2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI) | 979-8-3503-7025-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/IoTDI61053.2024.00027

Demo Abstract: A Prototype for Machine Learning with Batteryless Sensors

Geffen Cooper
The University of Texas at Austin
Email: geffen@utexas.edu

Tianda Huang
The University of Texas at Austin
Email: thuang1@utexas.edu

Radu Marculescu
The University of Texas at Austin
Email: radum@utexas.edu

***Abstract*—Wearable IoT devices rely on batteries, which pose challenges for long-term sustainable health monitoring due to the need for recharging or replacement. *Batteryless* sensing approaches, which harvest energy from the environment, offer an appealing alternative. However, given the discontinuous supply of harvested energy, it is unclear how to leverage sparse, asynchronous data from batteryless sensors for machine learning (ML) tasks such as human activity recognition (HAR). To this end, we present and profile a prototype of a system to simulate data acquisition from a set of kinetic energy harvesting devices. Our results demonstrate that there is a need to jointly optimize (1) when sensors should spend energy to communicate data, and (2) the training of the ML model that will receive the data.**

***Index Terms*—Batteryless sensing, human activity recognition**

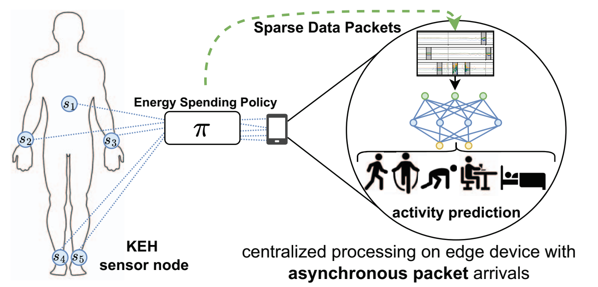

Fig. 1. KEH devices transmit data packets based on their energy policy π. A window of sparse packets is fed into our ML model to predict human activity.

I. Introduction

Batteryless approaches to human activity recognition (HAR) offer a promising solution to scalability and environmental sustainability challenges in long term health monitoring. As wearable HAR targets physical activity, kinetic energy harvesting (KEH) becomes a well suited source for power generation.

From the sensing side, current KEH-based HAR approaches have limited analytics, targeting basic activity prediction with simple input parameters such as packet inter-arrival time [1]. Such approaches do not scale to many activities and cannot differentiate low energy activities (sit vs. stand) as the energy harvested during these periods is not enough for transmitting packets. From the machine learning (ML) side, existing work with sparse data from batteryless devices is limited [2] and does not consider asynchronous data across body parts or the data acquisition process as a component to be optimized.

Given this gap between ML and batteryless sensing, we develop a hardware prototype and simulate data acquisition from a set of wearable KEH devices to understand how the *energy spending policy* of KEH devices impacts the performance of a downstream classification model for continuous HAR.

II. Approach

We outline how we simulate KEH devices to convert standard multi-body part HAR datasets into sparse asynchronous streams of accelerometer packets (see Figures 1 and 2).

Energy Consumption Model: We consider the energy required for device initialization E_{init}, accelerometer sampling and packet transmission E_{packet}, and leakage power P_{idle} as the three components of energy consumption. To turn on, the device must harvest enough energy to initialize. The device remains in a sleep state until it has harvested enough energy to sample and transmit one packet of accelerometer data. With each time step of Δt seconds, the device consumes $P_{\text{idle}} \cdot \Delta t$ J of energy. If the device dies, it must harvest E_{init} again. We profile a real device to obtain these values in section III.

Energy Harvesting Model: We apply a widely used mass-spring-damper model to estimate the kinetic energy that can be harvested from accelerometer traces, following the approach in [3]. This model converts the magnitude of an accelerometer signal to the displacement of a harvester's proof mass, from which power and energy can be calculated. Energy conversion efficiency specifies the fraction of total harvested energy that can be utilized by the device, affecting packet sparsity.

Energy Spending Policy: KEH devices must judiciously control their energy supply using a policy π, which can be a function of current and past device energy levels. Many works use *opportunistic* energy spending policies [1], which transmit data whenever enough energy (E_{packet}) is available. This may be suboptimal for a downstream ML model as the informativeness of the data is dependent on *when* it gets sampled. While an opportunistic policy is *responsive*, it is *greedy* and does not conserve energy for times in which little energy can be harvested. We implement a *conservative* policy which transmits every time the energy *accumulates* by $\alpha E_{\text{packet}},\ \alpha \in [1, 2]$, or if energy does not accumulate for some duration (Figure 2). In section IV we see how *opportunistic* and *conservative* policies impact a HAR classification model.

ML Model: To process *asynchronous* packets across devices, we use a convolutional neural network (CNN) for each

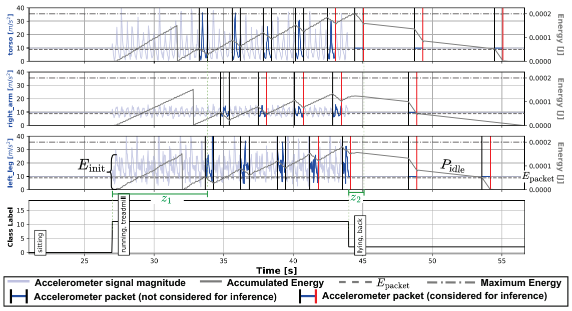

Fig. 2. A window of data acquisition from three body parts in the *DSADS* dataset using a conservative policy. An opportunistic policy would immediately use the energy once hitting E_{packet}. Packets with a red bar show the last $n = 4$ packets per body part that the model sees. Here, n is a hyperparameter corresponding to a 'maximum context length' to consider as input. The green intervals, z_1 and z_2, show the delay from activity start to packet arrival.

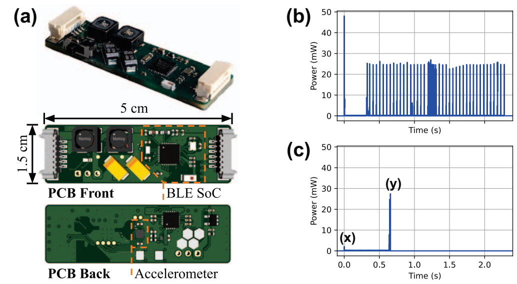

Fig. 3. **(a)** Our hardware prototype. **(b)** Power draw after system power-on and initialization (E_{init}). The initial spike is due to inrush current; the next block is due to the BLE connection process. **(c)** Power draw for accelerometer sampling and packet transmission (E_{packet}). Starting the accelerometer is the first spike **(x)**; packet transmission is the second spike **(y)**.

body part to convert accelerometer packets to embeddings. We use packet arrival time and the body part a packet came from to learn *positional embeddings* which provide spatial and temporal context relative to other packets. The embeddings are fed into a two layer transformer, global average pooling, and a fully connected layer for classification. Overall, the model input is the n most recent packets received along with the time elapsed since the last packet arrival for each body part.

III. Hardware Prototype

Device Design: To validate our energy consumption model, we assemble a custom hardware prototype consisting of a BMA400, a low-power accelerometer, and an nRF52811, a low-power system on a chip (SoC) with Bluetooth low-energy (BLE), integrated on a printed circuit board (PCB) (Figure 3a).

Profiling: E_{init} is measured from power-on until a BLE connection is established. E_{packet} is measured as the period starting from accelerometer power-on and ending once a full packet of accelerometer data (16 samples) is transmitted. P_{idle} is measured by placing the accelerometer and SoC in a low-power sleep state. Results can be seen in Figures 3b and 3c.

IV. Experimental Setup and Results

We evaluate our model by simulating the KEH model on accelerometer traces from the *Opportunity* [4] (5 classes) and *DSADS* [5] (19 classes) datasets. *Opportunity* has natural activity sequences while *DSADS* has isolated activity segments. Thus, we create synthetic activity sequences by randomly sampling activity order and duration (10-25 seconds). We evaluate on *dense predictions* where the model predicts an activity for *every* accelerometer sample in the test sequence. Our metric is Macro F1-score. We use energy conversion efficiencies of 0.3 and 0.5 for *DSADS* and *Opportunity*, respectively, reflecting pessimistic and optimistic values. In [3] they use 0.2, but this creates very sparse data the model could not learn from.

We compare our approach to an **upper bound** (UB), a state-of-the-art HAR model [6] with access to *full, synchronized* data, and a **lower bound** (LB), a multilayer perceptron which *only* uses packet interarrival times (similar to [1]). Table I shows that our models significantly improve over the LB, but are far from the UB. Figure 2 shows that there are regions, z_1 and z_2, where packets are *delayed*, making dense prediction challenging. Furthermore, the conservative policy has sufficient energy to transmit packets when none is harvested, which may explain the improvement over the opportunistic policy.

TABLE I
Macro F1-Score for Dense Prediction

Method	Data	Opportunity	DSADS
Attend and Discriminate (UB) [6]	Full	90.30	81.61
Conservative Policy (**Ours**)	Sparse	50.88	40.35
Opportunistic Policy (**Ours**)	Sparse	40.81	35.64
Interarrival Time Approach (LB) [1]	Sparse	23.84	5.42

V. Conclusion

In this work we have presented a prototype to simulate the data acquisition process for a set of KEH devices to understand the impact of energy spending policies on HAR classification. In our demo, we illustrate how the energy policy significantly impacts the informativeness of the data, hence it should be optimized in conjunction with the ML model.

Acknowledgements

This research was supported by seed funding provided by the Chandra Family Department of Electrical and Computer Engineering at the University of Texas at Austin.

References

[1] Z. Chen *et al.*, "Mp-har: A novel motion-powered real-time human activity recognition system," *IEEE IoT Journal*, 2023.

[2] A. Abedin, S. H. Rezatofighi, Q. Shi, and D. C. Ranasinghe, "Sparsesense: Human activity recognition from highly sparse sensor data-streams using set-based neural networks," in *IJCAI-19*, 7 2019, pp. 5780–5786.

[3] M. Gorlatova *et al.*, "Movers and shakers: Kinetic energy harvesting for the internet of things," *IEEE Journal on SAC*, vol. 33, no. 8, pp. 1624–1639, 2015.

[4] D. Roggen *et al.*, "OPPORTUNITY Activity Recognition," UCI Machine Learning Repository, 2012.

[5] B. Barshan *et al.*, "Daily and Sports Activities," UCI Machine Learning Repository, 2013.

[6] A. Abedin *et al.*, "Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors," *IMWUT*, vol. 5, no. 1, pp. 1–22, 2021.