2025 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C) | 979-8-3315-0215-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ACSOS-C66519.2025.00057

SunSift: Solar-Powered Intelligent Sensing through Informative Sample Selection

Shayan Gerami, Sepehr Tabrizchi, Omer Kurkutlu, Rebati Gaire, and Arman Roohi
Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL
E-mails: {sgera,stabr,okurku2,rrgaire,aroohi}@uic.edu

***Abstract*—We present `SunSift`, a lightweight framework enabling batteryless CNN inference on energy-harvesting IoT devices for distributed federated learning. SunSift employs a novel finite state machine with intelligent voltage-based transitions and selective MRAM checkpointing to minimize overhead while maximizing forward progress during intermittent execution. Our implementation achieves 71.51% accuracy in identifying informative samples from CIFAR-10, successfully distinguishing underrepresented categories from well-represented ones. The code and a video are made available at Source Code, and Video, respectively, for reproducibility and further research.**

I. Introduction and Background

The Internet of Things (IoT) continues to redefine edge computing, with trillions of devices expected to be deployed in domains such as smart cities, industrial monitoring, healthcare, and agriculture [1]. However, the traditional reliance on batteries imposes severe limitations in terms of sustainability, environmental cost, and long-term scalability. The global battery recycling market, valued at $19.4 billion in 2024, is projected to reach $69.4 billion by 2034, with a CAGR of 13.6% [2], provoking a shift toward sustainable, maintenance-free, and environmentally conscious alternatives. Energy harvesting systems, powered by solar, RF, or thermal energy, offer a compelling solution [3], [4], enabling devices to operate independently of wired power or battery replacements. Yet, such systems are fundamentally intermittent: power failures occur frequently and unpredictably, challenging conventional computing models and making complex, long-running tasks like deep neural network inference infeasible. Intermittent computing has emerged as a key paradigm to address this volatility by allowing devices to retain and resume computation across power failures [5], [6]. However, existing techniques such as static checkpointing and task-based models [7]–[9] fall short when applied to modern inference workloads. Static approaches suffer from excessive overhead due to frequent, full-state saves and recoveries, while task-based models impose significant programmer burden and often fail to provide fine-grained control needed by deeply pipelined operations like CNNs. Reactive methods, though more adaptive, still struggle to efficiently support hardware accelerators and reduce memory pressure in CNN inference [10]–[12]. In this demo, we introduce **`SunSift`**, a lightweight, energy-aware, and robust framework for intermittent CNN inference on battery-less intelligent sensing nodes. **`SunSift`** combines the adaptability of reactive intermittent execution with a novel state machine that minimizes checkpointing overhead and maximizes forward progress.

II. System Architecture

In this implementation, we use two Arduino Nano 33 BLE boards: one acts as the power management unit (PMU), simulating an energy-harvesting environment, while the second board executes the core system logic. The state machine on the second microcontroller includes seven states: *light sleep* (`LS`), *deep sleep* (`DS`), *store* (`Str`), *sense* (`Se`), *compute* (`Cp`), *Load* (`Ld`), and *transmit* (`Tr`). The PMU monitors voltage levels and generates interrupts to control the operation of the microcontroller (MCU). Three voltage thresholds are defined: the Operation Threshold, above which standard tasks such as sensing and computation proceed; the Safe Threshold, which triggers light sleep mode during intermediate stages of layer-by-layer computation if the voltage drops below this level; and the Store Threshold, below which the system transitions immediately to a store state before entering deep sleep mode to preserve critical data. The PMU uses three dedicated pins to signal threshold violations, each corresponding to one of the thresholds. When the voltage drops below a given threshold, the associated pin is set to HIGH. The MCU checks the status of these pins at various points in the execution flow, such as after each state transition and between computation layers. Upon detecting a HIGH signal, the MCU takes the appropriate action, such as entering sleep mode. When the voltage recovers and the pin returns to LOW, the MCU resumes normal operation.

In our demo configuration, we integrate TensorFlow Lite for Microcontrollers (TFLM) to perform a 5-layer CNN inference to determine the informative and non-informative images [13]. However, due to TFLM's internal abstraction, there is no direct access to layer-level kernel or activation indices during runtime. As a result, our system stops after each CNN layer, performing selective data storage and recovery only at layer boundaries rather than suboperation-level granularity. Although this limits intra-layer atomicity, it enables reliable layer-wise forward progress and reduces the risk of redundant computation after power loss.

The state machine of **`SunSift`** is shown in Fig. 1(a). Each state is associated with a specific color code, as detailed in Table I. **`SunSift`** begins operation in the `Se` state. After sensing a new image, the system transitions back to the `LS` state, where it waits for the PMU to check the energy status every 5 seconds. When sufficient energy is available, the system wakes up and advances to the `Cp` state. During `Cp`, the system evaluates the PMU signal after completing each layer to verify if adequate energy remains to proceed. If power is insufficient, the microcontroller returns to `LS`. While in `LS`, if the PMU detects that the energy level falls below a predefined

TABLE I
STATE REPRESENTATIONS THROUGH LED INDICATORS.

State	Number	LED 3 Yellow	LED 2 Red	LED 1 Blue
Compute (Cp)	1	0	0	1
Store (Str)	2	0	1	0
Sense (Se)	3	0	1	1
Load (Ld)	4	1	0	0
Light sleep (LS)	5	1	0	1
Deep sleep (DS)	6	1	1	0
Transmit (Tr)	0	0	0	0

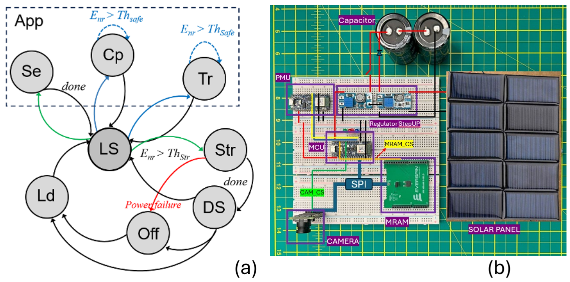

Fig. 1. The proposed **SunSift**. (a) Its finite state machine, and (b) the hardware setup.

threshold, it triggers an interrupt that shifts the system to the Str state. In this state, the system begins to store critical data in the Everspin MRAM [14]. Since a power failure may occur during Str, a dedicated flag in the MRAM is set to true only after all essential values are successfully written, ensuring data integrity. After that, the system transitions to the DS state. In this state, all data is lost; therefore, upon power restoration, the system must reload the previously stored data from MRAM to resume operation. Under normal conditions, following Cp, the system transitions to the transmission phase to send results to the server. However, in the event of a power failure, the system first verifies the MRAM flag. If valid data is present, it reloads the stored state and resumes operation; otherwise, it restarts from the Se state.

III. RESULT AND DISCUSSION

This work forms the foundation of a distributed federated learning framework for IoT devices. Our model is trained on 70% of the CIFAR-10 dataset and utilizes the remaining 30% to distinguish between informative and non-informative samples. The results demonstrate that the trained model can identify informative samples with an accuracy of 71.51%. This accuracy can be further improved by acquiring additional informative samples and fine-tuning the model. To evaluate the model in this work, we printed physical images of different objects and recaptured them using an ArduCam to simulate deployment conditions. As shown in Fig. 2, we tested four samples: bird (a), cat (b), human face (c), and mug (d). The associated informativeness probabilities indicate how familiar the model is with each image: values closer to 1 imply lower informativeness (i.e., the object is well represented in CIFAR-10), while values closer to 0 suggest higher informativeness (i.e., the object is underrepresented or absent in CIFAR-10 and hence more valuable for further training). For instance, the bird (a) and cat (b) images yield high scores of 0.65 and 0.60, respectively, reflecting their strong presence in CIFAR-10. In contrast, the human face (c) and mug (d), with scores of 0.35 and 0.38, are deemed more informative, as these categories

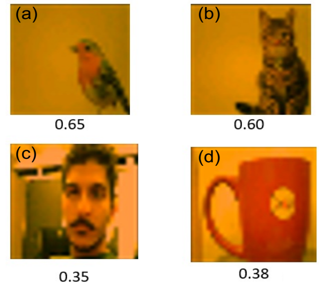

Fig. 2. (a)-(d) Different captured and processed samples.

are not part of the CIFAR-10 dataset, making them useful candidates for improving model generalization.

IV. CONCLUSION

SunSift successfully enables CNN inference on batteryless IoT devices through reactive intermittent execution, achieving 71.51% accuracy in identifying informative samples for federated learning despite frequent power failures. Our deployment on resource-constrained Arduino Nano 33 BLE boards validates the framework's practical viability for sustainable edge intelligence. Future work will explore finer-grained checkpointing and multi-device coordination, establishing a foundation for autonomous, environmentally conscious IoT systems that eliminate battery dependency while maintaining computational sophistication.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation (NSF) under grant no. 2504839, and 2447566.

REFERENCES

[1] P. Verma *et al.*, "A paradigm shift in iot cybersecurity: A systematic review," *IJRASET*, vol. 10, no. 10, pp. 1024–1034, 2022.
[2] Market.us, "Battery recycling market size, share, growth — global forecast report 2023 to 2032," 2023. [Online]. Available: https://market.us/report/global-battery-recycling-market/
[3] A. Banotra *et al.*, "Energy harvesting in self-sustainable iot devices and applications based on cross-layer architecture design: A survey," *Computer Networks*, vol. 236, p. 110011, 2023.
[4] A. S. Sarvestani *et al.*, "Rl-sep: Rl-based smart exit point selection for enhancing energy harvested system longevity," in *ACM SenSys*, 2025, pp. 638–639.
[5] A. Roohi and R. F. DeMara, "Nv-clustering: Normally-off computing using non-volatile datapaths," *IEEE TC*, vol. 67, no. 7, pp. 949–959, 2018.
[6] A. Y. Majid, *et al.*, "Dynamic task-based intermittent execution for energy-harvesting devices," vol. 16, no. 1, pp. 1–24, 2020.
[7] D. Balsamo *et al.*, "Hibernus: Sustaining computation during intermittent supply for energy-harvesting systems," *IEEE ESL*, vol. 7, no. 1, pp. 15–18, 2014.
[8] K. Maeng *et al.*, "Alpaca: Intermittent execution without checkpoints," *Proc. ACM on Programming Languages*, vol. 1, no. OOPSLA, pp. 1–30, 2017.
[9] S. Tabrizchi *et al.*, "Diac: Design exploration of intermittent-aware computing realizing batteryless systems," in *DATE*. IEEE, 2024, pp. 1–6.
[10] S. Sliper *et al.*, "Efficient state retention through paged memory management for reactive transient computing," in *DAC*, 2019, pp. 1–6.
[11] S. Tabrizchi *et al.*, "Src: Sustainable reactive computing for battery-free edge intelligence," in *IGSC*. IEEE, 2024, pp. 93–98.
[12] R. Gaire *et al.*, "Encode: Enhancing compressed deep learning models through feature—distillation and informative sample selection," in *ICMLA*. IEEE, 2023, pp. 633–638.
[13] R. Gaire and A. Roohi, "Carn: Complexity-aware routing network for efficient and adaptive inference," in *CVPR*, 2025, pp. 3318–3326.
[14] S. Aggarwal, "Stt–mram: High density persistent memory solution," in *Proc. Flash Memory Summit*, 2019.