

# Origin: Enabling On-Device Intelligence for Human Activity Recognition Using Energy Harvesting Wireless Sensor Networks

Cyan Subhra Mishra, Jack Sampson, Mahmut Taylan Kandemir, Vijaykrishnan Narayanan  
The Pennsylvania State University, USA. (Email: {cyan, jms1257, mtk2, vijaykrishnan.narayanan}@psu.edu)

**Abstract**— There is an increasing demand for performing machine learning tasks, such as human activity recognition (HAR) on emerging ultra-low-power internet of things (IoT) platforms. Recent works show substantial efficiency boosts from performing inference tasks directly on the IoT nodes rather than merely transmitting raw sensor data. However, the computation and power demands of deep neural network (DNN) based inference pose significant challenges when executed on the nodes of an energy-harvesting wireless sensor network (EH-WSN). Moreover, managing inferences requiring responses from multiple energy-harvesting nodes imposes challenges at the system level in addition to the constraints at each node.

This paper presents a novel scheduling policy along with an adaptive ensemble learner to efficiently perform HAR on a distributed energy-harvesting body area network. Our proposed policy, *Origin*, strategically ensures efficient and accurate individual inference execution at each sensor node by using a novel activity-aware scheduling approach. It also leverages the continuous nature of human activity when coordinating and aggregating results from all the sensor nodes to improve final classification accuracy. Further, *Origin* proposes an adaptive ensemble learner to personalize the optimizations based on each individual user. Experimental results using two different HAR data-sets show *Origin*, while running on harvested energy, to be at least 2.5% more accurate than a classical battery-powered energy aware HAR classifier continuously operating at the same average power.

**Index Terms**—Energy Harvesting, Human Activity Recognition, DNN, Wireless Sensor Network, Ensemble Learning

## I. INTRODUCTION

The advent of data driven computing, along with advances in low-power computing platforms, has given rise to the new generation of intelligent and connected devices that comprise the internet of things (IoT). These devices have become an integral part of our daily lives and, using techniques such as deep learning, these devices are becoming increasingly capable of performing complex inference tasks including machine translation, human activity recognition (HAR), bio-metric authentication, ECG measurement, fall detection etcetera [1], [2]. These inference tasks are typically driven by deep neural networks (DNNs), which are known for being compute heavy and power hungry [3]. Given the power and compute constraints of the IoT devices performing sensing, it is difficult to execute these inference tasks on the sensing device itself, excepting a few intermittent tasks such as bio-metric authentication. Instead, to perform complex and continuous inference, such as HAR, the data is typically offloaded either to the cloud or

to a nearby host device which in turn executes the inference or further redirects it [4] and, finally, returns the results to the IoT devices responsible for data display or actuation, dependent on the inference task.

Recent works [5], [6] suggest that processing data at the source is more efficient than sending them to the cloud and getting the results back, owing to the power and latency overhead of data communication. They propose optimizations to efficiently execute the DNNs on low power IoT devices [7], [8]. Other recent works [5], [7]–[9] have proposed using energy harvesting (EH) solutions to provide additional energy and increase the battery life in IoT devices. These works provide software, hardware and compiler-level solutions, which can be applied to build a battery-less system working entirely on harvested energy. Moreover, in addition to prolonging device lifetime, energy harvesting can help us reduce the environmental impact of batteries [10]. However, energy harvesting is no panacea due to the fickle nature of harvested energy. To tackle this, recent works [6], [9] use a non-volatile processor (NVP) to ensure sufficient forward progress in the face of frequent power emergencies.

The combination of EH, NVPs and other architectural and compiler optimizations have enabled the use of sensors as smart inference engines. However, these node-level optimizations are not entirely sufficient for sensor networks with multiple sensors collectively working together to achieve a goal, which are very common. Although fusing sensor data is not uncommon, it requires one central location where the inference can take place, requiring the communication of sensed data. In networks of energy harvested sensors, the power-hungry nature of communication results in intermittent coordination failures due to one or more of the sensors, or even the fusing node itself, lacking sufficient energy at the time that inter-node communication is required. This work aims to address this limitation by pursuing answers to the following questions - 1) *how do we leverage multiple available energy harvesting wireless sensors collectively*, and 2) *where should each individual sensor perform its own inference, considering that they collectively perform a single task?*

Our approach to address these questions relies on decentralizing the DNN execution and letting each sensor perform its own inference. These sensors, each individually working as a weak classifier, can together form an ensemble learning

environment to achieve better accuracy with lower communication overhead. For each sensor to perform inference using the limited and unstable harvested energy poses a scheduling problem, as non-deterministic time is required for the EH sensors to accumulate enough energy to perform the inference. This scheduling is made even more difficult as each sensor can harvest and consume different amounts of energy depending upon their location, have different sensor sampling rate, and require different DNNs to be executed. Further, all sensors might not be able to participate in the ensemble due to the fickle nature of harvested energy. This demands the aggregation process for the ensemble to be robust, yet light weight in order to perform accurate classification with minimum overhead. Therefore, this work proposes an intelligent scheduler along with efficient ensemble learning to enable DNN inference in a distributed energy harvesting wireless sensor network (EH-WSN).

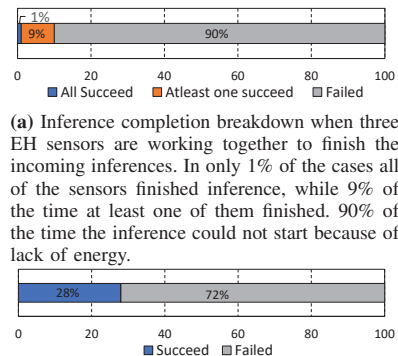
This work proposes a novel policy, *Origin*, which enables energy harvesting wireless sensors to perform efficient and accurate DNN inference. Specifically, *Origin* targets inherent features of sensor data from distributed body area networks in human activity recognition (HAR) tasks and leverages non-volatile processing, intelligent scheduling for energy-harvesting sensor nodes, and ensemble learning to classify human activity with minimum accuracy loss compared to a state-of-the-art battery powered system. To the best of our knowledge, this is the first work that tries to enable DNN inference for human activity recognition in a distributed energy harvesting wireless sensor network by leveraging ensemble learning. The paper makes the following key contributions:

- 1) We design a scheduling policy that chooses the salient sensor for performing the inference depending on the anticipated activity, i.e. the scheduler is *activity aware*.
- 2) We leverage temporal continuity of human activity, and persist the last successful classification result of a sensor. We use aggressive recall which reduces the number of total inferences performed and mitigates the requirement that all of the sensors be involved in the ensemble process during each inference.
- 3) Our proposed policy, *Origin*, combines an adaptive confidence matrix and the activity aware scheduler to perform efficient and accurate classification. The adaptive confidence matrix, which weights the output of each sensor depending upon the classification result, is updated on each successful classification.
- 4) Finally, we provide a detailed evaluation of *Origin*, and show that, even when powered by an unreliable EH source, the efficiency achieved by the this system results in better accuracy than that of a fully powered system running state of the art classifiers optimized for energy efficiency. *Origin* reaches 83.88% top-1 accuracy compared to the 81.16% accuracy of the baseline system.

## II. MOTIVATION

A major challenge while executing a DNN inference on an energy harvesting sensor is the power budget. The conventional

method, where the sensors collect the data and send it to the cloud or any other host device (such as connected mobile phones) is not an effective option as communicating large data demands more power, which is both highly variable and scarce in EH systems. Therefore, the better option, from a communication cost perspective, is to execute the inferences on the individual sensors and use an ensemble learning method (like majority voting) to aggregate these results for the final classification.



(a) Inference completion breakdown when three EH sensors are working together to finish the incoming inferences. In only 1% of the cases all of the sensors finished inference, while 9% of the time at least one of them finished. 90% of the time the inference could not start because of lack of energy.

(b) Inference completion breakdown when three EH sensors are working in round robin fashion, where one of the sensors performs inference while the other two are accumulating energy. 28% of the time the sensors could finish the inference, while 72% of the time the inference failed as the sensor could not harvest enough energy while not performing any inference.

**Fig. 1:** Fraction of inference completed on harvested energy using naïve scheduling.

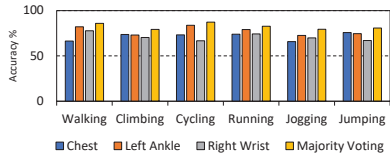
Each of the sensors in a multi-device HAR deployment receives different data depending on its location and the current human activity in progress. Therefore, different DNNs are needed to process data from these different locations. Consequently, the power requirement and the latency of these DNNs may vary and synchronizing them for collective execution would require scheduling that addresses these differences in resource requirements. Even if we were able to design a proper scheduling policy, for a conventional ensemble, all the sensors involved need to finish their computation. However, our preliminary results using the hardware setup of [6] and the DNN from [11] on the MHEALTH [12], [13] dataset suggests that only 10% of inferences could be completed in a WiFi powered system (Fig.1a). Therefore, we cannot always expect inference outcomes from all the sensors while doing HAR on EH-WSN. Clearly, the completion of the task is power bound: Adopting a wait-compute execution model, such that we have enough energy to complete some results, at a lower duty cycle, instead of always trying and failing would yield benefits. This leaves us with the following important questions:

- Are continuous inferences essential, or can we leverage the workload itself to skip some inferences without substantial accuracy loss, allowing enough energy to be accumulated for future inferences?
- Since all the sensors cannot be activated together due to the limited power, how do we effectively perform the ensemble?

## III. *Origin*: AN INTELLIGENT SCHEDULER MEETS A LIGHT WEIGHT AND ADAPTIVE ENSEMBLE LEARNER

We design a EH-WSN setup for HAR, where the user has three EH inertial measurement units (IMUs), at the

chest, left ankle and right wrist<sup>1</sup>. It is obvious that the raw data sensed by each sensor would be different, even for the same activity, because of the difference in movement and dynamics. For example, while cycling, the data sensed by the ankle, chest and wrist sensors would be entirely different because of the nature of the motion. Thus, the DNN architectures to infer these data are also different.

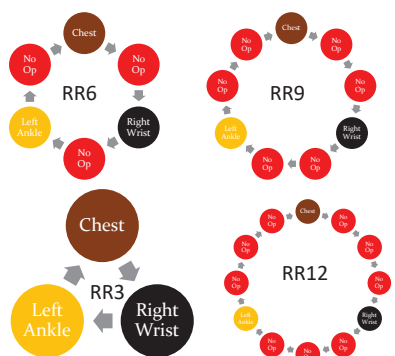


**Fig. 2:** Accuracy of the individual DNNs and with a majority voting ensemble for different activities.

energy-scarce applications. Fig. 2 gives the accuracy of these DNNs on MHEALTH dataset [12], [13]. A detailed description of the setup is explained in Section IV. In this section, we provide an overview of our proposed solution.

### A. Preamble to Origin

Human activity has temporal continuity, i.e. most activities last for some duration (in the range of hundreds of milliseconds to seconds). Therefore, there is an opportunity to skip some intermediate inferences over the period of an activity in order to increase harvesting duration and the probability that an initiated inference will complete. So long as the number of skipped inferences is modest, there will still likely be samples processed before an activity finishes. This can be extended further adopting a round-robin activation schedule to both increase harvesting periods per initiated inference on each node while increasing the odds that at least some node is attempting an inference at any given time.



**Fig. 3:** Different flavors of (extended) round-robin scheduling and their execution flow. Each policy is named after the number of nodes the cycle has, i.e. RR3 has 3 nodes with no no-ops and RR6 has 3 nodes with 3 no-ops.

We refer to this policy that stretches the basic round-robin policy as extended round-robin (ER-r). Using ER-r, we can complete more total inferences,

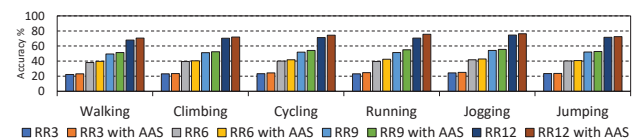
<sup>1</sup>This can also be extended to larger numbers of sensors and modalities

but this design is limited by the accuracy of individual sensors. Moreover, since all sensors are not equally capable of classifying each activity with same accuracy (Fig. 2), ER-r might lead to lower accuracy in many cases.

A better approach is to prioritize performing inferences on the sensor that has the highest local accuracy for the current activity. However, this poses a chicken and egg problem – to know which sensor is the best for classifying an activity we need to know what activity is going to be performed beforehand. However, while perfect future knowledge remains impossible, in the context of HAR, we can anticipate the next activity from the previous activity with high confidence. Intuitively, human activities do not usually stop abruptly, i.e. if a person is walking and has taken a step, there is a high probability that the person will continue walking rather than immediately switch to another activity. Therefore, to classify the next possible activity, we activate the sensor which is most accurate for classifying the anticipated activity. This motivates us to develop an *activity-aware scheduling* (AAS) policy which aims to activate the best suited sensor for the anticipated activity.

### B. Activity Aware Scheduling

To enable the activity awareness we keep a small lookup table of accuracy of all the sensors over all the classes. However, accuracy being a floating point number, is expensive in terms of energy to store and lookup. To minimize this overhead, instead of storing the accuracy, we store the rank of the sensors for individual activities. After a sensor detects an activity, it anticipates the next activity to be the current classified activity, looks up for the best sensor, and signals to activate it for the upcoming inference. However, this leads to another potential issue - what if the current inference is running on the best sensor, and the sensor does not have enough energy to run the next inference? In this case, the current sensor chooses the next best sensor for the job and signals it. The other sensor receives this as an external signal and activates itself to classify the activity. To incorporate the ER-r, we induce delays between sending the external signal and starting the inference on the same sensor. This delay depends of the extended round-robin policy. Combination of ER-r and AAS, results in more than 70% accuracy for most of the activities (Fig. 4).



**Fig. 4:** Accuracy results for AAS combined with ER-r.

Even though AAS provides significantly better results compared to standard round-robin, it is still unable to incorporate ensemble learning. The major challenge is the inability to run inferences in all the sensors simultaneously because of the harvested energy budget. Therefore, we need to find the classification result for all the sensors without activating them. Extending our assumption from AAS, we hypothesize that the most recent classification result of a sensor must be a good

representation of what its inference would be for the current activity. Hence, by memorizing or *recalling* the most recent classification result, we can get the inference result of a sensor even without activating it. Even though the sensors are running in the round-robin fashion, the non-participating sensors can still impact the classification result by virtue of *recalling* their most recent classification. Combining the *Recall* with AAS (which we term as AASR - Activity Aware Scheduling with Recall) opens possibilities for getting a more accurate classification. To minimize the communication overhead, and to ensure participation of all sensors, we build the *recall* strategy into the host device. The host device remembers the most recent classifications by all of the sensors. After receiving or recalling prediction from all sensors, the host performs a majority voting for the final classification.

AASR thus bridges the major gaps in the design that we intend to achieve. With the addition of *recall*, we have a fully functional ensemble learning system on a EH-WSN. AASR intelligently takes advantage of multiple DNNs (by bringing in activity aware scheduling), leverages the workload (by considering the activity continuity via recall and extended round-robin strategies), and also poses minimal overhead on the host device for running aggregation. However, as we did not want to burden the host device with complex computation, the aggregation task is very naïve in that it just performs recall and incorporates no intelligence. Hence, there is an opportunity to also improve the ensemble technique.

### C. Designing an Adaptive Ensemble Learner

AASR scheduling solves most of the issues on the sensor side without burdening host device, yet the host device still performs a naïve majority voting-based ensemble. Designing any sophisticated ensemble learning technique will either consume more resources of the host device, or need more information and computation at the edge, thus making our effort of finishing the inference at the edge not viable. However, if we can design a simple, light weight and adaptive ensemble technique, then our design will be holistic from both the sensor and the host side. The current scheduler is activity aware, i.e. while performing an inference it always tries to choose the best available sensor to perform the task at hand. Furthermore, the AASR poses negligible overhead both in terms of compute and memory. Our goal is to develop an activity aware ensemble technique which can further improve accuracy, when compared to AASR.

The idea of making the ensemble task activity aware has similarities to weighted majority voting, where we assign weights to the individual learners participating in the ensemble, such that a higher accuracy classifier contributes more weight towards the final result. However, from Fig. 2 it is clear that not all the sensors are equally good at classifying various activities; in fact, this builds the foundation of AASR. Therefore, assigning a static weight to the output of each classifier will not reflect that its accuracy is activity-dependent. For example, the classifier used in the left ankle sensor tends to be more accurate overall, but for classifying climbing action, it is not better than the chest sensor. Hence, to give the left ankle more weight while doing an ensemble for a climbing task makes

the classifier biased. Furthermore, the relative weight of each sensor is likely to shift from user to user. A simple solution is to assign the accuracy of each of the sensors for every class as its weight. Although accuracy is a close measurement of the confidence of the classification, it does not truly reflect it.

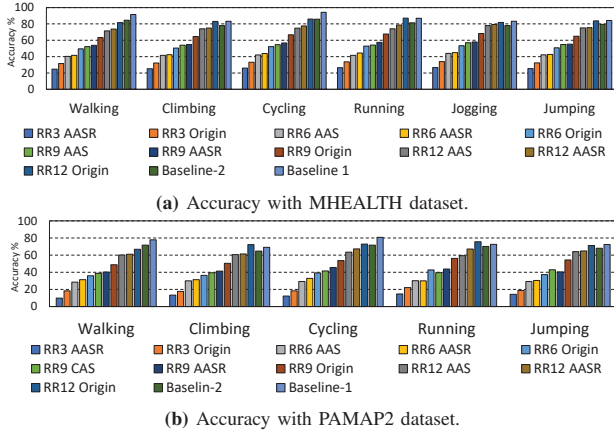
For example, let us consider two DNN classifiers ( $C_1$  and  $C_2$ ) classifying between 4 different classes ( $o_1, o_2, o_3, o_4$ ). The final probability vector from the last layer (softmax function)  $V_{C_1} = [0.94, 0.01, 0.02, 0.01]$ , and  $V_{C_2} = [0.80, 0.05, 0.08, 0.07]$ . Both the models have classified the input to be of class  $o_1$ . The accuracy of both system might be identical (over a large number of test sets), yet for the current test case, both the models are not equally confident about the classification. The question is, how do we measure the confidence of the given classification? It is obvious that the most confident classification for the same class would be  $[1, 0, 0, 0]$ , where the model is 100% confident on class  $o_1$  and the most confused prediction would be  $[0.25, 0.25, 0.25, 0.25]$ , where the classifier is equally confused between all the classes. Therefore, a good metric for the confidence would be the variance of the output probability vector. The higher the variance the more confident is the classification. Towards this, we build a lookup table by averaging the variance of output vectors of multiple test cases. This table, which we call the *confidence matrix*, gives us the confidence of each sensor for each class, and can be used as a weight for majority voting.

The next challenge is to adapt the confidence matrix for individual users. Each user has unique expressions of behaviour classes reflected in the sensor data. For example, gaits of two different people may significantly vary, and might be entirely different from the training data. Thus, it is important to keep learning and adapting to the user behavior. Since, we cannot keep re-training the DNNs because of their resource constraints, we choose to periodically update the confidence matrix. The initial confidence matrix, derived from the test cases, would be programmed into the host device. Further, after each successful classification, the sensors would send the confidence score for that classifier along with the output class. This confidence score would further update the weight matrix of the host device using a moving average method and keep updating it as the user keeps using the device.

### D. Origin: AASR meets Confidence Matrix

Combined together, the activity aware scheduler with recall (AASR) and the adaptive confidence matrix we present *Origin*: a holistic system where an intelligent scheduler meets an adaptive ensemble learner. This design optimizes the DNN execution in an energy harvesting wireless sensor network by collectively looking into all the involved components. The DNNs as individuals are optimized before to meet the power budget. In the earlier case of naïve scheduling we tried to build an efficient DNN by applying energy aware pruning [15]. However, since *Origin* follows an activity aware scheduling with extended round-robin, it can relax the power constraint pruning if needed. Instead of restricting the power constraint to the average power of the entire power trace, the constraint can be relaxed to the average power requirement of the extended round-robin





**Fig. 5:** Accuracy results of the different policies described in Section III. RR indicates the extended round-robin policy in use, e.g. RR6 AAS represents AAS with RR6.

policy in use. Further, the scheduling strategy was modified using activity aware scheduling with extended round-robin such that all the sensors get enough time to harvest and also the best possible sensor works on the classification task at hand instead of any arbitrary sensor. The added recall functionality enables ensemble learning. Moreover, the host device, which performs the ensemble, is equipped with a confidence matrix, which adapts to the user and performs weighted majority voting instead of a naïve majority voting. The associated confidence matrix boosts the classification accuracy and also resolves ties while voting.

#### IV. EVALUATION

In this section we explain the strategy for evaluating *Origin*. We discuss about the hardware and software framework, and the accuracy of *Origin* compared to two different baselines.

##### A. Energy Harvester and Sensor Setup

Our evaluation setup consists of three sensors at three different locations. First sensor at the chest, second on the right wrist and last sensor on the left ankle. Each sensor consists of four major components, namely, the sensing component, an IMU, which collects acceleration and attitude data, an energy harvester which harvest the surrounding RF (WiFi) energy, a compute component same as [6] and a wireless communication module (BLE or WiFi) to connect to a host device (battery backed mobile phone). We assume the communication cost to be negligible since it infrequently sends a few bytes of data to the host. To replicate the energy harvesting, we use a real power trace harvested from a WiFi source while doing various day to day tasks in an office environment [6]. The specifics of the energy-harvesting mechanism producing the power trace are beyond the scope of this work.

##### B. DNN Classifier and the Dataset

Our DNN design choices are inspired from the works in [11], [14]. However, instead of designing a centralized DNN which processes the combined data from all the sensors, we design three different smaller DNNs that work on their individual data.

Further, to build an energy efficient version of the DNNs, we applied the energy-aware DNN optimizations proposed in [3], [15]. We use two different datasets, MHEALTH [12], [13], and PAMAP2 [16], [17], for our evaluation which follow the similar sensor setup described in Section IV-A. The DNNs were trained on the training data-sets using the Keras [18] framework.

##### C. Accuracy Results

**Baseline:** We choose two baselines for our evaluation:

- 1) Baseline-1 consists of the original DNNs built along the lines of [11], [14] (without any pruning).
- 2) Baseline-2 uses state of the art pruning techniques described in [3], [15] to prune the DNNs of Baseline-1 to fit the average harvested power budget from our harvesting trace described in Section IV-A.

Both the baseline setups run on a fully powered system equipped with a steady power source. A majority voting ensemble method is used in both of these baselines to mimic ensemble learning. *Origin* uses the DNNs of Baseline-2 for the classification tasks. We plot the accuracy of different strategies described throughout the paper. Fig. 5a shows the accuracy results on the MHEALTH dataset, and Fig. 5b shows the results for the PAMAP2 dataset. Following are our observations:

- The overall accuracy tends to improve with increasing round-robin delay time. This behaviour is expected and can be attributed to the increasing number of completed inferences. The nature of the workload itself gives us an opportunity to not perform inference at a rapid rate. Further evaluations suggest *Origin* with RR-12 to be the best fit for HAR. Going beyond RR-12 might lead to missing an activity window for high intensity or rapid activities, and going below RR-12 might lead to energy scarcity at times. In case of abundant energy supply, one can use a round robin policy fit for the given EH source.

- Table I shows the accuracy comparison between the RR12-*Origin* with both the baselines. We observe that, for the MHEALTH dataset, RR12-*Origin* is 2.72% more accurate than the Baseline-2. For the PAMAP2 data-set, RR12-*Origin* is 2.53% better than Baseline-2. For certain cases like climbing in PAMAP2, and running in MHEALTH, *Origin* is more accurate than Baseline-1. These accuracy improvements can be attributed to *Origin's* use of a confidence matrix in classification, as opposed to the baseline models, which only perform majority voting based ensembling. Note that both the baselines are running on a fully powered system whereas *Origin* runs entirely on harvested energy. In practice, we can extend *Origin* further to other multi-sensor data-sets for HAR.

**Discussion:** Although *Origin* is proposed and works for energy harvesting wireless sensor networks, it can also be used with battery-powered or hybrid (a combination to battery powered and EH) systems to minimize the energy footprint while maximizing the accuracy. Furthermore, it uses multiple sensors effectively and hence poses minimum risk if one of the sensors fails. This makes *Origin* versatile and suitable for systems whose intermittence comes from either or both of power or device reliability limitations, both of which will be key factors in real-world IoT deployments. Moreover, ensemble learning

Activity	Policy Comparison				
	RR12 Origin	BL-2	BL-1	vs BL-2	vs BL-1
Walking	81.60896	84.46	91.56	-2.85104	-9.95104
Climbing	83.10679	77.93	83.24	5.176789	-0.13321
Cycling	85.88992	85.81	94.27	0.079918	-8.38008
Running	87.13474	81.29	86.91	5.844736	0.224736
Jogging	81.81809	78.04	83.17	3.778086	-1.35191
Jumping	83.69378	79.42	84.26	4.273776	-0.56622

**TABLE I:** Comparing RR12 Origin with both the baselines on MHELATH dataset. BL indicates the baseline models.

techniques combined with efficient scheduling occasionally gives more accuracy than a larger and unpruned centralized DNN that is more failure-prone and power hungry.

#### D. Adaptive Ensemble Learner

As discussed earlier in Section III-D, origin uses a lightweight and adaptive ensemble learner that performs weighted majority voting based on a confidence matrix. The confidence matrix adapts and learns from the user pattern. It is obvious that it is not feasible to train a DNN for all types and variances of human actions. Even for the same types of activity, some attributes will vary from user to user.

These variations can cause misclassifications and the adaptive nature of the confidence matrix mitigates this. To mimic the noisy and inconsistent behaviour of real-world scenarios, we test the adaptive nature of the ensemble learner for 3 different previously unseen users over a 1000 iterations (10000 successful classifications; each iteration has 10 classifications). The noisy data is generated by adding a Gaussian noise (with maximum SNR of 20dB) over the unseen test data. The first iteration shows the accuracy with the unchanged confidence matrix. Even though the accuracy claim of the models was nearly 85%, in the first iteration, the accuracy drops below 80% because of the added noise. As the confidence matrix gets updated with the newer confidence values sent from the sensor, we can see (from Fig. 6), that *Origin* keeps up with the claimed accuracy (base accuracy), and at times outperforms it. We can attribute this to the confidence matrix and ensemble learner, since over these 1000 iterations, only the confidence matrix gets updated. Note that the confidence matrix reaches the steady state of baseline accuracy within 100 iterations. This, in turn, will lead to better and more stable classification for every individual without extensive need for retraining and updating the DNN, which might be impractical for EH-WSNs due to the high communication cost while updating the parameters.

Fig. 6: Accuracy over time for different users: the confidence matrix adapts to the behaviour and activity pattern of the user and learns over time to give stable if not better accuracy.

#### V. CONCLUSION

Enabling DNN inference on edge devices has been gaining recent traction, especially for tasks like HAR. However, the compute heavy DNNs make it challenging because of their

power requirements, especially in EH-WSNs. Our proposal, *Origin*, holistically looks into multiple aspects of deploying a DNN on an EH-WSN for the purpose of HAR. *Origin* combines an intelligent activity aware scheduler with an adaptive and light weight ensemble learning method. Our experiments shows that DNN inference using *Origin*, running on a harvested energy only system, is more accurate than energy-constraint-optimized DNNs, running on a fully-powered system. Although the current work is limited to HAR, this can further be extended to many suitable tasks which need to leverage a distributed sensor system for DNN inference. We believe that the co-optimization of deep learning and energy harvesting techniques for edge devices will further invigorate research on the next generations of intelligent and sustainable IoT platforms.

#### VI. ACKNOWLEDGMENTS

This work was supported in part by Semiconductor Research Corporation (SRC), Center for Brain-inspired Computing (C-BRIC) and NSF Grant #1822923 (SPX: SOPHIA).

#### REFERENCES

- [1] "Taking an eeg with the eeg app on apple watch series 4 or later," 2020, <https://support.apple.com/en-us/HT208955>.
- [2] "Use fall detection with apple watch," 2020, <https://support.apple.com/en-us/HT208944>.
- [3] T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam, "Netadapt: Platform-aware neural network adaptation for mobile applications," in *ECCV*, September 2018.
- [4] "Google assistant for wearables," 2020, <https://assistant.google.com/platforms/wearables/>.
- [5] K. Ma, X. Li, S. R. Srinivasa, Y. Liu, J. Sampson, Y. Xie, and V. Narayanan, "Spendthrift: Machine learning based resource and frequency scaling for ambient energy harvesting nonvolatile processors," in *2017 (ASP-DAC)*, 2017, pp. 678–683.
- [6] K. Qiu, N. Jao, M. Zhao, C. S. Mishra, G. Gudukbay, S. Jose, J. Sampson, M. T. Kandemir, and V. Narayanan, "Resirca: A resilient energy harvesting reram crossbar-based accelerator for intelligent embedded processors," in *2020 HPCA*, 2020, pp. 315–327.
- [7] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence beyond the edge: Inference on intermittent embedded systems," in *ASPLOS*. ACM, 2019.
- [8] K. Maeng and B. Lucia, "Adaptive dynamic checkpointing for safe efficient intermittent computing," in *OSDI*. USENIX Association, 2018.
- [9] K. Ma, X. Li, S. Li, Y. Liu, J. J. Sampson, Y. Xie, and V. Narayanan, "Nonvolatile processor architecture exploration for energy-harvesting applications," *IEEE Micro*, vol. 35, no. 5, pp. 32–40, 2015.
- [10] K. Ma, X. Li, J. Li, Y. Liu, Y. Xie, J. Sampson, M. T. Kandemir, and V. Narayanan, "Incidental computing on iot nonvolatile processors," in *MICRO*, 2017.
- [11] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *IJCNN*, 2016.
- [12] O. Banos, C. Villalonga, R. García, A. Saez, M. Damas, J. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *BioMedical Engineering OnLine*, 2015.
- [13] O. Baños, R. García, J. A. H. Terriza, M. Damas, H. Pomares, I. R. Ruiz, A. Saez, and C. Villalonga, "mhealthdroid: A novel framework for agile development of mobile health applications," in *IWAAL*. Springer, 2014.
- [14] F. M. Rueda, R. Grzeszick, G. A. Fink, S. Feldhorst, and M. ten Hompel, "Convolutional neural networks for human activity recognition using body-worn sensors," *Informatics*, 2018.
- [15] T. Yang, Y. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *CVPR*. IEEE, 2017.
- [16] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *ISWC*. IEEE, 2012.
- [17] —, "Creating and benchmarking a new dataset for physical activity monitoring," in *PETRA*, F. Makedon, Ed. ACM, 2012.
- [18] F. Chollet *et al.* (2015) Keras. [Online]. Available: <https://github.com/fchollet/keras>