RESEARCH-ARTICLE

Reactive and Proactive Migration for Edge AI Training on Energy-Harvesting, Intermittently Powered Devices

FABIAN HENGSTBERGER, University of Vienna, Vienna, Vienna, Austria

ATAKAN ARAL, University of Vienna, Vienna, Vienna, Austria

Open Access Support provided by:

University of Vienna

Published: 01 December 2025

Citation in BibTeX format

UCC '25: 2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing
December 1 - 4, 2025
France, France

Conference Sponsors:
SIGARCH

https://doi.org/10.1145/3773274.3774693
ISBN: 9798400722851

Reactive and Proactive Migration for Edge AI Training on Energy-Harvesting, Intermittently Powered Devices

Fabian Hengstberger
University of Vienna
Vienna, Austria
a01546809@unet.univie.ac.at

Atakan Aral
University of Vienna
Vienna, Austria
atakan.aral@univie.ac.at

Abstract

In energy-constrained environments where edge devices operate on intermittent renewable sources, maintaining continuous AI model training poses significant challenges. This work evaluates two migration strategies, reactive and proactive, for preserving computational progress during power interruptions. The reactive strategy relies on centralized heartbeat monitoring to trigger checkpoint transfers, while the proactive strategy enables devices to offload state or data based on local energy thresholds. To improve reliability and efficiency, load balancing is applied to distribute computation across available devices, and battery buffering is integrated to stabilize power supply.

Extensive simulation results based on real-world data demonstrate that load balancing increases the migration success rate to 96% in the reactive setting and reduces migration frequency by 39-45% in the proactive scenario. Battery integration yields the most substantial improvement, reducing total proactive migrations by approximately 96% and limiting reactive migrations to just two. These findings highlight the effectiveness of adaptive migration and power-aware strategies in enabling robust, uninterrupted Edge AI training under intermittent energy availability.

CCS Concepts

• **Computer systems organization** → **Fault-tolerant network topologies**; • **Computing methodologies** → **Distributed computing methodologies**; **Cooperation and coordination**; **Simulation evaluation**; • **Hardware** → **Power and energy**.

Keywords

Edge AI, Edge Computing, Intermittent Computing, Migration Strategy, Energy Harvesting Devices, Availability, Resilience

ACM Reference Format:
Fabian Hengstberger and Atakan Aral. 2025. Reactive and Proactive Migration for Edge AI Training on Energy-Harvesting, Intermittently Powered Devices. In *2025 IEEE/ACM 18th International Conference on Utility and Cloud Computing (UCC '25), December 01–04, 2025, Nantes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3773274.3774693

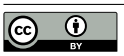
1 Introduction

In remote and isolated regions without access to a power grid, small edge devices play a crucial role in monitoring and data collection [9]. These devices must operate on harvested energy from sources like solar or wind [10], [28]. However, unpredictable weather can disrupt energy availability, causing devices to experience frequent power outages. Beyond data collection, these devices face the added challenge of training lightweight AI models on-site to process and analyse the gathered data and to perform predictions on trained AI models [31]. Furthermore, offloading to a central facility needs to be replaced by on-device training, as the network connectivity is unreliable and the bandwidth is limited in rural areas [4]. Moreover, the limited and unreliable energy supply creates another major challenge for both model training and data transmission between devices [29]. The problem of maintaining AI model training continuity during power interruptions requires novel approaches to computation state management [25]. Traditional checkpoint-based methods are insufficient when power failures occur frequently and unpredictably. Moreover, the distributed nature of edge deployments introduces additional complexity in coordinating model training across multiple devices with varying energy availability patterns. When an edge device faces imminent power failure, the computational state of ongoing AI model training must be preserved to avoid losing progress. This necessitates migration strategies that can transfer computation states, trained model parameters, and accumulated training data to other devices or systems that have sufficient energy reserves [30]. The challenge lies in developing efficient migration mechanisms that minimize energy consumption during the transfer process while ensuring seamless continuation of AI model training. Moreover, migration strategies in intermittent computing environments must predict power failures, identify suitable target devices, minimize migration overhead to conserve energy, and ensure migrated AI models maintain accuracy and learning effectiveness.

This work addresses the problem of enabling continuous AI model training in energy-harvested edge devices through effective migration strategies, and the following are the main contributions of this work.

(1) Comparative study of migration strategies for Edge AI under intermittent power.
(2) Optimization mechanisms that improve migration behavior.
(3) An energy-harvesting-driven simulation framework for Edge AI training.

The rest of this paper is organized as follows. In Section 2, we review the literature related to the topic, followed by the declaration of the migration strategies proposed in Section 3. The Experimental Setup is described in Section 4, the Evaluation of the results is done in Section 5, and Section 6 concludes the work.



UCC '25, Nantes, France

ACM ISBN 979-8-4007-2285-1/25/12
https://doi.org/10.1145/3773274.3774693

2 Related Work

To address migration challenges in intermittent Edge AI environments, recent research has explored strategies for optimizing both AI model training and data transmission. Developing intermittent devices brings several hurdles, for instance erratic runtimes and volatile memory states caused by unpredictable power outages [16]. In intermittent computing settings, edge devices are predominantly powered via energy harvesting [14]. A diverse array of harvesting techniques, e.g. solar, acoustic, radio-frequency, wind, thermal, mechanical and chemical, has been explored to sustain operation under erratic power input [3]. Notably, pairing solar and wind harvesters with supercapacitors has emerged as a particularly promising solution for delivering stable power in rural edge deployments [24] [11]. However, energy harvesting introduces several challenges, such as hardware profiling, energy-aware testing of edge devices, and adapting devices for deployment in uncontrolled environments [8]. Lee and Chang [15] address one aspect of this by boosting harvested energy by 8% without relying on batteries or other converters. Several studies focused on power-management schemes to maximize throughput. Akunov et al. [2] proposed an power optimized computation strategy called TETRA (Three Threshold Tracker), which enables an edge device to remain in a volatile state of computation, even when the energy harvester fails to supply sufficient energy for computation on the AI model. In a related effort to maximize the computational throughput of edge devices, Luo et al. [17] introduced an offline optimal computing power management strategy that leverages a battery in conjunction with a solar energy harvester. Their findings indicate that determining the optimal battery capacity and the appropriate size of the energy harvester for effective charging represents an optimization problem. Beyond traditional harvesters, research on Neuromorphic Edge AI shows that event-driven neuromorphic hardware can operate under minimal energy conditions [4]. This enables computing lightweight AI tasks even with unpredictable power availability. However, maintaining the relevance of locally trained models in diverse microclimates presents a challenge when data contexts vary across regions [13]. Not just energy harvesting, research also has explored offloading strategies to maximize throughput in intermittent Edge AI environments.

Research on fault-tolerant edge offloading frameworks provide strategies, using Support Vector Regression (SVR) to predict offloading decisions and try to optimize resource use [33]. Aral and Brandic [5] introduced a DBN-based modeling framework that uncovers correlated failures from common causes and learns dependency patterns in edge-computing systems. Building on this work, Zilic et al. [32] then developed EFPO, a failure-predictive, energy-efficient offloading framework. Recent work has also distilled key criteria for crafting efficient migration frameworks for general applications [1]. Offloading delivers the greatest benefit when workloads demand substantial computation but involve relatively small data sizes, since constrained compute resources reduce the odds of successful transfers, and moving large datasets in intermittent settings can drain precious energy [18]. Moreover, partial offloading of data yields further energy savings compared to migrating entire datasets. An alternative is checkpointing, capturing an application's state via timer-based triggers, load/store-error detection, or voltage thresholds, and migrating that snapshot rather than the full workload [26]. When training and migrating AI models across resource-constrained edge devices, reducing model complexity is essential [22]. Employing lightweight architectures not only promotes offloading among heterogeneous platforms but also ensures hardware-agnostic operation [27]. Finally, Aslanpour et al. [6] introduced Hedgi, a heterogeneous, serverless edge-computing architecture tailored for AI-driven IoT deployments. By leveraging a weight-tuned, multi-objective load balancing scheme, Hedgi tackles intermittent challenges by harmonizing energy awareness with cost efficiency. However, balancing these offloading operations with limited power resources remains a challenge, as data transmission and task re-computation can significantly impact energy consumption.

3 Proposed Migration Strategies

To address the challenge of intermittent computing, two migration strategies were implemented: a reactive approach and a proactive approach. Each strategy is designed to mitigate the impact of power fluctuations and ensure reliable data migration between devices.

3.1 Reactive Approach

Algorithm 1, the reactive approach, defines a migration strategy in which migration only takes place when the edge devices lack power and in which a centralized server continuously monitors the status of the edge devices via a heartbeat request at each time step. If a device is online and has sufficient power, it responds with an acknowledgment and the current checkpoint of the data or AI model, depending on the offloading strategy. If an edge device runs out of power and fails to respond to the heartbeat request, the server recognizes that the device is offline. In response, it tries to assign the last available checkpoint to a partner device with sufficient resources. When model offloading is deployed, the server selects a running partner device based on resource constraints, such as the number of services already assigned to it. In contrast, with data offloading, data is assigned to any available partner device, regardless of its current data load. If no device meets the required criteria, the server postpones the migration and retries to re-assign the checkpoint in the next time step.

The granularity of this strategy enhances via the controlling of the currently available resources of the remaining, online edge devices, like restrictions for the maximum amount of services per device and/or the number of partner devices each edge device gets assigned. The comparison and results of the restrictions are highlighted in Section 5.1.

3.2 Proactive Approach

The proactive approach in Algorithm 2 describes a migration strategy, in which edge devices analyse their current power supply and if necessary initiate a migration of their model or data before running out of power. Therefore, the edge devices check their power supply at every time step of the simulation. If the power is below a certain threshold, the device assumes a dropping power supply for the next time steps and initiates a migration of AI model or data. In comparison to the reactive approach, the proactive approach

Algorithm 1 Reactive Migration Strategy

```
//Edge Device (ED) Logic
for each ED do
    update current power
    update current state
    if device running then
        measure data via sensors
    end if
    for each service of the ED do
        if not offloading then
            train AI model
        else
            stop AI model
        end if
    end for
end for

//Server Logic
for each ED do
    get heartbeat
    if heartbeat == false and model/data ≠ empty then
        if ED has available partner ED then
            if partner ED meets max_services restriction then
                assign checkpoint to partner ED
            end if
        end if
    end if
end for
```

Algorithm 2 Proactive Migration Strategy

```
//Edge Device (ED) Logic
for each ED do
    update current power
    update current state
    if device running then
        measure data via sensors
    end if
    for each service of the ED do
        if not offloading then
            train AI model
        else
            stop AI model
        end if
        if current power < threshold and no migration then
            initiate migration to server
        end if
    end for
end for

//Server Logic
if Server has service(s) then
    if any ED online and ED has no service then
        if ED has no migration then
            initiate migration to ED
        end if
    end if
end if
```

migrates the model/data as a whole, rather than just a checkpoint. Therefore, the payload of the migration is much higher and the migration needs a couple of time steps to complete entirely. After a successful finished migration, the server holds the model/data and tries to find another device, which meets certain running criteria, to assign the model/data to it. If the migration from an edge device to the server fails, then the model/data resume residing on the device. If the device gets enough power in the future then it will continue running the model and collecting data.

Now, the question arises if an edge device is eligible initiating the migration to the server or not, which is covered as a comparison of 'True-Offloading vs. False-Offloading' in the results in Section 5.2.

3.3 Optimization Strategies

To further improve the proposed migration strategies, the following optimization strategies are implemented.

3.3.1 Load Balancing. A Load Balancing (LB) service on the centralized server activates when AI model offloading is used. The LB distributes migrated checkpoints or models across edge devices with available computing capacity. The device reserve is calculated as the difference between CPU cores and currently running services, which one core reserved for the operating system, communication and measurements. If the reserve is greater than 0, the LB assigns additional models until reaching full utilization, meaning reserve is 0. This strategy prevents reactive devices from being overloaded with AI models while ensuring proactive devices utilize available processing power. Moreover, LB aims to increase the model program counters by reducing total migrations and accelerating overall training.

3.3.2 Battery. To reduce the total migrations and gain partial resilience against intermittent power availability, a battery is integrated into each edge device. The battery is placed between the energy harvester and the edge device, such that the energy harvester charges the battery, while the battery continuously powers the edge device through a stable discharge. The battery model incorporates real-world constraints including charging efficiency, initial state of charge, and maximum depth of discharge to prevent long-term battery damage. The battery size is configured by its capacity, $[Q] = Ah$, and voltage, $[U] = V$. Energy consumption is simulated using a constant power parameter representing continuous device demand during operation.

4 Experimental Setup

4.1 Overview

To realize an accurate simulation[1], the simulation framework *EdgeSimPy* [23] is chosen. *EdgeSimPy* is a Python-based modelling and simulation framework that focusses on edge computing resource management policies, provides a wide range of abstracted edge

[1] https://github.com/hengsti/EdgeAI_MigrationStrategies

computing components, and collects the state of the simulation at every time step.

4.2 Topology and Devices

The base of the simulation is a topology to model the environment. Therefore, 10 Base Stations are defined, on which every Base Station is residing one device. Next, a server is instantiated on a Base Station to play a central role in the simulation, and it resides at a location, with a constant power supply. To complete the physical components of the simulation, 9 edge devices are defined. The edge devices are Raspberry Pi-5 (RPI5), a cheap single-board computer with enough computing power to run lightweight AI models on it [12]. Moreover, the RPI5s collect measurements, here temperature, over an external sensor, which builds up the data source to train and run the AI models for local weather prediction models [27]. In addition, each edge device operates two energy harvesting modules, 1 solar panel and 1 wind turbine. All devices, server and edge devices, communicate over an always available LoRaWAN (Long Range Wide Area Network), which is a low-power networking protocol and fits well in an intermittent computing setting [19], [20].

4.3 Energy Model

The energy supply for edge devices and real-world behavior of energy harvesting devices was simulated using the *Photovoltaic (PV) Solar Panel Energy Generation Data (UK Power Networks)* dataset [7], and consists of 69,480 rows of weather data collected in London from November 2013 to November 2014. It contains 38 columns of various measurements, recorded every 30 minutes across five locations in the city. For the simulation, the columns "WindSpeed" and "SolarEnergy" are used as the data source for energy harvesting. These renewable energy resources come with the advantage that they are minimal inversive for rural environments and have low operational costs in a real world scenario [21]. The edge devices in the simulation are equipped with a 0.5 m^2 solar panel and a small wind turbine. The wind speed (measured in m/s) and solar energy (measured in Langley) needs to be converted to Watts. Two conversion formulas are applied to transform the energy data into the appropriate units. Converting Solar Energy, $[E_{\text{solar}}] = Ly$, to Solar Power, $[P_{\text{solar}}] = W$, for a 0.5 m^2 large solar panel:

$$P_{\text{solar}} = E_{\text{solar}} \cdot 11.622 \cdot \frac{1}{30} \cdot 60 \cdot 0.5 = E_{\text{solar}} \cdot 11.622 \tag{1}$$

Converting Wind Speed, $[v] = \frac{m}{s}$, to Wind Power, $[P_{\text{wind}}] = W$, for a small wind turbine with a swept area of 0.5 m^2:

$$P_{\text{wind}} = 0.5 \cdot \rho \cdot A \cdot v^3 \cdot C_p \cdot \eta \tag{2}$$

where:

$\rho = 1.225, \text{kg/m}^3$ (air density), $A = 0.5\,\text{m}^2$

$C_p = 0.35$ (Power Coefficient), $\eta = 0.9$ (Generator Efficiency)

The simulation transforms the sequential weather data into a simulated weather model and assigns each device its own harvesting instance.

4.4 Measurements and AI models

Devices collect data from a simulated temperature sensor during operation when no migration is active. Each AI model maintains a state and program counter at every time step, transitioning to the prediction phase after completing training.

4.5 Offloading Strategies

To face the restrictions of intermittent computing in an edge computing setting, different offloading possibilities are created. *Data offloading* describes the option to migrate the collected measurements to another nearby edge device. *Model offloading* describes the migration of the AI model from one device to another device.

4.6 Migration Strategies

In addition to the proposed migration strategies, an oracle approach is implemented as a theoretical benchmark, generating and assigning checkpoints like the reactive strategy in Algorithm 1, but without limitations on concurrent services or partner devices. It guarantees success as long as one device remains operational, providing an idealized comparison for evaluating the reactive and proactive strategies.

5 Evaluation and Results

The following results were obtained from 69 simulation runs, each spanning 1000 timesteps and using the production topology. To start of, Figure 1 illustrates the Migration Success Rate (MSR, equation 3) for data and model offloading. For data offloading, the oracle strategy achieves a flawless MSR of 100%. In contrast, the proactive strategy, which relies on local power threshold estimation, reaches only 53.6% at 25 W and drops further to 48.4% at 15 W. The reactive strategy with four or more partner devices also achieves a perfect MSR, showing that such a configuration can match oracle-level reliability under favorable conditions.

$$\text{MSR} = \frac{M_{\text{succ}}}{M_{\text{tot}}} \tag{3}$$

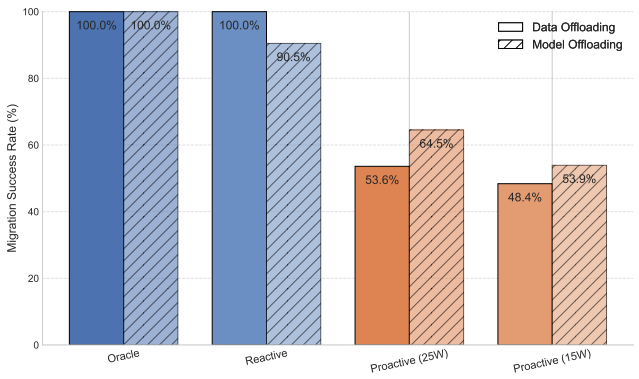


Figure 1: Migration Success Rate Comparison for Data and Model Offloading across Strategies

A similar trend can be observed in the results for model offloading in Figure 1. Here, the oracle approach once again reaches 100% MSR. The reactive strategy, when configured with four partner devices and a five-model service limit, achieves 90.5%. This configuration shows strong performance, though it still falls short of the

oracle level. The proactive strategy yields lower success rates, with 64.5% at 25 W and 53.9% at 15 W. These results highlight the inherent limitations of decentralized, threshold-based decision-making when compared to the coordinated control offered by the oracle strategy. In both data and model offloading, the oracle strategy provides a valuable upper bound for evaluating the migration success of practical strategies. While the reactive strategy approaches this bound under specific conditions, the proactive approach exhibits a clear drop in reliability, especially under lower power thresholds. This confirms that proactive migrations require additional optimization to close the MSR gap.

5.1 Reactive Approach

5.1.1 Data vs Model Offloading: optimal number of partner devices. The evaluation of the reactive strategy reveals a strong dependency between the number of assigned partner devices and the migration success rate. As shown in Figure 2a, configurations with fewer than 4 partners yield noticeably lower success rates. For example, the simulation with only two partners reaches a success rate of just 71.77% for the data, and 51.46% for the model offloading. In contrast, from 4 partner devices onward, the system achieves a 100% success rate for data offloading. Further increasing the partner devices to 5 and more, data and model offloading gain a 100% success rate. This behavior can be attributed to the volatile energy availability

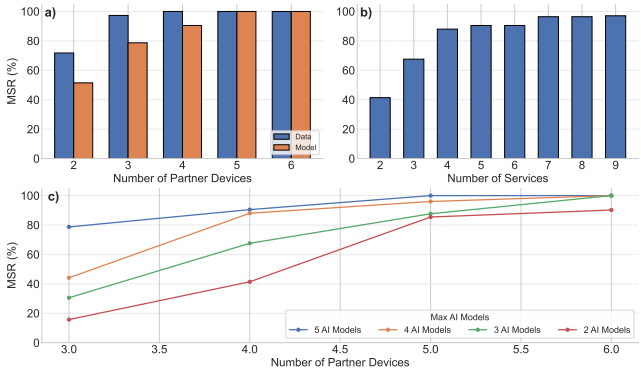


Figure 2: a) MSR for Data Offloading vs. Model Offloading (5 AI models), b) MSR for varying AI models and 4 Partner Devices, c) MSR for varying AI models and Partner Devices

across edge devices. If all assigned partners are temporarily offline, the server cannot complete the migration and must retry in the next simulation step. With an increasing number of partner devices, the chance that at least one of them is online rises significantly. A critical threshold is observed at 4 partners for data, and 5 partner for model offloading, where redundancy becomes sufficient to consistently guarantee successful migrations.

5.1.2 AI Model Offloading: Max. Services per Device vs Number of Partner Devices. Figure 2c show that the success rate of model offloading depends heavily on both the number of partner devices and the number of concurrent AI model services. For example, doubling the partner devices from 3 to 6 for a configuration with two concurrent services improves the migration success rate by 74.49%. With five services and six partners, the system reaches 100% success. Nevertheless, assigning more than three services to an edge device leads to increased context switching and degraded performance. As a result, the MSR from allowing more services per device plateaus quickly. The performance limitation is illustrated in Figure 2b, which presents success rates for different service limits while keeping the number of partner devices fixed at 4. Although increasing the service limit consistently improves the success rate, failures remain possible if too many partner devices are unavailable at the same time, and excessive context switches reduce the overall performance in training of the AI models.

5.1.3 Load Balancing. The results with LB enabled, confirm that load balancing leads to higher and more stable migration success rates. Especially for configurations with 4 to 6 concurrent models per device, MSRs approach 96%. Compared to non-balanced configurations, load balancing shifts the point of optimal CPU usage to lower service limits. As a result, edge devices can train models more efficiently while reducing overhead caused by context switching and failed migrations.

5.2 Proactive Approach

5.2.1 True-Offloading vs False-Offloading. To evaluate the behavior of the proactive migration strategy, a combined analysis of true and false migrations across both data and model offloading scenarios was conducted. Figure 3 presents the results for two threshold configurations, 25 W and 15 W. At 25 W, data offloading leads to a relatively balanced migration pattern across devices, with slightly more true migrations than false ones in most cases. However, with the 15 W threshold, the number of false migrations increases noticeably across the devices, highlighting a tendency toward premature offloading when devices still maintain operational potential. A similar pattern is observed for model offloading: while the 25 W configuration maintains a higher ratio of true migrations, the 15 W configuration again sees a substantial rise in unnecessary migrations. These results confirm that selecting an appropriate power threshold is critical in proactive environments. A threshold that is too conservative leads to excessive and unjustified offloading events, increasing communication overhead and draining energy reserves without tangible progress in model training or data processing. The analysis also underlines that proactive migration is inherently more sensitive to power fluctuations, if devices hover around the threshold boundary frequently. Consequently, static thresholds may not be sufficient for environments with highly dynamic energy conditions. Adaptive mechanisms or historical power trend analysis could help improve migration timing and reduce the proportion of false migrations.

5.2.2 Load Balancing. With LB enabled, the MSR rises to 66.7% at 25 W and 55.3% at 15 W, which improves slightly the overall migration behavior and reduces the total number of migration events, as illustrated in Figure 4. For the 25 W threshold, total migrations drop from 231 to 140 (39% reduction), while at 15 W, migrations reduce from 241 to 134 (45% reduction). These reductions allow devices to spend less time on migrations and increase AI model training time. Interestingly, the simulation using the 15 W threshold now results

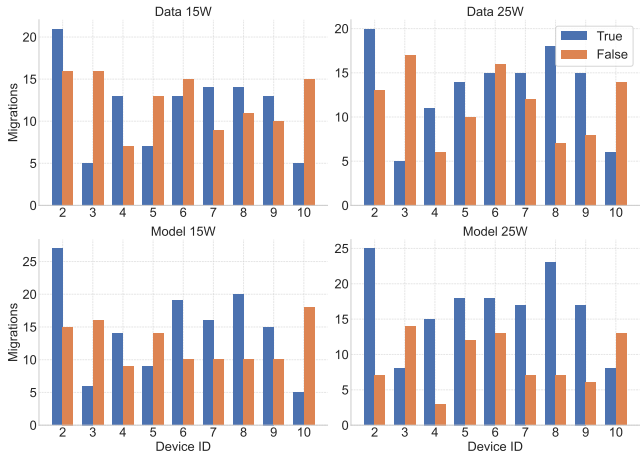


Figure 3: True and False Migrations for Data and Model Offloading with 25 W and 15 W Thresholds

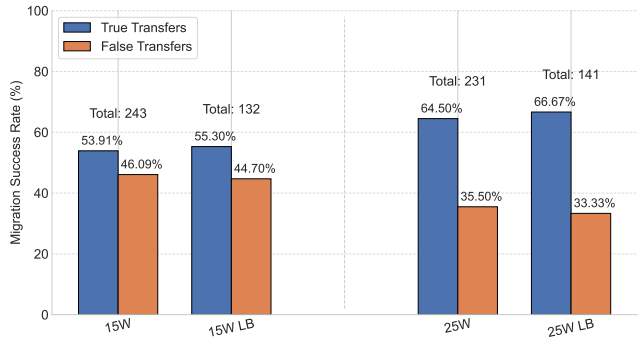


Figure 4: Comparison of Model Offloading with and without Load Balancing with 15W and 25W Thresholds

in fewer total migrations than the 25 W simulation. However, the MSR for the 25 W threshold still outperforms that of the 15 W setup.

5.3 Battery

To evaluate the impact of a battery in combination with an energy harvester and an edge device that continuously draws power, two different battery sizes are considered. To ensure comparability between battery configurations, the parameters *efficiency*, *initial_charge*, and *depth_of_discharge* are fixed at values of 0.8, 0.5, and 0.25, respectively.

5.3.1 13.77 Wh vs. 102 Wh Battery. The small battery with 13.77 Wh capacity (2.7 Ah at 5.1 V) can theoretically power a Raspberry Pi 5 that draws 25 W for about 34.0 minutes. Figure 5 illustrates the Battery State Of Charge (BSOC) for every edge device over 1000 simulation timesteps using the 13.77 Wh battery. The wave like pattern reflects periodic solar and wind availability. In the upper plot, the devices 3, 7, and 9 reach full charge over time, and in the lower plot, the devices 2 and 4 experience complete power loss, where device 2 briefly restarts but cannot remain online. The larger 102 Wh battery extends theoretical runtime to 4.08 hours

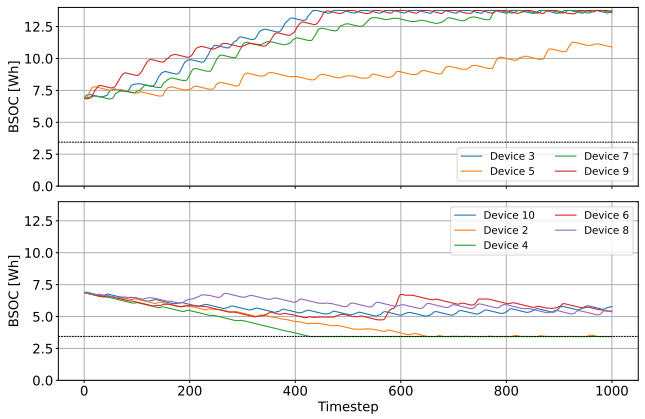


Figure 5: BSOCs over simulation timesteps with 13.77 Wh Batteries

at 25W. Although harvesters cannot fully charge these batteries, all devices remain online throughout the simulation Therefore, the battery's maximum capacity impacts the device availability and enables operation during adverse weather conditions when energy harvesters cannot collect and transform sufficient energy.

5.3.2 Proactive Strategy. Now, migrations starts at 40% BSOC, which leaves about 15% of capacity to finish the migration. For data and model offloading, the number of migrations drops about 95.67% from 231 to 10 with 0 false migrations, and the load balanced model offloading reduces the migrations about additional 1.73% to 97.40% with 0 false migrations, using the 13.77 Wh battery.

5.3.3 Reactive Strategy. The simulations run with the "4 partner per device"- and "max. concurrent AI models per device"-restriction, from 2 to 9 with LB enabled and disabled. Across all configurations and offloading strategies, only 2 successful migrations occur in total, indicating that suitable partner devices were available. This confirms that the total migrations can be significantly lowered, but load balancing does not provide additional benefits. However, the program counter of the AI models are increased by the fault-tolerance of the edge devices.

6 Conclusion

Both, the reactive and proactive approaches aim to enhance data and AI model offloading in dynamic, resource-constrained edge environments. The results indicate common themes in success rates being influenced by the number of partner devices and power availability. However, key differences arise in how migrations are initiated and the types of errors encountered. For the reactive strategy, balancing the optimal number of partner devices is crucial. A low number results in offloading delays, while too many partners can lead to cross-region data migrations, causing inaccurate training of AI models. The proactive strategy faces challenges related to minimizing unnecessary migrations triggered by frequent power supply fluctuations, reducing the overall system efficiency. Moreover, introducing a Load Balancer in the reactive approach achieves a 96% Migration Success Rate when reducing the maximum concurrent AI models per device from 6 to 4. For the proactive approach,

load balancing reduces overall needed migrations by 39% to 45%, enabling a significant increase in the AI models program counter. Extending edge devices with a small battery, powered by the energy harvester, reduces total migrations needed by approximately 96% in proactive settings for both, AI model and data offloading, while reactive strategy migrations reduce to 2. However, larger batteries extend device lifecycle but introduce optimization challenges between the energy harvester size and efficient battery charging rates.

Future work should focus on developing a hybrid approach combining reactive device monitoring with proactive resource allocation via dynamically adjusting power thresholds and load balancing. Additionally, real hardware testing is needed to determine optimal battery sizing for handling weather-induced power interruptions.

Acknowledgments

This research was funded in part by CHIST-ERA-22-SPiDDS-07 (TROCI Project) and Austrian Science Fund (FWF) [10.55776/I6647].

References

[1] Abednego Acheampong, Yiwen Zhang, Xiaolong Xu, and Daniel Appiah Kumah. 2022. A Review of the Current Task Offloading Algorithms, Strategies and Approach in Edge Computing Systems. *CMES - Computer Modeling in Engineering and Sciences* 134, 1 (2022), 35–88. doi:10.32604/cmes.2022.021394

[2] Khakim Akhunov, Eren Yildiz, and Kasim Sinan Yildirim. 2023. Enabling Efficient Intermittent Computing on Brand New Microcontrollers via Tracking Programmable Voltage Thresholds. In *Proceedings of the 11th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems* (Istanbul, Turkiye) *(ENSsys '23)*. Association for Computing Machinery, New York, NY, USA, 16–22. doi:10.1145/3628353.3628547

[3] Ahsan Ali, Hamna Shaukat, Hassan Elahi, Shaista Taimur, Muhammad Qasim Manan, Wael A. Altabey, Sallam A. Kouritem, and Mohammad Noori. 2025. Advancements in energy harvesting techniques for sustainable IoT devices. *Results in Engineering* 26 (2025), 104820. doi:10.1016/j.rineng.2025.104820

[4] Atakan Aral. 2024. The promise of neuromorphic edge AI for rural environmental monitoring. *Environmental Data Science* 3 (2024), e34. doi:10.1017/eds.2024.36

[5] Atakan Aral and Ivona Brandic. 2018. Dependency Mining for Service Resilience at the Edge. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. 228–242. doi:10.1109/SEC.2018.00024

[6] Mohammad Sadegh Aslanpour, Adel N. Toosi, Muhammad Aamir Cheema, Mohan Baruwal Chhetri, and Mohsen Amini Salehi. 2024. Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach. *Future Generation Computer Systems* 154 (2024), 266–280. doi:10.1016/j.future.2024.01.020

[7] Greater London Authority. [n. d.]. *Photovoltaic (PV) Solar Panel Energy Generation Data.* https://data.london.gov.uk/dataset/photovoltaic--pv--solar-panel-energy-generation-data

[8] Abu Bakar and Josiah Hester. 2018. Making sense of intermittent energy harvesting. In *Proceedings of the 6th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems* (Shenzhen, China) *(ENSsys '18)*. Association for Computing Machinery, New York, NY, USA, 32–37. doi:10.1145/3279755.3279762

[9] Alessio Catalfamo, Atakan Aral, Ivona Brandic, Ewa Deelman, and Massimo Villari. 2024. Machine Learning Workflows in the Computing Continuum for Environmental Monitoring. In *Computational Science – ICCS 2024: 24th International Conference, Malaga, Spain, July 2–4, 2024, Proceedings, Part V* (Malaga, Spain). Springer-Verlag, Berlin, Heidelberg, 368–382. doi:10.1007/978-3-031-63775-9_27

[10] Sravanthi Chalasani and James M. Conrad. 2008. A survey of energy harvesting sources for embedded systems. In *IEEE SoutheastCon 2008*. 442–447. doi:10.1109/SECON.2008.4494336

[11] Pooja Choudhary, Lava Bhargava, Virendra Singh, Manju Choudhary, and Ashok kumar Suhag. 2020. A survey – Energy harvesting sources and techniques for internet of things devices. *Materials Today: Proceedings* 30 (2020), 52–56. doi:10.1016/j.matpr.2020.04.115 International Conference on Advancement inNanoelectronicsand Communication Technologies(ICANCT-2020).

[12] Raspberry Pi Foundation. 2023. Raspberry Pi 5 Product Overview. https://www.raspberrypi.com/products/raspberry-pi-5/ Accessed: 2025-07-28.

[13] Haochen Hua, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. 2023. Edge Computing with Artificial Intelligence: A Machine Learning Perspective. *ACM Comput. Surv.* 55, 9, Article 184 (Jan. 2023), 35 pages. doi:10.1145/3555802

[14] Hrishikesh Jayakumar, Kangwoo Lee, Woo Suk Lee, Arnab Raha, Younghyun Kim, and Vijay Raghunathan. 2014. Powering the internet of things. In *Proceedings of the 2014 International Symposium on Low Power Electronics and Design* (La Jolla, California, USA) *(ISLPED '14)*. Association for Computing Machinery, New York, NY, USA, 375–380. doi:10.1145/2627369.2631644

[15] Hyung Gyu Lee and Naehyuck Chang. 2015. Powering the IoT: Storage-less and converter-less energy harvesting. In *The 20th Asia and South Pacific Design Automation Conference*. 124–129. doi:10.1109/ASPDAC.2015.7058992

[16] Brandon Lucia, Vignesh Balaji, Alexei Colin, Kiwan Maeng, and Emily Ruppel. 2017. Intermittent Computing: Challenges and Opportunities. In *2nd Summit on Advances in Programming Languages (SNAPL 2017) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 71)*, Benjamin S. Lerner, Rastislav Bodík, and Shriram Krishnamurthi (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 8:1–8:14. doi:10.4230/LIPIcs.SNAPL.2017.8

[17] Yu Luo, Lina Pu, and Chun-Hung Liu. 2025. Computing Power and Battery Charging Management for Solar Energy Powered Edge Computing. *IEEE Transactions on Mobile Computing* 24, 3 (March 2025), 1913–1927. doi:10.1109/TMC.2024.3489028

[18] Pavel Mach and Zdenek Becvar. 2017. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1628–1656. doi:10.1109/COMST.2017.2682318

[19] Habib Ur Rahman, Mudassar Ahmad, Haseeb Ahmad, and Muhammad Asif Habib. 2020. LoRaWAN: State of the Art, Challenges, Protocols and Research Issues. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*. 1–6. doi:10.1109/INMIC50486.2020.9318170

[20] P.P. Ray. 2018. A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences* 30, 3 (2018), 291–319. doi:10.1016/j.jksuci.2016.10.003

[21] Mahidur R. Sarker, Amna Riaz, M.S. Hossain Lipu, Mohamad Hanif Md Saad, Mohammad Nazir Ahmad, Rabiah Abdul Kadir, and José Luis Olazagoitia. 2024. Micro energy harvesting for IoT platform: Review analysis toward future research opportunities. *Heliyon* 10, 6 (2024), e27778. doi:10.1016/j.heliyon.2024.e27778

[22] Raghubir Singh and Sukhpal Singh Gill. 2023. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems* 3 (2023), 71–92. doi:10.1016/j.iotcps.2023.02.004

[23] Paulo S. Souza, Tiago Ferreto, and Rodrigo N. Calheiros. 2023. EdgeSimPy: Python-based modeling and simulation of edge computing resource management policies. *Future Generation Computer Systems* 148 (2023), 446–459. doi:10.1016/j.future.2023.06.013

[24] Sujesha Sudevalayam and Purushottam Kulkarni. 2011. Energy Harvesting Sensor Nodes: Survey and Implications. *IEEE Communications Surveys & Tutorials* 13, 3 (2011), 443–461. doi:10.1109/SURV.2011.060710.00094

[25] Milijana Surbatovich, Brandon Lucia, and Limin Jia. 2020. Towards a formal foundation of intermittent computing. *Proc. ACM Program. Lang.* 4, OOPSLA, Article 163 (Nov. 2020), 31 pages. doi:10.1145/3428231

[26] Sumanth Umesh and Sparsh Mittal. 2021. A survey of techniques for intermittent computing. *Journal of Systems Architecture* 112 (2021), 101859. doi:10.1016/j.sysarc.2020.101859

[27] Xubin Wang, Zhiqing Tang, Jianxiong Guo, Tianhui Meng, Chenhao Wang, Tian Wang, and Weijia Jia. 2025. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Comput. Surv.* 57, 9, Article 228 (April 2025), 39 pages. doi:10.1145/3724420

[28] Alexander J. Williams, Matheus F. Torquato, Ian M. Cameron, Ashraf A. Fahmy, and Johann Sienz. 2021. Survey of Energy Harvesting Technologies for Wireless Sensor Networks. *IEEE Access* 9 (2021), 77493–77510. doi:10.1109/ACCESS.2021.3083697

[29] Harrison Williams and Matthew Hicks. 2024. A Survey of Prototyping Platforms for Intermittent Computing Research. In *Proceedings of the 12th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems* (Hangzhou, China) *(ENSsys '24)*. Association for Computing Machinery, New York, NY, USA, 8–14. doi:10.1145/3698384.3699612

[30] Jie Xu and Shaolei Ren. 2016. Online Learning for Offloading and Autoscaling in Renewable-Powered Mobile Edge Computing. In *2016 IEEE Global Communications Conference (GLOBECOM)*. 1–6. doi:10.1109/GLOCOM.2016.7842069

[31] Roger Young, Sheila Fallon, and Paul Jacob. 2017. An Architecture for Intelligent Data Processing on IoT Edge Devices. In *2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim)*. 227–232. doi:10.1109/UKSim.2017.19

[32] Josip Zilic, Atakan Aral, and Ivona Brandic. 2019. EFPO: Energy Efficient and Failure Predictive Edge Offloading. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing* (Auckland, New Zealand) *(UCC'19)*. Association for Computing Machinery, New York, NY, USA, 165–175. doi:10.1145/3344341.3368818

[33] Josip Zilic, Vincenzo De Maio, Atakan Aral, and Ivona Brandic. 2022. Edge offloading for microservice architectures. In *Proceedings of the 5th International Workshop on Edge Systems, Analytics and Networking* (Rennes, France) *(EdgeSys '22)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3517206.3526266