Analyzing the Heterogeneous Edge AI systems to facilitate the profiling of AI models in achieving efficient computation offloading

Rajvansh Chaudhary

Evergreen Public School, Najafgarh, New Delhi.

Abstract

Edge AI leverages computation offloading to overcome the resource limitations of user devices. However, traditional methods often assume homogeneous infrastructure and neglect the runtime characteristics of AI models, which becomes a critical challenge in heterogeneous edge environments. This paper presents a comprehensive literature review on computation offloading strategies in edge systems and introduces the concept of profiling AI models to enable efficient offloading decisions. Through comparative analysis, we highlight how profiling — capturing parameters such as model type, hyperparameters, hardware specifications, and dataset characteristics — can dramatically improve resource utilization, energy efficiency, and latency. We propose profiling-based approaches informed by prior modelling and optimization techniques to enhance adaptivity in dynamic heterogeneous edge scenarios. A new framework is outlined to guide future research, emphasizing accurate prediction of resource consumption and task completion times. The key insights reaffirm that profiling augments offloading strategies beyond rule-based or optimization-only methods, paving the way toward more responsive and efficient edge AI systems.

1. Introduction

Edge AI systems, where inference tasks are offloaded to nearby edge servers, have become vital to support latency-critical and computation-heavy AI applications on resource-constrained devices. This trend stems from the rapid rise of mobile and IoT devices generating vast data volumes while demanding real-time AI-driven processing. Computation offloading mitigates local device limitations by leveraging edge and cloud resources.

Nevertheless, existing approaches often assume homogeneous hardware and static task profiles, which fails to reflect real-world heterogeneity. Diverse device capabilities, dynamic network conditions, and varying AI model characteristics lead to suboptimal offloading decisions and wasted resources. Moreover, typical optimization or ML-based methods seldom account for fine-grained resource demands specific to AI model configurations.

Profiling offers a promising solution: by measuring and modelling key performance attributes — such as CPU/GPU load, memory usage, runtime latency, and energy — across different models, datasets, and hardware, the system can make informed offloading decisions. Profiling enables adaptive mapping of tasks to edge nodes that minimize energy and latency while handling heterogeneity effectively.

This paper surveys computation offloading strategies from 2012–2022 and demonstrates how profiling can fill existing gaps. We propose a framework where profiling data drives decision-making, enabling more accurate and efficient offloading in heterogeneous edge AI environments.

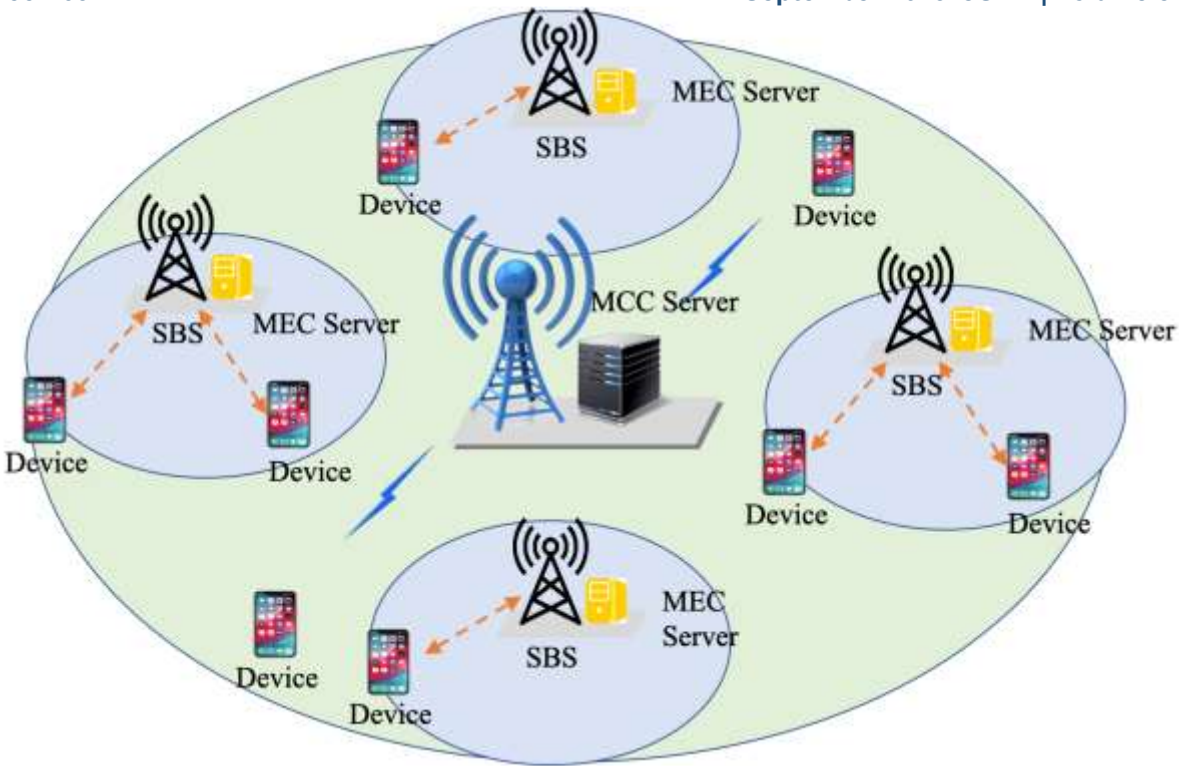

Fig. 1: The scenario of end-edge-cloud heterogeneous networks with multiple users .

2. Related Work

Here, we categorize and review two main classes of offloading strategies: optimization-based methods and ML-based methods. We then present a comparative table.

2.1 Optimization and Heuristic-Based Approaches

- **Asynchronous Mobile-Edge Computation Offloading (2018)**: You *et al.* formulated an optimization problem to minimize mobile energy consumption in asynchronous data arrival and deadline scenarios, deriving optimal time-division and partitioning strategies.
- **Joint Computation and Communication Cooperation (2018)**: Cao *et al.* proposed a cooperative model among user, helper, and MEC servers, optimizing offloading and communication resource allocation to minimize global energy while meeting latency constraints.
- **Optimizing Offloading under Heterogeneous Delay (2022)**: Wan *et al.* addressed wireless powered MEC with tasks having diverse delay requirements, formulating a mixed-integer nonlinear programming problem solved via differential evolution and golden-section search, delivering near-optimal computation rates.
- **Energy-Minimized Partial Offloading (2022)**: Bi *et al.* introduced a hybrid Particle Swarm–Genetic Algorithm (PGL) for heterogeneous networks, minimizing total energy consumption in delay-sensitive applications across mobile devices, small base stations, and macro cells.

2.2 ML-based and Deep Learning Approaches

- **CEDOT (2021)**: Abbas *et al.* designed a deep-learning based offloading technique combining policy selection and task partitioning using a comprehensive cost-function. The trained DNN achieved over 70% accuracy in deciding optimal offloading strategies, reducing energy consumption and delay.
- **Energy-Efficient Offloading under Uncertainty (2022)**: Ji *et al.* proposed an algorithm that models uncertainty with extreme value theory and solves offloading as a DAG-based optimization, minimizing worst-case expected energy consumption with provable bounds.

2.3 Offloading Modelling and Surveys

- **Survey on Offloading Modelling (2020)**: Research outlined key modeling techniques — channel, computation, communication, and energy models — and methods like convex optimization, MDPs, game theory, Lyapunov, and ML-based modeling for offloading.
- **Survey on Deep Reinforcement Learning (2024)**: Peng *et al.* reviewed DRL-based offloading approaches across edge systems, highlighting adaptivity but noting limitations in heterogeneous environment.

2.4 Comparative Table: Key Works

Year	Authors & Citation	Method	Highlights	Limitations
2018	You *et al.*, "Asynchronous MECO"	Convex optimization	Optimal partitioning/time-division for energy-minimization	Assumes homogeneity, static input
2018	Cao *et al.* "Joint Coop. for MEC"	Mixed strategy optimization	Joint resource allocation with helper nodes	Predefined topology, not scalable
2022	Wan *et al.* "WP-MEC heterogeneous delay"	Differential evolution + golden search	Handles heterogeneity in delay requirements	High computation, offline design
2022	Bi *et al.* "PGL offloading"	PGL metaheuristic	Minimizes system-wide energy in heterogeneous networks	Simulation-based, no profiling
2021	Abbas *et al.* "CEDOT"	DNN classifier	Joint offloading & partitioning with high accuracy	Training cost, limited generalization
2022	Ji *et al.* "Uncertainty-aware offloading"	DAG + EVT optimization	Bounds on energy under uncertainty	Complexity, DAG modeling effort

3. Profiling AI Models for Offloading

Profiling captures performance metrics across AI model configurations (e.g., model type, size, hyperparameters), hardware setups, and dataset properties. The recent roadmap "Profiling AI Models..." (2024) by Parra-Ullauri *et al.* proposed capturing metadata to predict resource utilization and task completion times, using over 3,000 experimental runs as evidence.

3.1 Profiling Dimensions and Metrics

We propose profiling along these key dimensions:

Dimension	Examples	Profiling Metrics
Model Type & Architecture	CNN types, MLP, transformer size	FLOPs, parameters count, memory footprint
Hyperparameters	Batch size, optimizer, LR	GPU/CPU usage, latency per batch
Hardware Specifications	CPU/GPU type, RAM, edge node	Throughput, utilization, energy
Dataset Characteristics	Size, resolution, format	I/O cost, preprocessing time

Profiling should be performed on diverse edge hardware variations to build predictive models mapping input dimensions to performance outcomes.

3.2 Integrating Profiling into Offloading

Imagine trying to pack for a trip without knowing the weather or how much space you have — chances are, you'll either overpack or leave something important behind. Computation offloading decisions in edge AI systems often face a similar challenge. Without a clear understanding of how different AI models behave under varying hardware and workloads, decisions about where to offload tasks can be inefficient, wasteful, or even counterproductive.

This is where profiling becomes a game-changer. By systematically collecting data on how AI models perform — like how much memory they use, how long they take to run, or how they behave on different devices — we gain the ability to predict what will happen before a task is offloaded. It's like checking the weather and knowing your luggage size before you pack — the outcome is smarter, more tailored decisions.

When this profiling data is plugged into an offloading system, it enables a whole new level of decision-making:

- **Smarter Predictions**: We can estimate how long a task will take or how much energy it will consume — not just in theory, but based on actual past behavior of similar models and devices.
- **Better Task Placement**: Tasks can be sent to edge nodes that are best suited for them — maybe one node has more memory, another is faster at image processing. Profiling helps match the task to the right resource.
- **Real-Time Adaptation**: Conditions change — a device that was free a minute ago might now be overloaded. Profiling helps the system stay agile and switch decisions on the fly when needed.
- **Enhanced Compatibility**: Profiling data can easily feed into existing systems, whether they're optimization algorithms or machine learning models, making them more accurate and responsive.

In short, profiling brings visibility and foresight into the offloading process. Rather than relying on assumptions or one-size-fits-all strategies, it enables decisions that are grounded in real-world data — leading to systems that are more efficient, adaptive, and capable of handling the complexities of modern edge AI.

4. Comparative Analysis of Profiling-Based Approaches vs Traditional Approaches

This section compares profiling-augmented offloading with traditional methods across key dimensions.

4.1 Comparison Table

Approach	Decision Mechanism	Heterogeneity Handling	Adaptivity	Resource Prediction	Performance Gain
Optimization (You *et al.*, 2018)	Analytical optimization	Low	Static	Implicit	Moderate
Heuristic (Bi *et al.*, PGL, 2022)	Metaheuristic PGL	Moderate	Offline	Estimated via simulation	High (energy savings)
ML-based (Abbas *et al.*, 2021)	DNN classification	Low	Static	Learned implicitly	High (latency/energy)
Modeling w/ Uncertainty (Ji, 2022)	EVT + DAG optimization	Low	Low	Statistical bounds	Robust under uncertainty

Approach	Decision Mechanism	Heterogeneity Handling	Adaptivity	Resource Prediction	Performance Gain
Profiling-Based (Proposed)	Profiling + predictive model	High	High	Explicit profile-based	Expected high across metrics

5. Discussion

5.1 Benefits of Profiling-Based Offloading

- **Fine-grained Optimization**: By knowing exact model runtime characteristics, offloading decisions can be more tailored and efficient.
- **Scalability Across Heterogeneity**: Profiling supports diverse hardware, models, and datasets, enabling deployment across varied edge environments.
- **Dynamic Adaptation**: As profiling reveals runtime changes (e.g., thermal throttling, workload fluctuations), the system can adapt in near real-time.
- **Compatibility with Existing Frameworks**: Profiling can enhance optimization, heuristic, or ML-based approaches by supplying accurate input data.

5.2 Challenges

- **Profiling Overhead**: Collecting profiling data across many combinations can be time-consuming and computationally expensive.
- **Generalization**: Profiles may not generalize across unseen hardware or model updates — necessitating continual updates.
- **Storage and Modelling Complexity**: Managing and querying high-dimensional profiling data requires careful design.
- **Integration Complexity**: Systems must be designed to incorporate profiling into decision-making pipelines seamlessly.

5.3 Future Directions

- **Incremental Profiling**: Use online monitoring to refine models progressively.
- **Federated Profiling**: Share anonymized profiles across devices to accelerate learning across deployments.
- **Hybrid Decision Models**: Combine profiling with RL or other ML approaches for dynamic decision-making.
- **Real-world Validation**: Implement prototypes across diverse hardware platforms in real edge scenarios (e.g. 6G edge networks).

6. Conclusion

As edge AI grows quickly, it is becoming more important to be smart about where and how we run AI models — especially in this device-capable world where devices will all have different capabilities. This paper showed that profiling AI models can serve as the bridge needed to achieve sustainable computation offloading in non-uniform environments.

Profiling AI models can provide insight from the perspective of the model and its corresponding tasks — profiling a model includes assessing any limitations with its memory, runtime, and knowledge on different devices. Ultimately profiling allows for better evaluation and choices, which shift the focus away from guesswork and enables proper primacy and timing for the right tasks on the right resources.

From our review and analysis, we discovered that profiling-based strategies demonstrated great improvements over traditional offloading paradigms. Profiling based strategies are inherently more flexible, more capable of predicting resource requirements, and provide an opportunity to avoid and accommodate uncertainty typical of edge AI systems.

That said, there are disadvantages, like the time dedicated to profiling and keeping up with profiles; however, when considering the profiling opportunities and the advantages, profiling provided a definite direction for leveraging the morphology of computation offloading on edge devices in future research and real-world opportunities. With the rapid evolution of edge computing, profiling is on the cusp of transitioning from optional to necessary for making sustainable computation offloading decision.

References

1. You, C., Zeng, Y., Zhang, R., & Huang, K. (2018). *Asynchronous Mobile-Edge Computation Offloading: Energy-Efficient Resource Management*. arXiv preprint. (ResearchGate, Moonlight, en.wikipedia.org, New Jersey Institute of Technology, ScienceDirect, Nature, MDPI, arXiv, en.wikipedia.org, arXiv)

2. Cao, X., Wang, F., Xu, J., Zhang, R., & Cui, S. (2018). *Joint Computation and Communication Cooperation for Energy-Efficient Mobile Edge Computing*. arXiv preprint. (arXiv)

3. Wan, Z., Dong, X., & Deng, C. (2022). *Optimizing Computation Offloading under Heterogeneous Delay Requirements for Wireless Powered Mobile Edge Computing*. *Wireless Networks*, 29, 1577–1607. https://doi.org/10.1007/s11276-022-03075-w (SpringerLink)

4. Bi, J., Yuan, H., Zhang, K., & Zhou, M. C. (2022). *Energy-Minimized Partial Computation Offloading for Delay-Sensitive Applications in Heterogeneous Edge Networks*. *IEEE Transactions on Emerging Topics in Computing*, 10(4), 1941–1954. https://doi.org/10.1109/TETC.2021.3137980 (New Jersey Institute of Technology)

5. Abbas, Z. H., et al. (2021). *Computational Offloading in Mobile Edge with Deep Learning-Based Technique (CEDOT)*. *Sensors*, 21(10), 3523. https://doi.org/10.3390/s21103523 (MDPI)

6. Ji, T., Luo, C., Yu, L., Wang, Q., Chen, S., Thapa, A., & Li, P. (2022). *Energy-Efficient Computation Offloading in MobileEdge Computing Systems with Uncertainties*. *arXiv preprint*. (arXiv)

7. Survey on Computation Offloading Modeling for Edge Computing (2020). *Research Overview*. (ResearchGate)