On-Device AI for Maritime Communication: Trends, Challenges, and a Simulation-Only Case Study

1st Shrutika Sinha
Dept. of Financial Information Security
Kookmin University
Seoul, South Korea
shrusinha09@kookmin.ac.kr

2nd Soo-Hyun Park
School of Computer Science
Kookmin University
Seoul, South Korea
shpark21@kookmin.ac.kr

***Abstract*—This paper reviews key challenges and emerging trends in On-Device AI, including model efficiency, energy constraints, and communication computation trade-offs in edge systems. Emerging directions such as TinyML, lightweight training, and hardware-aware optimization are discussed. In addition to the survey, a simulation case study is given in which an on-device representation of reinforcement learning is used to select adaptive Wi-Fi/LTE links using the relaying maritime environment with battery and channel variability. The findings have shown that learning-based policies are able to offer context-dependent decision-making as opposed to fixed heuristics and that On-Device AI has a potential in the future to be used in wireless and resource-constrained settings.**

***Index Terms*—Edge AI, NTN, On-Device AI, Reinforcement Learning.**

I. Introduction

The convergence of artificial intelligence (AI), edge computing, and the Internet of Things (IoT) is transforming data processing and response [1]. Instead of depending solely on cloud servers, computation is being more and more transferred to edge devices, smartphones, embedded sensors, and industrial controllers, allowing for ultra-low latency, increased data privacy, and scalability for enormous numbers of devices [2].

On-device AI has emerged as the foundation for achieving intelligent and autonomous systems in next-generation 6G and Non-Terrestrial Network (NTN) architectures [3]. With the abundance of IoT devices, lightweight, adaptive, and self-learning intelligence becomes imperative in settings with intermittent or unreliable connectivity [4].

On-Device AI represents the next evolution of this transformation. It means machine learning inference execution and (with growing frequency) adaptive or federated training at the location of the hardware, not directly connected to the cloud [5]. In contrast to traditional edge computing where the offloading of computation is done to the close by server or gateway, On-Device AI is an intelligence that is imbued directly into the device. The endpoints are made to be self-contained decision-making units with the ability to perceive, reason and control enabling systems to have the ability to run autonomously even with untrustworthy or intermittent connectivity [6]. Figure 1 shows the features of On-Device AI. On-Device AI is characterized by the following features:

1. Real-time responsiveness, which allows real-time decision-making on the tasks that have a high latency, like voice recognition, image analysis, or anomaly detection [7].

2. Resource efficiency, obtained by optimal utilization of CPU, memory, and power by methods such as model compression, pruning, and quantization [8].

3. Viewing as privacy conservation, as the raw data is kept in the device, which makes it less exposed to interception and compromising with contemporary data-protection models [9].

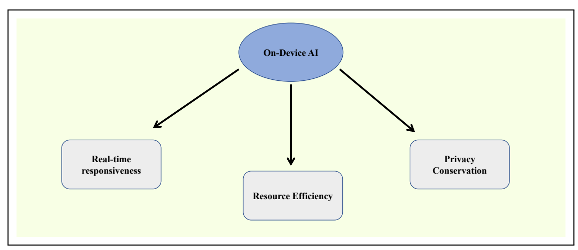

Fig. 1. On-Device AI features

This paradigm is related to the vision of native AI in 6G systems and NTN systems, where communication, sensing, and computation are co-designed to form a context-aware and self-optimizing network entity. In this paper, we will offer a brief and yet an in-depth review of the current state of the On-Device AI with references to new developments, current problems, and future tendencies that will influence the development of intelligent communication systems in the next decade in maritime sector. In addition to the survey, the paper includes a simulation case study demonstrating an on-device reinforcement learning agent performing adaptive Wi-Fi/LTE link selection in a maritime relay scenario. While On-Device AI can be applied across many application domains, this work specifically focuses on its relevance to maritime communication systems, where connectivity is intermittent, power is limited, and centralized control is not always feasible. This example illustrates how the concepts discussed translate into practical decision-making under real-world constraints such as fluctuating channel quality and limited battery capacity. The simulation case study presented in this paper is lightweight and abstract, and could not be considered a comprehensive performance benchmark, but a conceptual expression of how on-device reinforcement learning can encode energy throughput trade-offs of maritime communication environments.

II. Challenges in On-Device AI

Despite rapid progress in algorithms and hardware, the deployment of AI models directly on devices continues to face several enduring technical and systemic challenges. These constraints define the current research frontier and highlight the gap between laboratory prototypes and large-scale, real-world adoption. Figure 2 represents the challenges in On-Device AI.

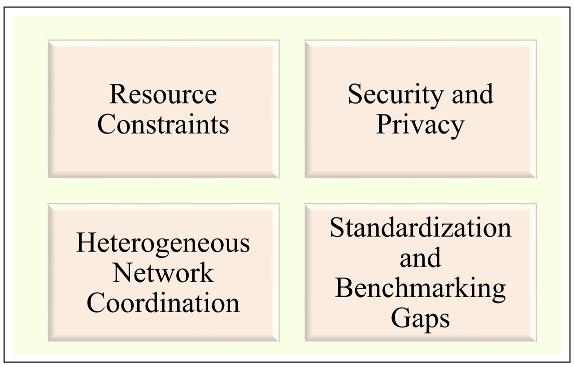

Fig. 2. Challenges of On-Device AI

A. Resource Constraints

Embedded and IoT devices are limited by their computational capacity as well as memory and battery life, which is the most important limitation [10]. Deep neural networks are often trained by using high-performance requiring billions of parameters and requiring extensive floating-point operations, which micro controllers and mobile processors cannot compute [11]. Despite the fact that methods that include compressing models, pruning, quantization, and neural architecture search have addressed those concerns, there are always trade-offs between accuracy and efficiency. One of the current optimization problems is how to achieve real-time inference on devices with small dynamic power budgets [12].

B. Security and Privacy

While on-device processing inherently preserves privacy by not storing data centrally, it also introduces new risks [13]. Local models are vulnerable to model inversion, adversarial perturbation, and side-channel attacks that can extract sensitive information or bias outputs [14]. The problem is one of balancing cryptographic security with latency constraints especially for time-sensitive applications like autonomous driving or medical monitoring [15].

C. Heterogeneous Network Coordination

On-device intelligence tends to run in challenging, heterogeneous network environments that include Wi-Fi, 5G/6G, satellite, and NTN connections [16]. Distributed learning and inference coordination over such diverse connections creates synchronization delays, packet loss, and asymmetric resource availability [17]. In federated or collaborative learning systems, devices with different connectivity quality or computation rate can severely impact convergence rates [18]. Research on adaptive aggregation, dynamic clustering, and communication-efficient learning protocols is working to alleviate these discrepancies, but strong solutions for highly dynamic topologies such as vehicular or maritime networks are still lacking [19].

D. Standardization and Benchmarking Gaps

In contrast to cloud-based AI, on-device AI has no common standards for assessing performance across multiple hardware and workloads. There is no agreement on benchmarking measures that together capture energy efficiency, latency, inference accuracy, and model robustness under practical constraints [20]. The lack of such standardized frameworks makes cross-platform comparison difficult and delays the advancement of optimizing algorithms for heterogeneous edge devices [21].

III. Emerging Trends And Research Directions

As on-device artificial intelligence evolves further, a number of cutting-edge trends are redefining its research and deployment landscape. These advancements not only push the technical limits of on-device computation but also redefine the ways in which intelligence engages with communication, sensing, and environmental limitations.

A. TinyML

TinyML is among the most active frontiers in embedded intelligence [22]. It is centered on running deep learning models in a memory footprint usually below one megabyte, which makes it possible to apply sophisticated analytics on microcontrollers and battery-powered IoT devices [23]. With extreme quantization, sparse matrix computations, and compiler optimization, TinyML systems are currently able to run object detection, keyword spotting, or anomaly detection fully offline [24].

B. Edge Foundation Models

Building upon large pre-trained models, edge foundation models, which adapt these general-purpose networks to localized, domain-specific conditions [25]. Lightweight fine-tuning and transfer learning can be done on or close to the device rather than in large computer clusters that are used to regional training [26]. This method boosts few-shot and continual learning, which enables AI systems to adapt behavior using personalization without retraining.

C. Maritime, Polar, and Extreme-Environment Intelligence

With the connectivity spanning outside of terrestrial limits, extreme environment on-device artificial intelligences have emerged as an important research theme [27]. The needs of the maritime, polar and desert deployments are ultra-robust models that can be used in autonomous operation but with extreme bandwidth constraints, fluctuating signal quality and high temperatures.

Such scenarios require a 2-way communication between the energy-gathering components and dynamically-adjusting adaptive AI models that respond to the power availability. This autonomous intelligence aids applications such as unmanned marine vehicles (UMVs), remote sensor buoys, and polar networks. These applications are examples of native AI being used with NTN systems, in which communication and cognition are co-evolved to provide reliability in limited conditions.

IV. Simulation Case Study

A. Scenario Description

To prove the relevance of On-Device AI to communication-constrained settings, a simulation-based maritime relay scenario is taken into consideration. The system includes a robot buoy that has two heterogeneous wireless RF communication: Wi-Fi and LTE, that are used as a single-hop relay to a vessel. The buoy is battery constrained and intermittent wireless by virtue of mobility and fading of channels. The aim of the device is to automatically choose the most appropriate interface to transmit data in the consideration of throughput, channel conditions, and energy consumption.

B. Environment Design

The simulation was implemented in Python using the Gymnasium RL framework. The environment state is represented as a four-dimensional vector:

$$S_t = [d_t,\ \mathrm{RSSI}_t,\ B_t,\ Q_t] \tag{1}$$

where d_t is the distance between nodes, RSSI_t denotes the measured Wi-Fi signal strength, B_t represents the remaining battery percentage, and Q_t denotes the transmission buffer size. The agent selects one of three actions.

$$A_t \in \{\text{Wi-Fi},\ \text{LTE},\ \text{Sleep}\} \tag{2}$$

The reward function balances throughput, energy cost, and transmission reliability, penalizing excessive LTE usage and buffer overflow.

C. Reinforcement Learning Policy

An Advantage Actor-Critic (A2C) agent was trained for 80,000 timesteps using the Stable-Baselines implementation. A lightweight MLP policy was used to ensure feasibility for later on-device deployment. The acquired policy should dynamically select transmission modes according to the environmental conditions instead of the fixed RSSI thresholds.

D. Baseline Heuristic Method

A rule-based heuristic was implemented for comparison. The heuristic uses Wi-Fi whenever $\mathrm{RSSI} > -80\,\mathrm{dBm}$ and switches to LTE only when the transmission buffer exceeds a predefined threshold. Unlike the RL agent, this method does not consider long-term battery impact or channel variability. The heuristic baseline is selected to represent a commonly used threshold-based strategy, and is not intended to reflect the full range of optimized link-selection algorithms, which are left for future comparative studies.

V. Results And Discussions

Figures below illustrates the performance comparison between the proposed A2C-based on-device agent and a heuristic Wi-Fi/LTE switching rule.

A. Learning Behaviour

As demonstrated by the A2C learning curve in Fig. 3, the stabilization of the learning process rises gradually and eventually stabilizes once the agent experiences about 600 training episodes, which means that the agent has managed to reach a stable decision-making policy. Whereas the fluctuations in rewards exist because of the stochastic fading and buffer dynamics, the trend exhibits the successful learning in the specified environment.

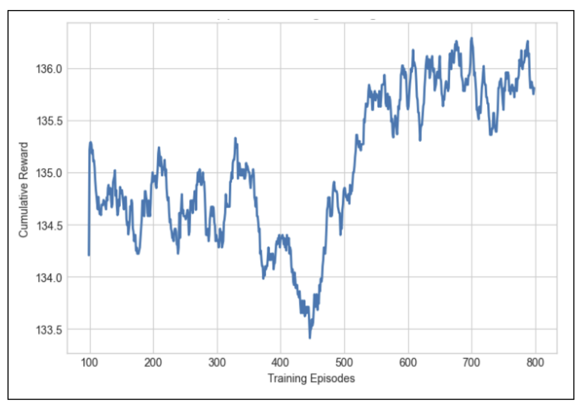

Fig. 3. A2C Learning Convergence

B. Throughput Performance

As shown in Fig. 4, both approaches initially achieve similar throughput while Wi-Fi coverage remains sufficient. However, as channel quality degrades with increasing distance, the heuristic switches to LTE, whereas the A2C agent continues transmitting exclusively using Wi-Fi. Despite avoiding LTE entirely, the A2C agent delivers slightly more total data by the end of the simulation, indicating that consistent use of Wi-Fi was sufficient to satisfy the traffic demand under the modeled conditions.

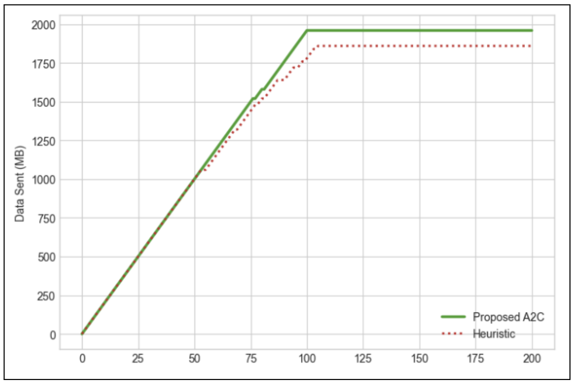

Fig. 4. Cumulative Throughput

C. Link Selection Strategy

Fig. 5 confirms that the A2C policy never selects LTE or sleep actions during evaluation and instead relies entirely on Wi-Fi transmission, even at lower signal strengths near $-80\,\mathrm{dBm}$. This behavior suggests that, under the current reward structure and channel model, the agent learned that LTE offers no meaningful benefit relative to its significantly higher energy cost.

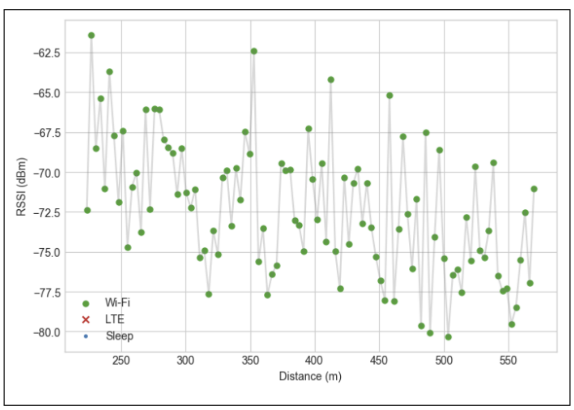

Fig. 5. A2C Link Selection Strategy

D. Battery Consumption

The trends of battery depletion in Fig. 6 depict almost identical decay of both styles until around step 85, where the usage of LTE makes the heuristic use energy a little more rapidly. A2C does not use LTE, so its consumption curve is smoother and a bit more efficient. The conclusions of both methods are that they all end in the total depletion of the battery, which underscores the need to develop energy-conscious strategy refinement in future employment.

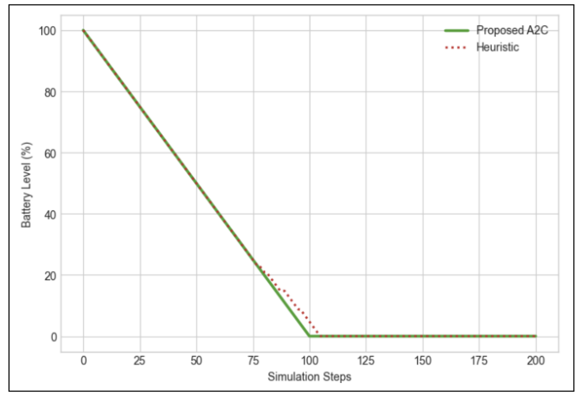

Fig. 6. Battery Depletion Analysis

E. Discussion

Overall, these results confirm the argument that the reinforcement learning policy is not just executing the reproduction of the heuristic decision threshold, but rather it learns a policy that is relevant to the long-term reward maximization. The best decision that will be learned by the agent in this case, that is, in the accurate conditions of the model formulation and the simulated conditions of the cost of the channel under the coastal conditions, is to not use LTE and instead depend on Wi-Fi since the energy cost related to LTE transmission is very high.

This observed behavior is a rational energy saving trade-off that is coded by the reward structure, based on a rational trade-off and not a trivial policy finish, and shows that reinforcement learning is able to discover resource-sensitive communication strategies in a policy guaranteeing an apparently straightforward policy. Notably, the acquired strategy will depend on the presumed propagation, mobility and cost models. Greater propagation loss, higher distance, or different weightings of rewards are likely to become crucial and desirable action at LTE selection.

Future work will therefore extend this assessment to more diverse mobility patterns and channel conditions to evaluate policy adaptability across operating regimes. In addition, embedded deployment tests on resource-constrained platforms such as NVIDIA Jetson Orin will be conducted to further validate the feasibility of on-device reinforcement learning for maritime IoT deployments.

VI. Conclusion

On-device AI is being seen as one of the enablers of autonomous and resilient intelligence in future wireless and IoT systems. The enabling technologies, design issues, and emerging research trends including TinyML, federated learning, and hardware-aware optimization in resource-constrained systems were initially surveyed in this paper. In order to supplement the survey-based understanding, a simulation case study was introduced in a maritime relay environment, where an A2C-based agent was measured in adaptive Wi-Fi/LTE link selection against channel variability and battery limitations.

The findings indicate that the trained policy flows with the encoded trade-offs and only uses LTE when it is necessary rather than a fixed heuristic strategy, indicating how the reinforcement learning can be useful in autonomous communication decisions at the device level. While the findings are limited to a controlled, single-hop simulation model, they provide an early indication of how On-Device AI may support autonomous link management in future maritime IoT deployments. The focus of this paper is not to provide to explore and validate On-Device AI specifically within maritime communication environments as a first step toward broader applicability. Future work will extend the environment to more diverse mobility and fading conditions and deploy the trained policy on embedded hardware such as Jetson Orin to evaluate real-world performance.

ACKNOWLEDGMENT

This research was supported by Korea Institute of Marine Science and Technology Promotion(KIMST) Funded by the Ministry of Oceans and Fisheres(RS-2022-KS221571).

REFERENCES

[1] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," IEEE Internet Things J., vol. 7, no. 8, pp. 7457–7469, Apr. 2020, doi: 10.1109/JIOT.2020.2984887.

[2] I. Ficili, M. Giacobbe, G. Tricomi, and A. Puliafito, "From sensors to data intelligence: Leveraging IoT, cloud, and edge computing with AI," Sensors, vol. 25, no. 6, p. 1763, Mar. 2025, doi: 10.3390/s25061763.

[3] A. Iqbal et al., "Empowering non-terrestrial networks with artificial intelligence: A survey," IEEE Access, vol. 11, pp. 100986–101006, Sep. 2023, doi: 10.1109/ACCESS.2023.3314732.

[4] L. Ismail and R. Buyya, "Artificial intelligence applications and self-learning 6G networks for smart cities digital ecosystems: Taxonomy, challenges, and future directions," Sensors, vol. 22, no. 15, p. 5750, Aug. 2022, doi: 10.3390/s22155750.

[5] X. Wang et al., "Empowering edge intelligence: A comprehensive survey on on-device AI models," ACM Comput. Surv., vol. 57, no. 9, pp. 1–39, Apr. 2025, doi: 10.1145/3724420.

[6] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," IEEE Internet Things J., vol. 8, no. 18, pp. 13849–13875, Sep. 2021, doi: 10.1109/JIOT.2021.3088875.

[7] M. J. Reis and C. Serôdio, "Edge AI for real-time anomaly detection in smart homes," Future Internet, vol. 17, no. 4, p. 179, Apr. 2025, doi: 10.3390/fi17040179.

[8] U. Kulkarni et al., "AI model compression for edge devices using optimization techniques," in Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI, vol. 2, Cham, Switzerland: Springer, 2021, pp. 227–240, doi: 10.1007/978-3-030-68291-017.

[9] J. J. Moon et al., "A new frontier of AI: On-device AI training and personalization," arXiv preprint arXiv:2206.04688, Jun. 2022, doi: 10.48550/arXiv.2206.04688.

[10] M. Capra, R. Peloso, G. Masera, M. R. Roch, and M. Martina, "Edge computing: A survey on the hardware requirements in the Internet of Things world," Future Internet, vol. 11, no. 4, p. 100, Apr. 2019, doi: 10.3390/fi11040100.

[11] Y. Deng, "Deep learning on mobile devices: A review," in Mobile Multimedia/Image Processing, Security, and Applications 2019, vol. 10993, May 2019, pp. 52–66, doi: 10.1117/12.2518469.

[12] E. J. Husom et al., "Sustainable LLM inference for edge AI: Evaluating quantized LLMs for energy efficiency, output accuracy, and inference latency," ACM Trans. Internet Things, Apr. 2025, doi: 10.1145/3767742.

[13] E. Collins and M. Wang, "Federated learning: A survey on privacy-preserving collaborative intelligence," arXiv preprint arXiv:2504.17703, Apr. 2025.

[14] W. Yang et al., "Deep learning model inversion attacks and defenses: A comprehensive survey," Artif. Intell. Rev., vol. 58, no. 8, p. 242, May 2025, doi: 10.1007/s10462-025-11248-0.

[15] L. Shanmugam, R. Tillu, and S. Jangoan, "Privacy-preserving AI/ML application architectures: Techniques, trade-offs, and case studies," J. Knowl. Learn. Sci. Technol., vol. 2, no. 2, pp. 398–420, 2023, doi: 10.60087/jklst.vol2.n2.p420.

[16] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," High-Confidence Comput., vol. 1, no. 1, p. 100008, Jun. 2021, doi: 10.1016/j.hcc.2021.100008.

[17] F. P. Lin, C. G. Brinton, and N. Michelusi, "Federated learning with communication delay in edge networks," in Proc. IEEE GLOBECOM, Dec. 2020, pp. 1–6, doi: 10.48550/arXiv.2008.09323.

[18] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," Comput. Sci. Rev., vol. 50, p. 100595, Nov. 2023, doi: 10.1016/j.cosrev.2023.100595.

[19] C. Qiu et al., "Hierarchical aggregation for federated learning in heterogeneous IoT scenarios: Enhancing privacy and communication efficiency," Future Internet, vol. 17, no. 1, p. 18, 2025, doi: 10.3390/fi17010018.

[20] P. Bartoli et al., "Benchmarking energy and latency in TinyML: A novel method for resource-constrained AI," arXiv preprint arXiv:2505.15622, May 2025, doi: 10.48550/arXiv.2505.15622.

[21] B. Varghese et al., "A survey on edge performance benchmarking," ACM Comput. Surv., vol. 54, no. 3, pp. 1–33, Apr. 2021, doi: 10.1145/3444692.

[22] J. Lin et al., "Tiny machine learning: Progress and futures [feature]," IEEE Circuits Syst. Mag., vol. 23, no. 3, pp. 8–34, Oct. 2023, doi: 10.48550/arXiv.2403.19076.

[23] S. Heydari and Q. H. Mahmoud, "Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions," Sensors, vol. 25, no. 10, p. 3191, May 2025, doi: 10.3390/s25103191.

[24] J. Moosmann, M. Giordano, C. Vogt, and M. Magno, "Tinyissimoyolo: A quantized, low-memory footprint, TinyML object detection network for low power microcontrollers," arXiv preprint arXiv:2306.00001, May 2023, doi: 10.48550/arXiv.2306.00001.

[25] R. Bommasani, "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021, doi: 10.48550/arXiv.2108.07258.

[26] H. I. Liu et al., "Lightweight deep learning for resource-constrained environments: A survey," ACM Comput. Surv., vol. 56, no. 10, pp. 1–42, Jun. 2024, doi: 10.1145/36572.

[27] J. Naoukin, M. Isik, and K. Tiwari, "A survey examining neuromorphic architecture in space and challenges from radiation," arXiv preprint arXiv:2311.15006, Nov. 2023, doi: 10.48550/arXiv.2311.15006.