Poster Abstract: Learning-based Sensor Scheduling for Event Classification on Embedded Edge Devices

Abdulrahman Bukhari
abukh001@ucr.edu
University of California, Riverside
Riverside, California, USA

Hyoseung Kim
hyoseung@ucr.edu
University of California, Riverside
Riverside, California, USA

ABSTRACT

Incremental learning on embedded edge devices is feasible nowadays due to the increasing computational power of these devices and the reduction techniques applied to simplify the model. However, edge devices still require significant time to update the learning model and such time is hard to be obtained due to other tasks, such as sensor data pulling, data preprocessing, and classification. In order to secure the time for incremental learning and to reduce energy consumption, we need to schedule sensing activities without missing any events in the environment. In this paper, we propose a reinforcement learning-based sensor scheduler that dynamically determines the sensing interval for each classification moment by learning the patterns of event classes. The initial results are promising compared to the existing scheduling approach.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; • **Computing methodologies** → *Reinforcement learning*.

KEYWORDS

IoT, embedded edge devices, time-series sensing, scheduling, reinforcement learning

ACM Reference Format:
Abdulrahman Bukhari and Hyoseung Kim. 2023. Poster Abstract: Learning-based Sensor Scheduling for Event Classification on Embedded Edge Devices. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 09–12, 2023, San Antonio, TX, USA*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3576842.3589176

1 INTRODUCTION

Many smart IoT systems require continuous learning. This can be achieved by performing both inference and learning tasks, including feature extraction, classification, clustering and updating the learning model, directly on local edge devices or offloading some heavy workloads (e.g., model updating) to remote cloud servers.

Consider an event classification system with incremental learning capabilities for smart homes (see Fig. 1). This system consists of a set of sensors and one edge device, with each sensor communicating with the edge device that performs inference and incremental learning. The edge device determines when and how often to pull sensor data from the sensors, which we call *sensor scheduling*. This is a challenging problem. The edge device can classify the current event in the environment only when the new sensor data arrives, so a shorter sensing interval is better for classification performance. On the other hand, a longer interval is better to reduce energy for data transmission and to use the resulting idle time to run incremental learning tasks that are essential to detect new event types.

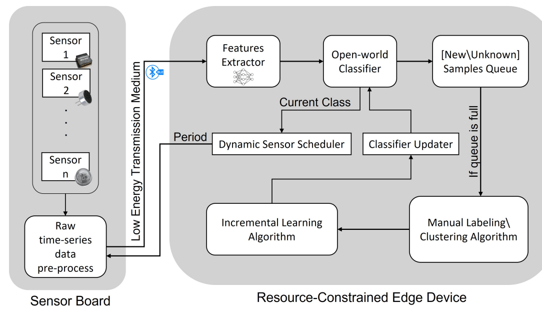


Figure 1: Time-series sensing and incremental learning system [2]

In this work, we propose a reinforcement learning-based dynamic sensor scheduler that aims to minimize the sensing interval while keeping the classification performance within the user's expectations. In the past, we studied a scheduling approach [2] that determines sensing periods for individual event classes. However, that approach yields a fixed period that does not change at runtime, which makes it not suitable for changing environments and leads to energy wastage. There are other studies that used reinforcement learning for sensing systems [1, 3], but their focus is mainly on learning energy patterns, not on sensor scheduling and event classification. Our proposed approach addresses these limitations.

2 PROPOSED METHOD

We present the proposed scheduler that uses Q-learning. The purpose of the Q-learning model is to determine a suitable sensing interval, T_{sp}, given the state of the current event and the previous T_{sp} values. We define the current state as the time it takes for the current event to change to a different event, with the knowledge of prior classification results. This time is defined as T_{ideal}. We want to maximize T_{sp} such that,

$$T_{sp} - T_{ideal} \leq CL \tag{1}$$

where CL is the classification latency constraint, i.e., the maximum delay a user can tolerate to detect a new event.

Eq .(1) is considered as the first criterion Cr_1 in training the Q-learning model. The purpose of this is to make the scheduler

IoTDI '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05.
https://doi.org/10.1145/3576842.3589176

Table 1: Q-learning model with different Cr_1 and Cr_2 reward/penalty (R/P) ratios

Cr_1 (R/P)	Cr_2 (R/P)	% Trans. Reduction	CL Missed
50/10	5/1	12%	7
10/50	5/1	19%	0
50/50	5/1	12%	1
50/10	10/1	13%	2
50/10	1/10	6%	20
50/10	50/10	4%	34
50/50	20/20	8%	13
20/20	50/50	1%	34
50/50	30/30	12%	1

pick a T_{sp} value between the range of values that meet the CL constraints, e.g., $T_{sp} \in [0 - 100]$ seconds. The second criterion Cr_2 is that the scheduler should try to choose a T_{sp} value that is larger than or equal to the previous T_{sp} in order to increase the idle time until the class of the current event changes. Each criterion is parameterized by the ratio between a reward and a penalty when the condition is met. The ratios within each criterion and between them impact the performance of the proposed scheduler to meet the user's preference: increasing the energy efficiency or lowering the classification delay.

3 EVALUATION

We performed two experiments to evaluate the proposed learning-based scheduler. The first one is to understand the impact of the reward/penalty ratio of each criterion on the scheduler. The second experiment is to compare the classification latency and energy efficiency (idle time) under the proposed learning-based dynamic scheduler and our prior static scheduling approach [2].

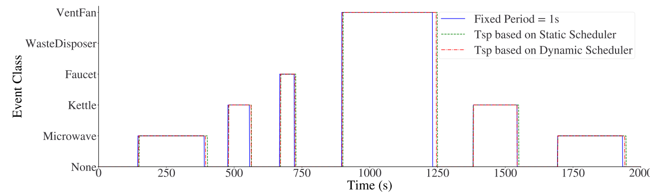
Dataset We used a simulated dataset that contains six different classes of events that can be captured by commodity IoT sensors from a user environment. The six classes include [None, Microwave, Kettle, Faucet, Waste Disposer, Vent Fan] and the CL value selected for each class is [5,12,15,5,8,15] seconds, respectively. The training set contains 7,000 seconds of sensor reading at each second, randomly generated with different durations for each event. The testing dataset is randomly generated at each test, with a range of different lengths between 5,000-8,000 seconds.

3.1 Model Criteria Study

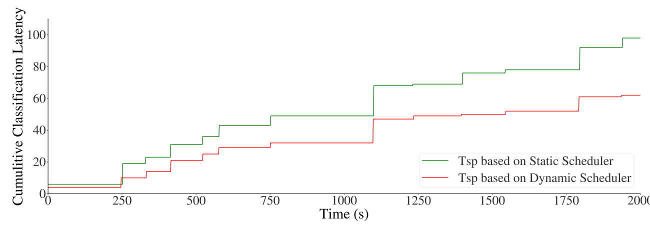
In the first experiment, we train the Q-learning model with different penalty/reward ratios in different set-ups. We fix Cr_2 and alter Cr_1, and vice-versa, to see the effect of each criterion. We also alter both ratios simultaneously to understand their performance impact. In Table 1, we only show the combinations that have a large impact on the scheduler performance. In the first 6 rows of Table 1, we see that changes in Cr_1 do not impact the performance significantly while changes in Cr_2 do. However, Cr_1 can regulate Cr_2 to achieve the desirable results, as can be seen in the last 3 rows.

3.2 Classification Latency and Idle Time

We compare the classification latency of the proposed learning-based dynamic scheduler and the existing static scheduler. In Fig. 2a, we simulate 2,000 seconds of data transmission between the sensor



(a) Classification latency of static and dynamic schedulers (ground truth = fixed period of 1s)



(b) Accumulated classification latency under static and dynamic schedulers

Figure 2: Classification latency comparison

and an edge device. Although both schedulers are able to follow the original pattern of the sensor data, the proposed dynamic scheduler slightly outperforms the static one. It is more clear in Fig. 2b that the static scheduler accumulates more latency over time.

In addition, the proposed dynamic scheduler needed only 9% of active time for the edge device, i.e., the device can idle or sleep for the remaining time, which is 1% better than the static scheduler. Although the difference is small, the proposed scheduler is superior in meeting the CL constraint (Eq. (1)), with only 4 CL misses over the entire test time span while the static scheduler missed 21 CL.

4 CONCLUSION

This work proposes a reinforcement learning-based dynamic sensor scheduler for event classification on embedded edge devices. The evaluation results are promising compared to our prior work. The model can be further improved by optimizing the reward/penalty ratio. We plan to investigate the cost of running Q-learning on edge devices compared to the static scheduler with a real dataset rather than the simulated data we used.

ACKNOWLEDGMENTS

This work was sponsored by NSF (1943265), NIJ (2019-NE-BX-0006), and NIFA (2020-51181-32198).

REFERENCES

[1] Fayçal Ait Aoudia, Matthieu Gautier, and Olivier Berder. 2018. RLMan: An Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks. *IEEE Transactions on Green Communications and Networking* 2 (2018), 408–417.

[2] Abdulrahman Bukhari, Seyedmehdi Hosseinimotlagh, and Hyoseung Kim. 2022. An Open-World Time-Series Sensing Framework for Embedded Edge Devices. In *2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*.

[3] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, and Rajesh K. Gupta. 2020. ACES: Automatic Configuration of Energy Harvesting Sensors with Reinforcement Learning. *ACM Trans. Sen. Netw.* 16, 4, Article 36 (jul 2020), 31 pages.