhancing the accuracy of predictions. The output of the LSTM model is the predicted energy remaining of the mobile node at the next time step. This predicted value serves as a critical input for the threshold-based control policy to enhance node resilience. By proactively estimating the energy level of the mobile node, the model enables timely decision-making regarding node management strategies.

Integrating the LSTM-based energy consumption prediction model with a threshold-based control policy enables proactive energy management and resilience enhancement for mobile nodes in edge-cloud environments. The model's adaptive nature facilitates real-time decision-making, mitigating the adverse effects of uncertain factors such as network instability and energy constraints on node performance and longevity.

2.4 Threshold-based Control Policy for Node Resilience

In this section, we present a proposed method for auto-resilience of nodes in an energy-aware edge cloud environment, leveraging the predicted values obtained from the LSTM model. Given our understanding of the characteristics of mobile nodes and their energy consumption patterns, we can define thresholds for key actions, namely, staying at the current location, moving to an anchor node, and workload scaling, to minimize energy consumption while ensuring node resilience. The proposed mechanism for threshold-based policy control is outlined as follows in Figure 3.

To determine what actions should be taken, the threshold-based control policy relied on the Node's energy reduction ratio following:

$$Node's\ energy\ reduction\ ratio\ \alpha = \frac{E_{rm}^{(t)}(pred)}{C} \tag{3}$$

Assume that we need an arbitrary percent of the total energy from the node's battery, which is the minimum for keeping ping node survival in strict

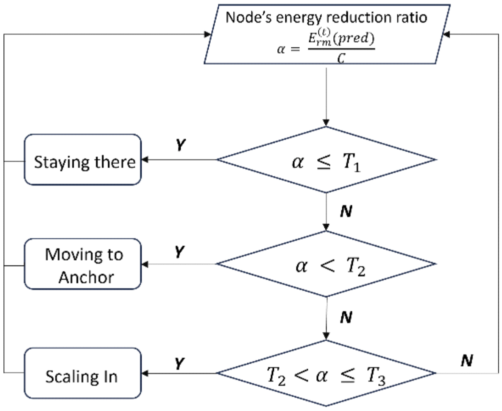

Fig. 3. Threshold-based control policy for auto resilience strategy

conditions. Then, in this proposal, control thresholds for mobile nodes can be defined following as: (i) T_1: If the node's remaining energy reaches this threshold, the policy agent will take action to keep the node at its current location. Then, the mobile node will remain at its current location to conserve energy and avoid unnecessary movements. (ii) T_2: Another threshold determines when the node should initiate movement towards an anchor node or energy source. If the predicted energy remaining falls below this threshold, indicating imminent energy depletion, the node will proactively relocate to an anchor node to replenish its energy reserves. (iii) T_3: Similarly, this threshold is set for workload scaling based on predicted energy levels. When the predicted energy remaining crosses a predefined threshold, the node may dynamically adjust its workload to minimize energy consumption, redistributing processing tasks or reducing computational intensity.

By employing this threshold-based control policy, mobile nodes can autonomously manage their energy resources, optimizing energy utilization while ensuring resilience in dynamic edge-cloud environments. The algorithmic approach facilitates proactive decision-making, enhancing system efficiency and prolonging the operational lifespan of mobile nodes.

Ⅲ. Experimental Evaluation

This section presents the experiments to simulate edge-cloud scenarios, considering the network unstable factor and node energy consumption.

3.1 Experimental Setup

First, we have acquired real data on energy consumption from the Volvo electric bus in Gothenburg – Sweden[22] to incorporate into our experiment. The energy consumption pattern of the electric bus is shown in Figure. 4. This dataset was collected from buses running in real-world conditions. The bus route encompasses twelve stops, mirroring the movement and stopping behavior of the mobile edge nodes under our consideration. Furthermore, various influencing factors, such as curb weight, payload, powertrain configuration, and weather changes, are similar to mobile edge nodes' characteristics, including dynamic workload deployment, network connection status, and distance moved. We, therefore, retrieve this real data for training and validation of the LSTM model for energy consumption prediction and to employ the predicted value in the evaluation of the threshold-based node resilience schemes, with further processing by mapping to the remaining energy of the node corresponding to network connection status, as shown in Figure. 5.

In the experiment setup, we utilize a dataset with two main metrics: node energy remaining over time and network connection status. The node energy remaining data consists of 320 data points inferencing

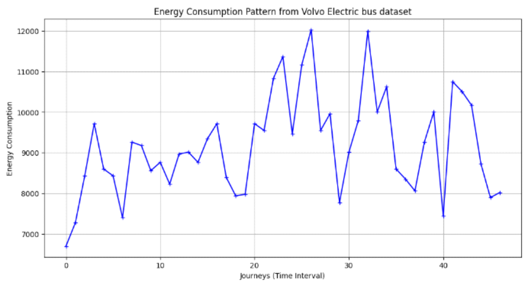

Fig. 4. Energy consumption pattern of the electric bus from the real environment[22]

from real energy consumption data patterns of the electric bus dataset, with the energy level decreasing from 100 to 0. The network connection status is represented by STABLE (indicated by a pulse signal of 1) and Unstable (indicated by a pulse signal of 0).

To augment the dataset and enhance its robustness, we employ bootstrapping techniques. Bootstrapping involves resampling the original dataset with replacement to create multiple new data points. In this case, we perform bootstrapping four times, generating additional data instances while preserving the underlying characteristics of the original dataset. Following the bootstrapping process, we split the augmented dataset into training and testing subsets using a ratio of 70:30. The training data, comprising 70% of the total dataset, is used to train the LSTM model for energy consumption prediction. Meanwhile, the testing data, consisting of the remaining 30% of the dataset, is reserved

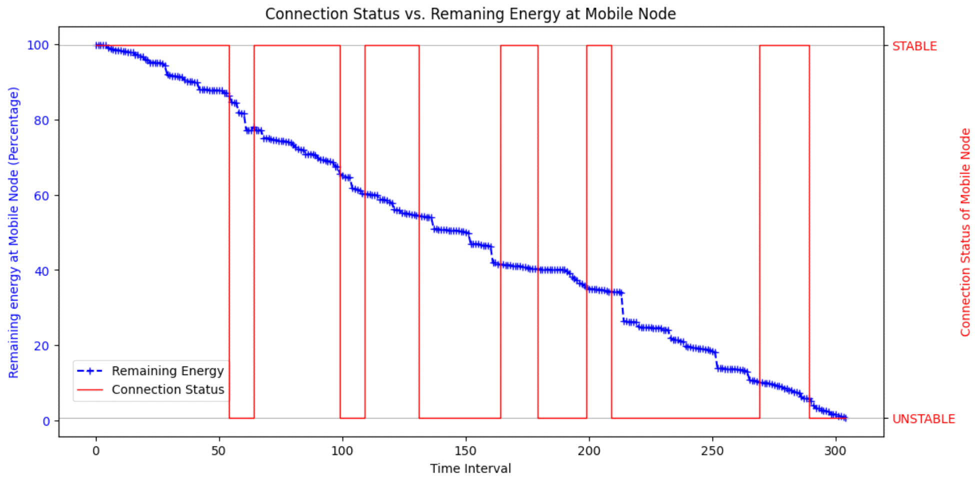

Fig. 5. Connection Status and Remaining Energy data pattern at Mobile Node

for evaluating the performance of the trained model.

3.2 Experimental Results

3.2.1 Prediction Performance

In the first experiment, we evaluate the performance of the LSTM model for energy consumption prediction using two key metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). These metrics provide insights into the accuracy and precision of the model's predictions compared to the ground truth energy consumption values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2,\ MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (4)$$

Where N is the number of data points, y_i is the actual value of energy remaining of the mobile node for the i^{th} data point, and $\hat{y}_i$ is the predicted value of the remaining energy using the LSTM model.

Additionally, we compare the performance of the LSTM model with other baseline models, including Artificial Neural Networks[23] (ANN), Gated Recurrent Units[24] (GRU), Autoencoders[25], and Variational Autoencoders[25] (VAE). The result is shown in Table I.

Table I shows that LSTM has the lowest MSE (0.1276), and MAE (1.0233) values compared to other models. This shows the relative effectiveness of the LSTM model in capturing the temporal dependencies and nonlinear relationships inherent in mobile nodes' energy consumption patterns with network connection status in edge-cloud environments.

Table 1. Comparison of Energy Prediction Performance between models over MSE and MAE

Metrics	ANN	GRU	AE	VAE	LSTM
MSE	0.836	0.701	0.424	0.264	**0.1276**
MAE	3.127	2.922	2.634	2.223	**1.0233**

3.2.2 Energy-Aware workload scaling for minimizing energy consumption of mobile node

In this experiment, we measure the energy efficiency achieved by the threshold-based control policy, quantifying the energy saved through proactive energy-aware resilience management strategies such as staying at the current location, moving to anchor nodes, and scaling workload.

In the second experiment, we focus on assessing the energy efficiency of mobile nodes through energy-aware workload scaling strategies, particularly when nodes run under low-energy conditions (defined as energy remaining <20% (T_3) of total node energy capacity). The experiment evaluates the node's ability to improve energy consumption by implementing different scaling strategies in response to dwindling energy reserves. We consider three distinct workloads, labeled A, B, and C, each characterized by their energy consumption rate per unit of time and the number of replicas needed to sustain it, as shown in Table 2. Workloads A, B, and C represent varying computational intensities and resource demands, reflecting real-world scenarios in edge-cloud environments such as data processing, transmission, and storage.

When the node's energy level falls below the 20% threshold, showing a critical energy state, the experiment triggers energy-aware workload scaling to minimize energy consumption. In this experiment, we evaluate three scaling strategies, including (i) *Step Scale*: This strategy involves gradually reducing the number of replicas by one unit in each scaling cycle, aiming to incrementally adjust the workload to conserve energy resources while maintaining operational functionality. (ii) *Scale to Minimum*: In this strategy, the node simultaneously scales to the minimum number of replicas required to sustain the workload, minimizing energy consumption and avoiding unnecessary resource allocation. And (iii) *No Scale*: Under this strategy, the node keeps the current number of replicas without initiating any scaling actions. The result is shown in Figure 6.

As shown in Fig. 6, the energy remaining in the blue line reduces quickly over time and reaches zero.

Table 2. Workload's energy consumption characteristic.

Workload	Energy consumes. (energy unit per time)	Initial replicas
A	0.2	3
B	0.4	3
C	0.1	3

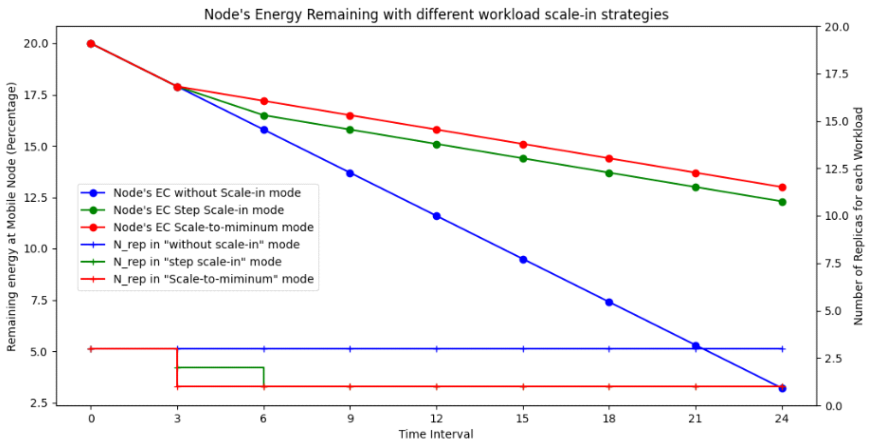

Fig. 6. Node's Energy remaining with different workload scale-in strategies

In this case, the number of *replicas* stays unchanged; therefore, the node's energy consumption drops rapidly if no scaling strategy exists. On the contrary, the red and green lines apply scale-to-minimum and step scale, respectively. We can see that the blue and red lines have a negligible difference in the node's remaining energy after 24 time units, which is above 10% enough to keep the node alive or move to the nearest anchor. Without applying the scale-in policy (blue line), a node's energy can drop rapidly if it is in a low-energy state. By implementing these scaling strategies and monitoring energy consumption patterns, we can optimize energy utilization and prolong the operational lifespan of mobile nodes in low-energy scenarios.

3.2.3 Threshold-based Energy-aware control policy for increasing node survival time

Finally, we investigate the effectiveness of a Threshold-based Energy-aware policy in increasing the survival time of mobile nodes under low-energy conditions. The experiment aims to show the impact of policy-based energy management on node longevity compared to scenarios without policy intervention.

To experiment, we deliberately reduce the energy consumption of mobile nodes to critically low levels, simulating challenging operating conditions where nodes operate near the lower bounds of their energy reserves—reducing threshold T1 (20% of total node energy) to start scaling with step scaling and leading to threshold T3 (5% of total node energy) to keep the node in the current location and going in idle mode. This reduction in energy availability is a stress test for evaluating the resilience and survivability of mobile nodes in energy-constrained environments. Then, we compare two scenarios with and without threshold-based energy-aware control policy.

The comparison results are shown in Figure 7, where we can see that the node's energy begins to decrease rapidly when the remaining energy is below 20% without applying the scaling strategy (orange line). As a result, the node's lifespan ends prematurely at about 250 time intervals. On the other hand, node lifetime is improved when the threshold-based energy-aware control policy is applied, as shown on the blue line. The node's remaining energy is maintained with a step-scale-in strategy, extending the node's lifespan to the lowest threshold (5% of total node energy) as defined by the cyan line. The node's energy remains at this level, ensuring minimum operations of the mobile node to locate and intervene in case of need. In addition, preliminary results indicate that mobile nodes equipped with the threshold-based energy-aware policy exhibit significantly longer survival times than nodes operating without policy intervention. Specifically, the policy-based control mechanism increases node survival time by 80-time units, highlighting the tangible benefits of proactive energy management in enhancing node resilience and

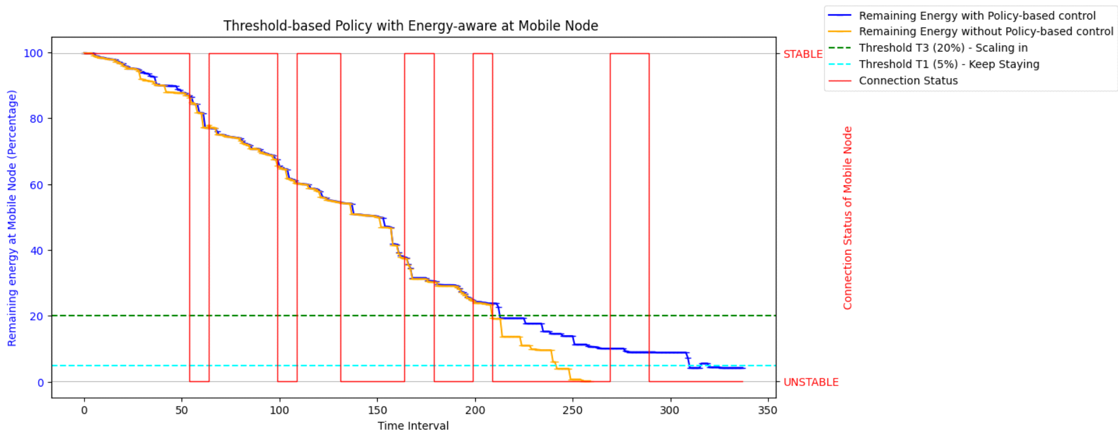

Fig. 7. Comparison of with and without threshold-based energy-aware control policy

prolonging operational lifespan under low-energy conditions.

Ⅳ. Conclusion and Future Works

In the context of mobile edge-cloud environments, the seamless operation of mobile nodes is crucial for various applications, from IoT (Internet of Things) to military operations. However, these nodes' inherent mobility and the environment's dynamic nature pose significant challenges to node resilience. In this study, we proposed an energy-aware architecture for enhancing mobile nodes' resilience and energy efficiency in edge-cloud environments. Leveraging machine learning-based energy consumption prediction techniques and threshold-based control policies, our architecture enables adaptive energy management and optimizing node performance in dynamic conditions. Through experimental evaluations with the data from a real-world environment, we demonstrated the effectiveness of our approach in prolonging node longevity and minimizing energy consumption under varying workload demands and network conditions. Our results underscore the importance of proactive policy-based control mechanisms in ensuring continuous and reliable operation of mobile nodes in edge-cloud environments, laying the groundwork for resilient and efficient edge-cloud systems supporting diverse applications.

In future research, we will explore advanced optimization techniques to enhance energy efficiency within mobile edge-cloud environments. This exploration may encompass refining and developing mechanisms tailored to optimize energy consumption patterns dynamically. Additionally, we aim to investigate the integration of context-awareness techniques to facilitate adaptive node behavior. These techniques may include using Reinforcement Learning (RL) algorithms, which enable mobile nodes to learn and adapt their actions based on environmental and performance feedback. By incorporating context awareness into our architecture, we look to enhance the adaptability and resilience of mobile nodes, enabling them to respond intelligently to changing network conditions and workload demands.

References

[1] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, "Edge computing: Current trends, research challenges, and future directions," *Comput.*, vol. 103, pp. 993-1023, 2021.

[2] Y. Mansouri and M. A. Babar, "A review of edge computing: Features and resource virtualization," *J. Parallel and Distrib. Comput.*, vol. 150, pp. 155-183, 2021.

[3] W. Z. Khan, et al., "Edge computing: A survey," *Future Generation Comput. Syst.*, vol. 97, pp. 219-235, 2019.

[4] V. Prokhorenko and M. Ali Babar, "Architectural resilience in cloud, fog and

edge systems: A survey," in *IEEE Access*, vol. 8, pp. 28078-28095, 2020. (https://doi.org/10.1109/ACCESS.2020.2971007)

[5] S. Li, C. Li, Y. Huang, B. A. Jalaian, Y. T. Hou, and W. Lou, "Enhancing resilience in mobile edge computing under processing uncertainty," in *IEEE J. Sel. Areas in Commun.*, vol. 41, no. 3, pp. 659-674, Mar. 2023. (https://doi.org/10.1109/JSAC.2023.3234706)

[6] F. S. Abkenar, et al., "A survey on mobility of edge computing networks in IoT: State-of-the-art, architectures, and challenges," in *IEEE Commun. Surv. & Tuts.*, vol. 24, no. 4, pp. 2329-2365, Fourthquarter 2022. (https://doi.org/10.1109/COMST.2022.3211462)

[7] R. Singh, R. Sukapuram, and S. Chakraborty, "A survey of mobility-aware multi-access edge computing: Challenges, use cases and future directions," *Ad Hoc Netw.*, vol. 140, p. 103044, 2023.

[8] H. Trinh, et al., "Energy-aware mobile edge computing and routing for low-latency visual data processing," in *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2562-2577, Oct. 2018. (https://doi.org/10.1109/TMM.2018.2865661)

[9] T. Fan, Y. Qiu, C. Jiang, and J. Wan, "Energy aware edge computing: A survey," in *High-Performance Comput. Appl. in Numerical Simulation and Edge Comput.: ACM ICS 2018 Int. Wkshps., HPCMS and HiDEC*, Beijing, China, Jun. 12, 2018, Revised Selected Papers 2, pp. 79-91, Springer Singapore, 2019.

[10] G. Perin, F. Meneghello, R. Carli, L. Schenato, and M. Rossi, "EASE: Energy-aware job scheduling for vehicular edge networks with renewable energy resources," in *IEEE Trans. Green Commun. and Netw.*, vol. 7, no. 1, pp. 339-353, Mar. 2023. (https://doi.org/10.1109/TGCN.2022.3199171)

[11] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," in *IEEE Commun. Surv. & Tuts.*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017. (https://doi.org/10.1109/COMST.2017.2745201)

[12] S. K. Zaman, A. I. Jehangiri, T. Maqsood, N. Haq, A. I. Umar, J. Shuja, Z. Ahmad, I. B. Dhaou, and M. F. Alsharekh. "LiMPO: Lightweight mobility prediction and offloading framework using machine learning for mobile edge computing," *Cluster Comput.,* vol. 26, pp. 99-117, 2023.

[13] C.-L. Wu, T.-C. Chiu, C.-Y. Wang, and A.-C. Pang, "Mobility-aware deep reinforcement learning with glimpse mobility prediction in edge computing," *ICC 2020 - 2020 IEEE ICC*, pp. 1-7, Dublin, Ireland, 2020. (https://doi.org/10.1109/ICC40277.2020.9149185)

[14] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. M. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," in *IEEE Internet of Things J.*, vol. 9, no. 2, pp. 1517-1530, Jan. 2022. (https://doi.org/10.1109/JIOT.2021.3091142)

[15] L. Ale, N. Zhang, X. Fang, X. Chen, S. Wu, and L. Li, "Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning," in *IEEE Trans. Cognitive Commun. and Netw.*, vol. 7, no. 3, pp. 881-892, Sep. 2021. (https://doi.org/10.1109/TCCN.2021.3066619)

[16] S. Gupta and J. Chakareski, "Lifetime maximization in mobile edge computing networks," in *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3310-3321, Mar. 2020. (https://doi.org/10.1109/TVT.2020.2965440)

[17] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," in *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114-119, Jun. 2019. (https://doi.org/10.1109/MCOM.2019.1800155)

[18] A. Cañete, K. Djemame, M. Amor, L. Fuentes, and A. Aljulayfi, "A proactive energy-aware auto-scaling solution for edge-based infrastructures," *2022 IEEE/ACM 15th Int. Conf. Utility and Cloud Comput.*

(UCC), pp. 240-247, Vancouver, WA, USA, 2022.
(https://doi.org/10.1109/UCC56403.2022.00044)

[19] Q. Zhang, X. Lin, Y. Hao, and J. Cao, "Energy-aware scheduling in edge computing based on energy internet," in *IEEE Access*, vol. 8, pp. 229052-229065, 2020.
(https://doi.org/10.1109/ACCESS.2020.3044932)

[20] Colah's blog, "Understanding LSTM Network," Posted on Aug. 27, 2015, Available on https://colah.github.io/posts/2015-08-Understanding-LSTMs/, Last accessed Apr. 29th 2024.

[21] L. R. Medsker and L. C. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[22] R. Basso, B. Kulcsár, and I. Sanchez-Diaz, "Electric vehicle routing problem with machine learning for energy prediction," *Transport. Res. Part B: Methodological*, vol. 145, pp. 24-55, 2021.

[23] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshadm, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, 2018.

[24] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *2017 IEEE 60th MWSCAS*, pp. 1597-1600, 2017.

[25] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its various variants," in *2018 IEEE Int. Conf. Syst., Man, and Cybernetics (SMC)*, pp. 415-419, 2018.

JangWon Lee

Feb. 2020 : B.S. degree, Soongsil University
Feb. 2022 : M.S. degree, Soongsil University
Mar. 2022~Current : Ph.D. student, Soongsil University
<Research Interests> Cloud computing, Autonomous, and 5G/6G network infrastructure
[ORCID:0009-0002-6194-2520]

YoungHan Kim

Feb. 1984 : B.S. degree, Seoul University
Feb. 1986 : M.S. degree, KAIST
Feb. 1990 : Ph.D. degree, KAIST
Mar. 1994~Current : Professor, Soongsil University
<Research Interests> ICT technology, wireless communications, data communications
[ORCID:0000-0002-1066-4818]

Dooho Keum

Feb. 2015 : M.S. degree, Ajou University
Feb. 2020 : Ph.D. degree, Ajou University
Jan. 2022~Current : Senior Research Engineer, LIG Nex1
<Research Interests> Military IoT, Trust based Routing, and Wireless Ad-hoc/Mesh/Sensor Networks
[ORCID:0000-0002-8267-2331]

Gyu-min Lee

Feb. 2014 : B.S. degree, Ajou University

Feb. 2016 : M.S. degree, Ajou University

Aug. 2022 : Ph.D. degree, Ajou University

Feb. 2023~Current : Senior Research Engineer, LIG Nex1

<Research Interests> C5ISR, Tactical network architecture, SDN/NFV

[ORCID:0000-0002-6384-795X]

Myoung-hun Han

Feb. 2007 : B.S. degree, Chung-Ang University

Aug. 2009 : M.S. degree, Chung-Ang University

Aug. 2021 : Ph.D. degree, Chung-Ang University

Oct. 2014~Current : ADD

<Research Interests> Tactical Network, Network M&S, Satellite Network

Suil Kim

Feb. 1986 : B.S. degree, Soongsil University

Feb. 1988 : M.S. degree, Soongsil University

Aug. 2000 : Ph.D. degree, KAIST

Mar. 1988~Current : ADD

<Research Interests> Military Tactical Satellite, Wireless communications, Network Centric Operation Environment