EMI: Energy Management meets Imputation in Wearable IoT Devices

Dina Hussein, *Student Member, IEEE,* Nuzhat Yamin, *Student Member, IEEE,* and Ganapati Bhat, *Member, IEEE*

***Abstract*—Wearable and Internet of Things (IoT) devices are becoming popular in several applications such as health monitoring, wide area sensing, and digital agriculture. These devices are energy-constrained due to limited battery capacities. As such, IoT devices harvest energy from the environment and manage it to prolong operation of the system. Stochastic nature of ambient energy, coupled with small battery sizes may lead to insufficient energy for obtaining data from all sensors. As a result, sensors either have to be duty cycled or subsampled to meet the energy budget. However, machine learning (ML) models for these applications are typically trained with the assumption that data from all sensors are available, leading to loss in accuracy. To overcome this, we propose a novel approach that combines data imputation with energy management. Data imputation aims to substitute missing data with appropriate values so that complete sensor data are available for application processing, while energy management makes energy budget decisions on the devices. We use the energy budget to obtain complete data from as many sensors as possible and turn off other sensors instead of duty cycling all sensors. Then, we use a low-overhead imputation technique for unavailable sensors and use them in ML models. Evaluations with six diverse datasets show that the proposed EMI approach achieves 25-55% higher accuracy when compared to duty cycling or subsampling without using additional energy.**

I. Introduction

Wearable and Internet of Things (IoT) devices, coupled with machine learning (ML) are transforming multiple facets of human life including health monitoring, digital agriculture, and wildfire monitoring [1–3]. These devices integrate multiple sensors and processors to obtain relevant data and process them in-situ using ML models. Development of the innovative applications has been primarily enabled by low-cost sensors, ML algorithms, and processor technologies [1, 2].

Despite the potential offered by IoT devices, they face significant challenges in adoption due to limited battery capacities and frequent recharging needs [1]. Energy harvesting (EH) and management have emerged as a promising approach to mitigate battery limitations in wearable and IoT devices [4–6]. Energy management (EM) algorithms in wearable and IoT devices aim to distribute energy in the system such that application quality is maximized. A common goal of EM in wearable and IoT devices is achieving energy neutral operation (ENO), whereby the energy consumed in a given horizon (e.g., a day) is equal to the harvested energy [5, 7]. The goal of EM in ENO is to allocate energy to each sensor in the system such that the application quality is maximized.

EM algorithms on wearable and IoT devices aim to meet the energy needs of all sensors and processors to fulfill application requirements [5, 7, 8]. Applications maintain accuracy or quality of service when sufficient energy is available for all components. However, due to variations in ambient energy, this is not always possible. EM algorithms typically allocate maximum energy to each sensor while ensuring ENO, however, reduced energy allocation can lead to subsampling or lower duty cycles, impacting application quality.

ML models on wearable and IoT applications are typically trained with the assumption that data from all sensors are available during runtime inference [9]. However, this assumption does not hold when sensors have lower duty cycle or are subsampled due to energy limitations. This leads to significant reduction in accuracy of applications. Our evaluations show that duty cycling can lead to 40% reduction in accuracy for the Shoaib et al. dataset [2]. Obtaining high accuracy with variable levels of energy is a challenging problem since it either requires storage of multiple models (higher memory) or detailed characterization of the energy-accuracy trade-off.

This paper proposes a novel approach, referred to as Energy Management with Imputation (EMI), that integrates data imputation with EM to enable high application accuracy with insufficient energy availability. The proposed approach is based on the following *key insights*: **1)** We can redistribute the energy across sensors such that a subset of sensors have required energy to provide data without turning off or subsampling. Since each sensor affects the accuracy differently, EMI first turns on sensors that have highest impact on the accuracy using a look-up table that is obtained during design time. Other sensors in the system are turned off due to energy redistribution. **2)** Then, we can utilize data from available sensors to impute data for sensors that are turned off. Specifically, the imputation process uses information from available sensors to estimate readings for sensors that are not available due to energy limitations. Once imputed, the data are passed to the ML models for application processing (e.g., classification). Key advantage of the proposed EMI approach is that it avoids incomplete sensor data. More importantly, EMI does not use additional energy in a given interval since our approach is based on redistribution, thus making it highly energy efficient and energy neutral.

We validate EMI with six diverse health monitoring datasets [2, 10–14]. For each of these datasets, we first characterize required energy levels for sensors and accuracy drop with missing sensors. Then, we evaluate EMI under various energy availability conditions to obtain accuracy improvements with EMI. Specifically, we utilize solar EH data

D. Hussein, N. Yamin and G. Bhat are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99164 USA e-mail: (dina.hussein@wsu.edu, nuzhat.yamin@wsu.edu, ganapati.bhat@wsu.edu). This work was supported, in part, by the National Science Foundation (NSF) CAREER Award CNS-2238257.

Manuscript received March 31, 2024; revised June 16, 2024; accepted July 14, 2024. This article was presented in the International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS 2024) and appears as part of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD).

from the National Renewable Energy Laboratory (NREL) in Golden, Colorado [15] to simulate EMI under real-world energy conditions. Evaluations with EH data show that EMI consistently achieves 25–55% higher accuracy than either duty cycling or subsampling of sensors. We also evaluate EMI with uniformly distributed random energy allocations since EH data is unable to capture all possible operating conditions. Validation with random energy data shows that EMI is able to achieve 30-35% higher accuracy, on average, when compared to baseline methods. Finally, measurements on Odroid-XU3 [16] and Texas Instruments CC2652R devices show that EMI incurs less than 1 J of energy overhead for every hour of operation. We note that EMI does not draw additional energy from the battery since energy used for redistribution and imputation is derived from the initial EM allocation. In summary, this paper makes the following novel contributions:

- A novel algorithm that combines data imputation with EM in wearable and IoT devices to maximize accuracy when sufficient energy is not available for complete operation. To the best of our knowledge, this is the first approach that combines EM with data imputation,
- Characterization of accuracy when lower duty cycle or subsampling are used to maintain ENO in IoT devices,
- Experimental validation with six diverse datasets with multiple EM and imputation methods showing efficacy of EMI in a wide range of energy conditions.

Rest of the paper is organized as follows. Section II reviews prior work in EM algorithms and imputation for IoT devices. Section III provides background on the IoT system architecture and introduces the EMI problem. We provide details about the proposed EMI algorithm in Section IV, followed by detailed experimental results in Section V. Finally, Section VI concludes the paper with some future research directions.

II. Related Work

Small batteries in wearable and IoT devices have necessitated the integration of EH technologies [6, 7, 17]. EH from ambient sources requires algorithms that optimally manage the energy [5, 8]. Prior EM algorithms [5, 8] proposed in the literature typically allocate energy to maintain ENO. While ENO is useful to improve operating time, it can lower energy budgets for sensors, leading to lower duty cycles or subsampling. Unavailability of data, in turn, can lead to lower application accuracy.

Recent research has proposed approaches to balance energy availability and accuracy by designing multiple design points (DPs) [18, 19]. For instance, the approach in [18] aims to mitigate the accuracy loss by providing multiple DPs with varying energy-accuracy trade-off. Specifically, it chooses a combination of DPs to maximize accuracy under an energy budget. However, designing multiple DPs requires extensive characterization and may require large number of DPs to be effective. Therefore, there is a strong need to develop approaches that achieve ENO while maintaining high accuracy.

Usage of ML models in wearable devices has become popular to handle complex sensor data and applications [1–3]. Popularity of ML models has led to the study of methods for low power and efficient inference [20–22] on mobile and wearable devices. For instance, the approach in [20] analyzes the performance of various components, such as big cores, LITTLE cores, and graphics processors for neural inference. Similarly, the work in [21] proposes high-throughput inference of convolutional neural networks. These approaches are important and complementary to the proposed EMI algorithm since they lower application energy requirements to enable ENO.

Imputation of sensor data is a promising approach to improve reliability of wearable and IoT applications [9, 23–25]. Prior approaches aim to impute sensor data that are missing due to malfunction or user error [9, 26–28]. Inspired by this, we propose to *strategically* turn off sensors to manage energy in the system and then impute them with lower energy. Recent work also uses generative networks to impute missing sensor data [23, 24, 29]. For instance, generative adversarial imputation networks [23, 30] use available data and a mask to impute missing values using a fully connected generator network. Similarly, the work in [24] uses autoencoders to impute data as a function of the available data. Generative networks are typically useful when imputation of raw data is important. At the same time, they incur higher overhead due to large number of parameters.

Prior research has also used statistical methods, such as mean and median of observed samples around the missing data to obtain imputation patterns from training set to fill in missing data [31, 32]. These methods are intuitive and commonly used for missing data imputation due to their low overhead. Clustering-based approaches have also been proposed for data imputation [25]. These approaches first identify clusters in training data and identify patterns that can be used to impute data at runtime. Specifically, these approaches learn the inter-relationship between sensor clusters to infer possible imputation patterns at runtime [25]. We propose to use these imputation algorithms in conjunction with EM approaches to maximize application accuracy. Overall, the key idea of EMI is to strategically turn off sensors at runtime when energy budget is limited and impute them using available data.

III. Background and Problem Setup

A. Wearable and IoT System Architecture

We consider a general IoT system with multiple sensors and a central processor that performs application processing, as shown in Figure 1. The system contains a battery to power sensors and processors. Since the battery alone is not sufficient to provide energy at all times, we utilize light EH to augment the battery. The battery and EH are managed through an EM algorithm that decides energy budget for each sensor to maintain ENO while maximizing application quality. With this setup, EMI uses the following models for application and EM.

1) Application Model: Let us consider that the IoT system includes M sensors to collect application data. We must perform periodic application execution to ensure that events are not missed. Specifically, each sensor data window consists of T samples, resulting in $M \times T$ data matrix for M sensors. Without loss of generality, we assume that applications consist of classification tasks where the model f_θ provides one of A

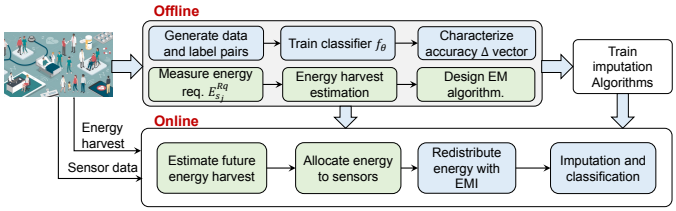

Fig. 1: Overview of the proposed EMI approach.

labels for each window. ML model f_θ is trained using supervised learning methods with offline training data assuming that all sensors are available during inference.

2) Energy Management Model: We consider that a battery with capacity B J is used to power the IoT device. The system also includes one or more EH modalities to augment the battery. Next, we assume that a sensor in the system requires $E^{Rq}_{s_i}(n)$ $(1 \leq i \leq M)$ in an interval n to obtain all required data. Given this, the goal of an EM algorithm is to allocate energy $E^a_{s_i}$ to each sensor s_i to meet its energy requirements while maintaining ENO to ensure long-term sustainability.

Energy from ambient sources follows a daily pattern with day-to-day variations. Therefore, EM algorithms in IoT devices aim to achieve ENO for a one-day horizon [5, 33]. Energy availability also changes during the day due to weather conditions. To this end, we divide a day into N equal intervals for EM decisions. The length of each interval is typically longer than an application window since making decisions with short intervals leads to additional overhead.

B. Problem Setup

The overall goal for wearable and IoT devices is to maximize application accuracy while maintaining ENO using EM algorithms. EM algorithm decisions typically ensure equitable energy to all sensors while maximizing performance. While this is useful, it does not lead to high application accuracy when available energy budgets are not sufficient to meet sensor requirements. Maximizing application accuracy directly in an EM algorithm is challenging because we must characterize a wide range of energy-accuracy trade-offs for all sensors. Even with complete information, we must lower the duty cycle or subsample sensors in energy deficient intervals, leading to accuracy degradation. EMI aims to maximize application accuracy by turning off some sensors and imputing remaining sensors while maintaining ENO. To this end, EMI takes EM decisions $E^a_{s_i}$ as inputs and obtains final energy consumption $E^c_{s_i}$ $(1 \leq i \leq M)$ that maximizes accuracy.

IV. Proposed EMI Approach

A. Overview of EMI

Figure 1 shows an overview of the proposed EMI approach. We start with generation of training data and classification models for each dataset. EMI uses 1-dimensional convolutional neural networks (1D-CNNs) for consistency while noting that any supervised learning classifier is applicable. EMI assesses accuracy loss when a sensor is unable to provide data, substituting zeros to simulate realistic behavior when sensors are off due to energy constraints. This accuracy loss, indicating each sensor's importance, is stored in a look-up table for runtime usage. We also train multiple imputation algorithms to provide data for sensors turned off by EMI, demonstrating the benefits of combining imputation with EM. Next, EM algorithms require knowledge of energy requirement of each sensor to obtain allocations. To this end, we characterize the energy requirement of each sensor and store it in a table.

EMI runtime framework takes the application classifier f_θ, imputation algorithms, accuracy loss table, and sensor energy requirements as inputs. Using the energy requirements and potential EH values, we first obtain the energy allocation $E^a_{s_i}$ $(1 \leq i \leq M)$ for each sensor in a decision interval n. EMI takes these energy allocations and performs redistribution and turns off sensors if energy is insufficient. Then, one of the imputation algorithms is used to obtain data for sensors that are turned off. Finally, f_θ is used to perform classification.

B. Energy Management for ENO

We use an EM formulation that has been used in several prior approaches to achieve ENO [5, 33]. Assuming a battery to store energy and EH of $E^H(n)$ in each interval n, we can formulate the EM problem as follows:

$$\text{max.} \quad \mathcal{O} = \sum_{n=0}^{N-1} \sum_{i=1}^{M} \ln \left(\frac{E^a_{s_i}(n)}{E^{Rq}_{s_i}(n)} * 100 \right) \tag{1}$$

$$\text{s. t.} \quad E_B(n+1) = E_B(n) + \eta E^H(n) - \sum_{i=1}^{M} E^a_{s_i}(n) \tag{2}$$

$$E_B(N) = E_B(0) \geq E_T, E_B(n) \geq E_{min} \quad 0 \leq n \leq N-1 \tag{3}$$

$$E^a_{s_i}(n) \leq E^{Rq}_{s_i}(n), \quad 1 \leq i \leq M \tag{4}$$

Equation 1 aims to allocate up to the requirement for each sensor and uses logarithm of ratio of $E^a_{s_i}(n)$ and $E^{Rq}_{s_i}(n)$ as the objective. Maximizing the sum of the logarithms ensures that all sensors get as much energy as possible to fulfill their requirements. Next, the first constraint describes battery energy dynamics where the battery $E_B(n+1)$ is given by the previous level, EH $E^H(n)$, efficiency η, and actual energy consumption $E^a_{s_i}$ of the sensors. The constraints in Equation 3 specify that battery energy $E_B(N)$ at the end of the horizon is greater than a target level E_T to ensure ENO. We also impose a minimum energy constraint to ensure that battery levels are always greater than E_{min}. Finally, Equation 4 ensures that energy allocated to sensors is less than or equal to the requirements.

Problem Solution: The EM problem in Equations 1–4 is a convex optimization problem with linear and non-linear constraints. Convexity in a problem requires that both objectives and constraints are convex or linear in nature [34]. The constraints in the EM optimization problem are linear in the optimization variable $E^a_{s_i}(n)$. Furthermore, we can show that the objective function is concave by calculating the Hessian. First, the gradient of the objective while omitting known quantities (100 and energy requirements) can be obtained as:

$$\nabla \mathcal{O} = [1/E^a_{s_1}(0), \cdots, 1/E^a_{s_M}(N-1)] \tag{5}$$

Taking derivative of the gradient, we get the Hessian as:

$$\mathcal{H}(\mathcal{O}) = [-1/(E^a_{s_1}(0))^2, \cdots, -1/(E^a_{s_M}(N-1))^2] \quad (6)$$

Each element of the Hessian is negative since energy allocations are non-negative and $E^{Rq}_{s_i}(n)$ is always positive. Consequently, the objective function is strictly concave [34]. Since we maximize the objective, the resulting optimization problem is convex. Indeed, convex optimization for EM in IoT devices has been used in several prior works [35–37], thus showing its promise in obtaining effective EM decisions.

The problem solution gives the energy allocation of each sensor $E^a_{s_i}(n)$. Optimal solution requires exact knowledge of future EH, which is not feasible at runtime. Therefore, EM algorithms typically predict future EH and use heuristics to obtain the solutions [5, 33]. In this work, we utilize four complementary EM algorithms to understand performance of EMI under various conditions, as described in the following.

1) Optimal EM Oracle (EM-O): The EM problem can be solved optimally using a convex solver with knowledge of actual EH values [34]. While this is not practical, knowing the optimal EM decisions in the best case provides an upper bound on the performance of EMI. To this end, we leverage the CVX solver [38] to obtain optimal EM decisions $E^a_{s_i}(n)$ for each interval. Next, actual energy consumption $E^c_{s_i}(n)$ may not match energy allocation $E^a_{s_i}(n)$ due to redistribution from EMI, thus changing optimal decisions for future intervals. To this end, we re-run the optimization algorithm at the beginning of each interval n to adjust future allocations.

2) EM with Iterative Gradient Projection (EM-IGP): Recent research has proposed iterative gradient projection (IGP) algorithms to efficiently solve convex optimization problems [39]. The iterative algorithms can fine tune the number of iterations, step sizes, and tolerances to trade-off accuracy and efficiency. We employ an iterative algorithm that uses the Lagrangian to obtain energy allocations. As a first step, we can write the optimization problem in a compact form as:

$$\text{max.} \quad \mathcal{O} \quad (7)$$

$$\text{s. t.} \quad A(\mathbf{E}^H - \sum_{i=1}^{M} \mathbf{E}^A_{s_i}) - \mathbf{B} \geq 0 \quad (8)$$

$$C(\mathbf{E}^{Rq}_{s_i} - \mathbf{E}^A_{s_i}) \geq 0, \quad 1 \leq i \leq M \quad (9)$$

where $\mathbf{E}^H = [E_B(0) + \eta E^H(0), \eta E^H(1), \cdots, \eta E^H(N-1)]$ captures the energy input to the system. The first interval includes initial battery energy $E_B(0)$, while all other intervals include $\eta E^H(n)$. Similarly, $\mathbf{E}^A_{s_i}$ is the energy allocation vector for each sensor s_i. Vector $\mathbf{B} = [E_{min}, E_{min}, \cdots, E_T]$ captures the minimum energy level at the end of each interval. The last element of $\mathbf{B}$ is populated with E_T to capture the target energy for ENO. $\mathbf{E}^{Rq}_{s_i}$ is a vector used to denote the energy requirements in each interval for sensors. Using this notation, the constraint in Equation 8 captures the battery energy dynamics, minimum energy, and target constraints. $N \times N$ lower triangular matrix A with ones for non-zero values is used to represent the constraints in a compact manner. Similarly, Equation 9 ensures that each energy allocation is below the requirement. C is an $N \times N$ identity matrix to capture all intervals at once. Using the above formulation, we can write the Lagrangian of the problem as:

$$\mathcal{L} = \mathcal{O}(\mathbf{E}^A_{s_i}) + \lambda_1^T(A(\mathbf{E}^H - \sum_{i=1}^{M} \mathbf{E}^A_{s_i}) - \mathbf{B}) + \lambda_2^T C(\mathbf{E}^{Rq}_{s_i} - \mathbf{E}^A_{s_i})$$

where λ_1 and λ_2 are the Lagrange multipliers and $\mathcal{O}(\mathbf{E}^A_{s_i})$ is a vector valued function to capture utility. We can utilize the Karush-Kuhn-Tucker (KKT) conditions to obtain optimality conditions [34] as:

$$\nabla\mathcal{O}(\mathbf{E}^A_{s_i}) - \lambda_1^T A - \lambda_2^T C = 0 \quad (10)$$

$$\lambda_1^T(A(\mathbf{E}^H - \sum_{i=1}^{M} \mathbf{E}^A_{s_i}) - \mathbf{B}) = 0, \lambda_1 \geq \overline{\mathbf{0}} \quad (11)$$

$$\lambda_2^T C(\mathbf{E}^{Rq}_{s_i} - \mathbf{E}^A_{s_i}) = 0 \quad \lambda_2 \geq \overline{\mathbf{0}} \quad (12)$$

where $\overline{\mathbf{0}}$ is a zero vector and $\nabla\mathcal{O}(\mathbf{E}^A_{s_i})$ is the gradient of the utility vector that follows from Equation 1.

We can obtain solution to the EM problem by solving the system of equations in Equation 10–12. Since our goal is to obtain an energy efficient solution, we utilize the IGP algorithm proposed in [39]. Algorithm 1 shows the IGP algorithm used in EMI. The algorithm takes EH values, A, C matrices, tolerance levels, and maximum iterations as inputs. Using the inputs, we first initialize energy allocation and dual variables randomly. Then, we perform the following steps until either the tolerance is met or maximum number of iterations are executed: 1) allocate energy to each sensor for all intervals using inverse of the gradient, 2) update values for dual variables using a feasible descent direction. The algorithm returns energy allocations upon exit. Tolerance, step size, and maximum number of iterations allow us to control the accuracy and runtime overhead. We set the tolerance and step size to 0.00001 and 0.01 in our experiments to achieve convergence in about 100 iterations while consuming at most 30 mJ energy.

Algorithm 1: Iterative gradient projection algorithm

Input: $A, C, \mathbf{E}^H, \mathbf{B}, \mathbf{E}^{Rq}_{s_i}$, tolerance, Step Size δ, and $\text{Iter}_{\max}$
$\mathbf{E}^A_{s_i} \leftarrow rand(N \times M, 1)$
$\lambda_1, \lambda_1^* \leftarrow rand(N \times M, 1)$
$\lambda_2, \lambda_2^* \leftarrow rand(N \times M, 1)$
while $||\lambda_1^* - \lambda_1|| >$ tolerance *and* $||\lambda_2^* - \lambda_2||$ *and* iter $\leq \text{Iter}_{\max}$ **do**
  $\lambda_1 \leftarrow \lambda_1^*$
  $\lambda_2 \leftarrow \lambda_2^*$
  $\mathbf{E}^A_{s_i} \leftarrow \nabla^{-1}\mathcal{O}(\lambda_1^T A + \lambda_2^T C)$ (follows from Equation 10)
  $\lambda_1^* \leftarrow \max\{\lambda_1 + \delta \times (A(\sum_{i=1}^{M} \mathbf{E}^A_{s_i}) - \mathbf{E}^H + \mathbf{B}), \overline{0}\}$
  $\lambda_2^* \leftarrow \max\{\lambda_2 + \delta \times (\mathbf{E}^A_{s_i} - \mathbf{E}^{Rq}_{s_i}), \overline{0}\}$
end
return $\mathbf{E}^A_{s_i}$, λ_1^*, and λ_2^*

Algorithm Complexity: The algorithm must find the energy allocations for M sensors across N intervals, resulting in $N \times M$ variables when optimizing for N intervals. Setting the allocations by calculating the gradient results in complexity of $O(N \times M)$. Next, vector matrix multiplications in each iteration result in complexity of $O((NM)^2)$. Combined with the number of iterations, we obtain a worst case complexity of $O(\text{Iter}_{\max}(NM)^2)$. In practice, we observe that the algorithm converges in a small number of iterations. The iterative

gradient EM algorithm is executed with actual EH values to understand the benefits of EMI with a low-overhead algorithm.

3) EM with Energy Predictions (EM-Pred): The EM-O and EM-IGP algorithms for EM provide useful baselines for EMI, however, they are not realistic for deployment since they use actual values of EH. To this end, we propose to use one of two methods to predict future EH: a) moving average of past three days of EH values and b) ML-based hierarchical prediction from [40]. Using the three day moving average, we can write the expected value of EH $\hat{E}^H(n)$ in an interval n as:

$$\hat{E}^H(n) = \frac{\sum_{i=1}^{3} E^{*H}_{d-i}(n)}{3}, \quad 0 \leq n \leq N \tag{13}$$

where E^{*H} is the actual energy in past days and d denotes the current day. We use average of past three days since it provides accurate predictions with low overhead. Similarly, the hierarchical approach in [40] uses two low-overhead neural networks to predict EH. At first, it predicts energy level (low, medium or high) and then it predicts future energy with an energy prediction network corresponding to the energy level. This approach provides EH estimates with higher accuracy, albeit with a slightly higher overhead. The EH predictions are used in Algorithm 1 to obtain allocations for each sensor.

4) EM with Constraint Relaxation (EM-Heur): The algorithms presented above use iterative approaches that account for the EM constraints. They may present higher overhead for some wearable applications. Therefore, we use the heuristic algorithm presented in [41] as an additional EM algorithm. The heuristic first relaxes the minimum energy constraint to obtain a closed form solution using KKT conditions. The constraint is then applied at runtime by capping the energy consumption if there is a violation. The heuristic provides EM decisions with negligible overhead, albeit with a higher potential of constraint violations. We include it in EMI when overhead of iterative approaches is not desirable.

C. Sensor Data Imputation

One of the critical components of EMI is data imputation at runtime to maximize accuracy. We must have a low-overhead imputation algorithm since the system is operating under limited energy budget. A number of approaches have been used in the literature to impute data [23, 24, 27]. Broadly, we can divide these into constant data, generative, and clustering-based imputation [25, 28]. We propose to utilize one approach from each of these classes in EMI, as detailed next.

1) Average Filling Imputation: One of the simplest methods to impute data is obtaining an average value for each sensor at design time. These average patterns can be saved on the IoT device in a look-up table so that they can be used at runtime. Average-filling provides a low-overhead, yet effective method for imputing data. Average-filling is attractive when reconstruction of raw data is not the main goal for imputation.

2) GAIN Imputation: Generative approaches have been effective in imputation tasks [23, 24]. We utilize the generative adversarial imputation networks (GAIN) method [23] in EMI. GAIN architecture includes a generator and a discriminator to impute data at runtime. The generator takes sensor data with missing values and a missing mask as inputs. The missing mask specifies instances where data are missing, as shown in Figure 2. Using these inputs, the generator obtains data for missing sensors. We employ GAIN in EMI to handle cases where imputation of raw data is required.

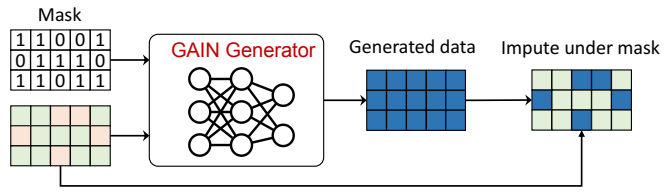

Fig. 2: Overview of imputation using GAIN.

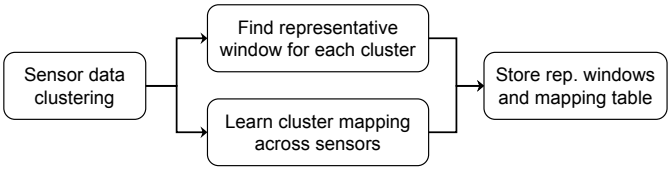

Fig. 3: Overview of clustering-based imputation.

3) Clustering-Based Imputation: Sensor data in wearable and IoT applications often follow repetitive patterns with variations across users and time. We can use unsupervised clustering to group sensor data into clusters, then select a representative data pattern from each cluster for runtime usage. We can also predict clusters for unavailable sensors based on available sensors by learning the relationship between sensors. For instance, consider that we have two sensors in a system and each sensor has three clusters. We may observe that when data from sensor 1 lies in cluster 1, data from sensor 2 lies in cluster 3. If EMI turns off sensor 2 due to limited energy, we can use the cluster information of sensor 1 to predict that data in sensor 2 belongs to cluster 3. The representative data for cluster 3 are then used as the imputation. The imputation process is detailed below and summarized in Figure 3.

Sensor Data Clustering: EMI uses k-means clustering to obtain clusters of sensor data. We choose the number of clusters k through a design space exploration while ensuring that data in each cluster are similar and clusters are not sparse.

Representative Data for Each Cluster: After clustering, EMI selects a representative data pattern for each cluster to use in imputation. The representative data are determined by calculating the average pairwise Euclidean distance between each window in the cluster to all other windows. The window with the smallest average distance is chosen as the representative data, effectively representing an 'average' case for the cluster.

Cluster Mapping and Imputation: We use cluster information of training data to generate a *mapping* table to capture interrelationships across sensors. Each row of the table contains a unique mapping of sensor clusters. For instance, our previous example with two sensors would contain $\{1, 3\}$ as a mapping for sensors one and two, respectively. At runtime, we use the cluster information of sensors turned on by EMI as keys to the mapping table to predict clusters of unavailable sensors. Representative data corresponding to the predicted clusters are used to impute data.

D. Accuracy-Preserving EMI Algorithm

Energy allocations from the EM algorithms are used by the sensors to collect data for use in ML models. The ML models have complete data and achieve maximum accuracy in a given interval n when $E^a_{s_i}(n) = E^{Rq}_{s_i}(n)$ for all sensors. As such, EMI does not perform any optimization when sufficient energy is available and $E^c_{s_i}(n)$ is set to $E^a_{s_i}(n)$. However, if energy allocations from the EM algorithm are not sufficient to power all sensors, EMI redistributes energy to maximize accuracy.

1) EMI Energy Redistribution: Insufficient energy to sensors leads to either lower duty cycle or subsampling to ensure that energy consumption does not exceed the available budget. Lower duty cycle or subsampling typically leads to zeros in the data for unavailable time instances, leading to lower accuracy. Including all possible duty cycles during training is intractable since there is a large space of possible duty cycles. To this end, EMI redistributes available energy to sensors such that sensors that are turned on provide complete data. The design space of EMI is 2^M with M sensors, since each sensor is either turned on or off. The design space becomes larger if EMI chooses one of multiple operating points once a sensor is turned on, such as changing the sampling frequency. However, we consider the smaller design space of turning sensors on or off in this work. We also note that correlation between sensors can be analyzed to disable any redundant sensors. EMI does not focus on correlation between sensors and assumes that datasets provide sensors that are important to the application. Moreover, concept of EMI will be applicable with a reduced set of sensors, since they may also face energy limitations.

Algorithm 2 shows the procedure used by EMI to redistribute energy across sensors. Inputs to the algorithm include EM decisions, energy requirement of all sensors, and a vector representing ML model accuracy loss when a sensor is unavailable. More specifically, an element Δ_i in the $M \times 1$ vector represents the reduction in accuracy for the ML model when data from sensor i are unavailable when compared to the accuracy with complete data. The accuracy vector is obtained during training by setting all readings of sensor i to zero and measuring the accuracy drop Δ_i. Given the inputs, the algorithm first calculates total available energy E^A and initializes an empty sensor status vector $\mathcal{S}$ where all sensors are turned off. We also initialize an energy consumption vector to track assigned energy for each sensor. The algorithm then performs an iterative loop until E^A is distributed to sensors.

Each iteration of the algorithm performs the following steps: 1) Find the sensor with highest accuracy drop for energy assignment, 2) Assign the required $E^{Rq}_{s_j}$ and update the status and energy assignment vectors for s_j, 3) Remove Δ_j from the accuracy vector and update available energy E_A. The algorithm continues until we exhaust all energy or the remaining energy is not sufficient to power a sensor. EMI chooses to not power a sensor partially to save energy for future intervals since missing sensor data will be imputed. EMI also considers situations where the available energy is not sufficient to power even a single sensor. In such cases, EMI allocates all available energy to the sensor with highest accuracy drop since providing partial data and imputing other sensors is better than turning off the system. The algorithm returns the energy allocation and status for each sensor.

Algorithm 2: EMI Energy Redistribution Algorithm

Input: Energy allocation $E^a_{s_i}(n)$, Energy requirements $E^{Rq}_{s_i}(n)$ $(1 \leq i \leq M)$, Accuracy loss vector $[\Delta_1, \ldots, \Delta_M]$
$E^A \leftarrow \sum_{i=0}^{M} E^a_{s_i}(n) - E_{impute} - E_{EM}$
$\mathcal{S} = [0, \ldots, 0]$ and $\mathcal{E} = [0, \ldots, 0]$
while $E^A \geq 0$ **do**
  $s_j \leftarrow \arg\max[\Delta_1, \ldots, \Delta_M]$
  if $E^A < E^{Rq}_{s_j}$ *and* $\mathcal{S}_i = 0 \;\; (1 \leq i \leq M)$ **then**
    $\mathcal{S}_j \leftarrow 1$ and $\mathcal{E}_j \leftarrow E^A$
  else if $E^A < E^{Rq}_{s_j}$ **then**
    return $\mathcal{S}$ and $\mathcal{E}$
  Remove Δ_j corresponding to s_j
  $\mathcal{S}_j \leftarrow 1$ and $\mathcal{E}_j \leftarrow E^{Rq}_{s_j}$
  $E^A \leftarrow E^A - E^{Rq}_{s_j}$
end
return $\mathcal{S}$ and $\mathcal{E}$

2) Runtime EMI Summary: Runtime execution of EMI in each interval consists of the following steps:

1) Determine energy allocation to each sensor $E^a_{s_i}(n)$,
2) Redistribute energy using EMI Algorithm 2,
3) Utilize data from available sensors to impute missing data using the approaches in Section IV-C,
4) Classification using ML model f_θ and imputed data.

V. Experiments and Results

A. Experimental Setup

Wearable and IoT device model: We use Odroid-XU3 [16] as the primary device to implement EMI and measure its overhead. The Odroid-XU3 board integrates four Cortex A15 and four Cortex A7 cores along with sensors to measure power consumption. Odroid-XU3 has been used in several wearable and IoT devices studies [42, 43]. The quad-core ARM Cortex-A7 is similar to the quad-core processor in Qualcomm Snapdragon 400 SoC embedded in state-of-the-art wearable smart watches [44, 45]. Indeed the authors in [44] use Odroid-XU3 board instead of the snapdragon 400 SoC inside an LG watch since the Odroid board has in-built power sensors and allows individual cores to be turned off. Additionally, we use the Texas Instruments (TI) CC2652R microcontroller (MCU) [46] to assess EMI overhead on low-power IoT devices [47]. We assume photovoltaic (PV) cells are available for EH from light sources.

Application Datasets: We use six diverse datasets to validate EMI. Brief descriptions of each dataset are provided below.

Shoaib et al. [2]: The Shoaib dataset provides data for 10 users performing common activities: biking, downstairs, jogging, sitting, standing, upstairs, and walking. Data is collected using five accelerometers placed at distinct positions on the body.

PAMAP2 [10]: PAMAP2 is another popular activity dataset that includes data from nine users with five activities: lying, sitting, walking, running, and jogging. The dataset uses three motion sensors that are placed on the arm, chest, and ankle.

w-HAR [11]: w-HAR is a multi-modal activity dataset that provides data from 22 users for 8 activities: jumping, lying, sitting, standing, walking, up/down stairs, and transition. The

TABLE I: Summary of 1D-CNN parameters for the six datasets

Dataset	Input	Conv1	Conv2	Conv3	FC1	FC2	Accuracy
Shoaib et al.	(5, 600)	(8, 600)	(16, 300)	(32, 150)	64	7	98.9
PAMAP2	(3, 1536)	(8, 1536)	(16, 768)	(32, 384)	64	5	97.6
EMG	(8, 200)	(8, 200)	(16, 100)	(32, 50)	64	4	93.2
w-HAR	(4, 64)	(8, 64)	(16, 32)	(32, 16)	64	5	96.9
SelfRegulationSCP1	(6, 896)	(8, 896)	(16, 448)	(32, 224)	64	2	95.0
WESAD	(5, 200)	(8, 200)	(16, 100)	(32, 50)	64	3	97.4

dataset includes an accelerometer and a stretch sensor [48]. w-HAR provides evaluation of EMI for multi-modal sensors.

SelfRegulationSCP1 (SCP) [12]: SCP provides electroencephalogram (EEG) from six leads for a health application that controls spelling devices for paralyzed patients. The SCP dataset helps in evaluating EMI in a healthcare application.

WESAD [14]: WESAD is a multi-modal dataset using wearable sensors for affect detection. The data are collected from 15 users undergoing three different affective states (neutral, stress, amusement) using five different sensors placed on the chest. The sensors are electrocardiogram (ECG), electrodermal activity (EDA), electromyogram (EMG), respiration (RESP), and body temperature (TEMP). WESAD dataset also provides evaluation of EMI under multi-modal sensor settings.

EMG Physical Action (EMG) [13]: EMG is an activity recognition dataset for participants with aggression during their tasks. The dataset is collected for four different activities (Walking, Kicking, Jumping and Headering) with myoelectrical contractions. The dataset includes recordings from eight EMG sensors placed on the upper arms and legs of the users.

Energy Harvest Data: We employ five years (2016-2020) of per-minute solar irradiation data measured by NREL in Golden, CO as our EH data. The irradiation data is combined with characteristics of a solar cell to obtain the EH values.

We utilize four EM algorithms to obtain energy decisions, as detailed in Section IV-B. We set the number of EM intervals N to 24 to ensure that EM decisions are made every hour. Performing EM with an hourly interval allows the IoT device to adapt to variations in ambient energy. At the same time, we note that EMI is applicable with any decision making interval for EM. The CVXPY library [38] in Python with actual EH values is used to obtain EM decisions for the EM-O method. Similarly, the IGP method or heuristic are used for other EM algorithms. Either the three-day average or hierarchical ML are used for energy estimates.

Application Classifiers and Training: We utilize 1D-CNNs to perform classification for all six datasets. The CNNs consist of three convolutional layers, one max pooling, and two fully connected layers for each dataset with differing input layer sizes. We use the Adam optimizer [49] to train the CNNs. Table I shows the structure of the 1D-CNN for each dataset and corresponding training accuracy with sufficient energy.

Baseline Methods: We utilize two baseline methods commonly used in wearable and IoT devices for comparisons. Brief descriptions of each baseline are provided below:

Sensor subsampling: Energy consumption of a sensor typically increases with sampling frequency [50]. Consequently, we can subsample sensors by reducing the sampling frequency to meet available energy allocation from the EM algorithm.

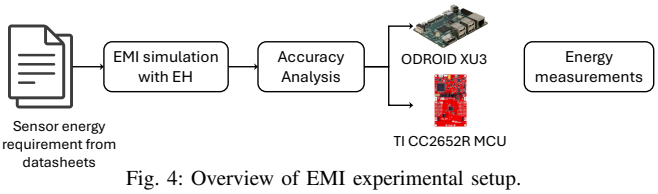

Fig. 4: Overview of EMI experimental setup.

Duty cycle control: Another approach to meet the energy budget is lowering the duty cycle of sensors. Indeed, duty cycle control has been used in prior approaches for ENO [5, 18].

B. EMI Evaluation Methodology

Accurate evaluation of EMI in the real-world requires prototyping with sensors, processors, EH modules and user studies. However, this is time consuming and expensive for multiple applications. Therefore, we use a combination of simulations along with measurements on the Odroid-XU3 and TI CC2652R MCU to evaluate EMI. Figure 4 shows the major steps in the EMI evaluation. We start with the datasets in the previous section to obtain respective classifiers. We also utilize datasheets for sensors in respective datasets to obtain energy requirements based on power consumption and sampling rates.

Using the inputs, we simulate EMI using EH measurements obtained from NREL. Specifically, we stream EH measurements into the simulation setup to make EM decisions and use EMI to re-distribute energy. The EMI decision is applied to obtain available sensor data from the datasets and imputation is used for remaining sensors. We note that the datasets do not provide unique data for five years. Therefore, EMI creates random shuffles of the data to generate additional examples for five years of EH data. Moreover, EH values are scaled up or down depending on the sensor energy requirements to ensure that evaluations are not performed with little to no EH. The EH is scaled such that days with maximum EH match the required energy for the sensors. Finally, the trained classifiers are used to perform classification to evaluate accuracy. Overall, the simulations provide application accuracy when using EMI.

At the end of simulations, we utilize the Odroid-XU3 or TI CC2652R MCU to measure the execution time and energy consumption of EMI computations. Specifically, we implement the EMI functions, including the EM algorithm and energy re-distribution in C. The energy measurements help in understanding the trade-off of using various EM and imputation approaches. We note that the overhead of EMI is accounted during the energy re-distribution process and the system turns off if the available energy is insufficient for EM.

C. Validation with Real-World Energy Data

We start the experimental evaluation with real-world energy data from Golden, CO. We use four EM algorithms for the analysis: EM Oracle (EM-O), Iterative Gradient Projection (EM-IGP), IGP with Energy Predictions (EM-Pred), and the heuristic (EM-Heur). Accuracy results from EMI are then compared with duty cycle control and subsampling baselines.

1) Accuracy with Optimal EM Oracle (EM-O): First, we use the ideal case that uses actual EH values to perform EM using EM-O. While it is infeasible to implement this on the device, EM-O provides an upper bound on EMI performance.

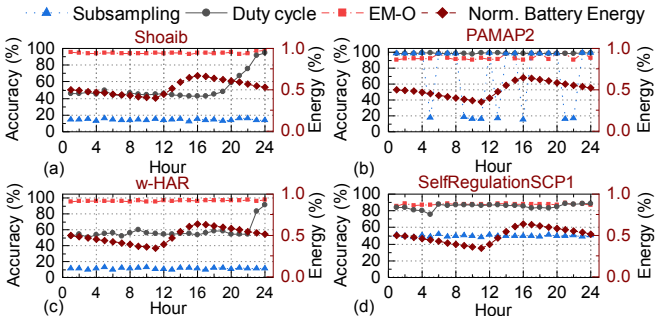

Fig. 5: Comparison of hourly classification accuracy and corresponding battery energy obtained by EM-O-Cluster with baseline methods for (a) Shoaib, (b) PAMAP2, (c) w-HAR, and (d) SelfRegulationSCP1 datasets on Sept. 17, 2017.

EMI Accuracy in a Day: Figure 5 shows the hourly classification accuracy on a day with moderate EH. We show results with EMI-Cluster where we use clustering-based imputation as it is representative of all three imputation algorithms. We compare EMI-Cluster accuracies to the baseline duty cycle control and sub-sampling on the left y-axis. The right y-axis shows the normalized device battery level throughout the day. The EM algorithm is unable to allocate sufficient energy to sensors while maintaining ENO. The lower energy allocations are due to lower levels of ambient energy available to the system. Lower energy allocations result in low accuracy for subsampling or duty cycle control, since they are unable to obtain complete data for classification. In contrast, EMI-Cluster is able to achieve close to 100% accuracy for most intervals in all four datasets shown in the figure. The accuracy of duty cycle control in PAMAP2 is higher than in other datasets because it can tolerate some sensors being turned off. The figure also shows that EMI-Cluster is able to maintain ENO for all four datasets since the battery energy (right y-axis) at the beginning and end of the day are equal.

Monthly Accuracy: EH patterns vary throughout the year due to seasonal changes. Wearable IoT devices must maintain high application accuracy and perform across all seasons. Therefore, we analyze the monthly accuracy achieved by EMI in this section. Figure 6 shows the average accuracy and variance in each month across five years. We see that clustering and average-filling versions of EMI achieve higher accuracy than the baseline subsampling and duty cycle control methods. Furthermore, clustering-based imputation in EMI achieves the highest accuracy when compared to average-filling and GAIN in most of the datasets. Clustering-based imputation achieves higher accuracy compared to average filling because it is able to obtain more imputation patterns as per the available sensor data. In contrast, average-filling uses the same pattern for all missing data in a given dataset. Next, GAIN has lower accuracy due to its higher overhead in imputation and turns off more sensors compared to clustering and average-filling. This highlights the importance of developing low-overhead approaches for imputation. The accuracy achieved by clustering-based and average-filling imputation approaches in EMI is close to the standard accuracy with no missing values (red line) for most of the months. Subsampling and duty cycle control are able to obtain higher accuracy in summer months due to generally higher EH availability. Overall, EMI achieves better performance across all seasons while maintaining ENO.

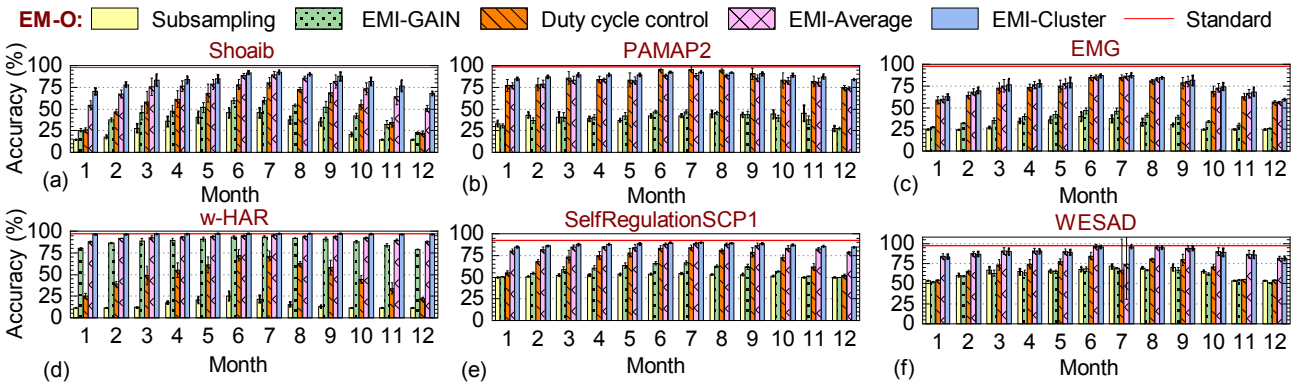

Fig. 6: Accuracy (Mean and standard deviation) comparison of EMI with different imputation algorithms and baselines with EH per month for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

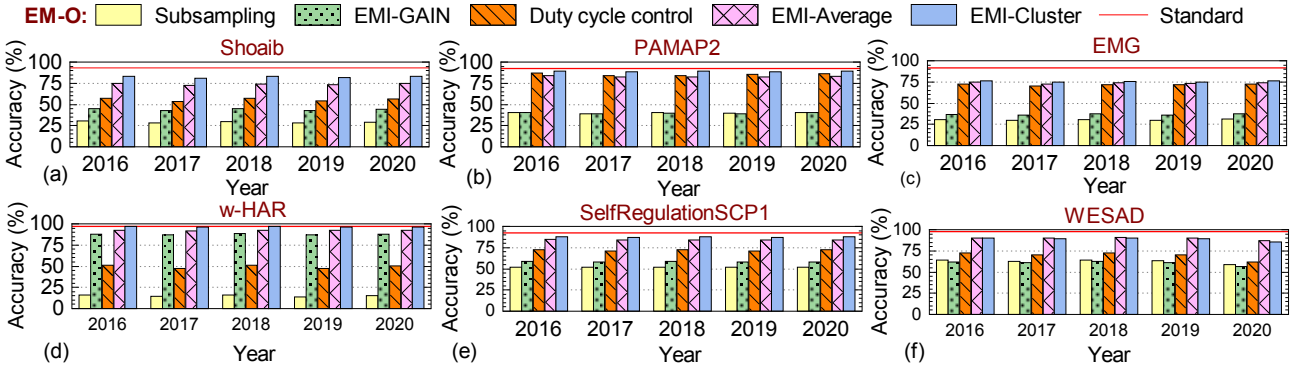

Fig. 7: Accuracy comparison of EMI with different imputation algorithms and baselines with EH per year for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

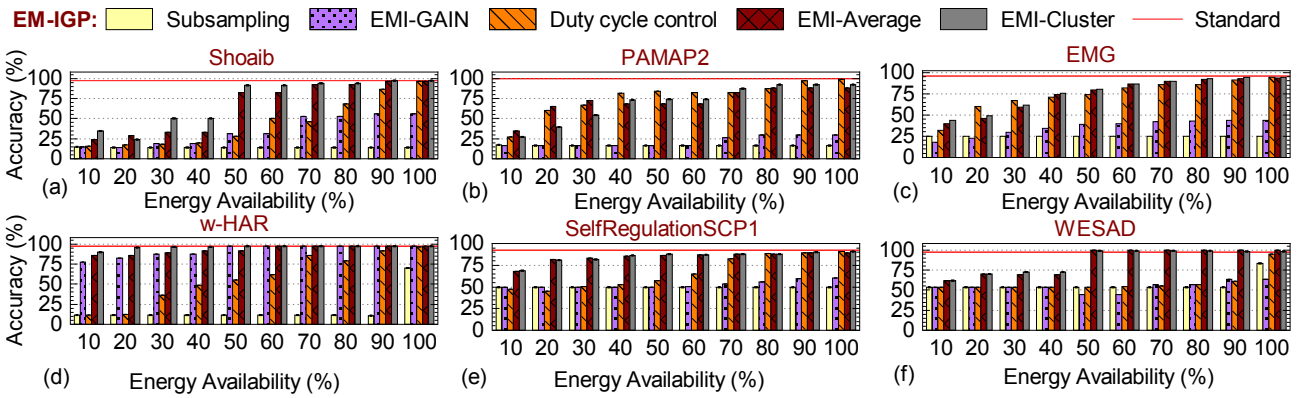

Fig. 8: Accuracy comparison of EMI with EM Iterative Gradient Projection (EM-IGP) and different imputation algorithms and baselines for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

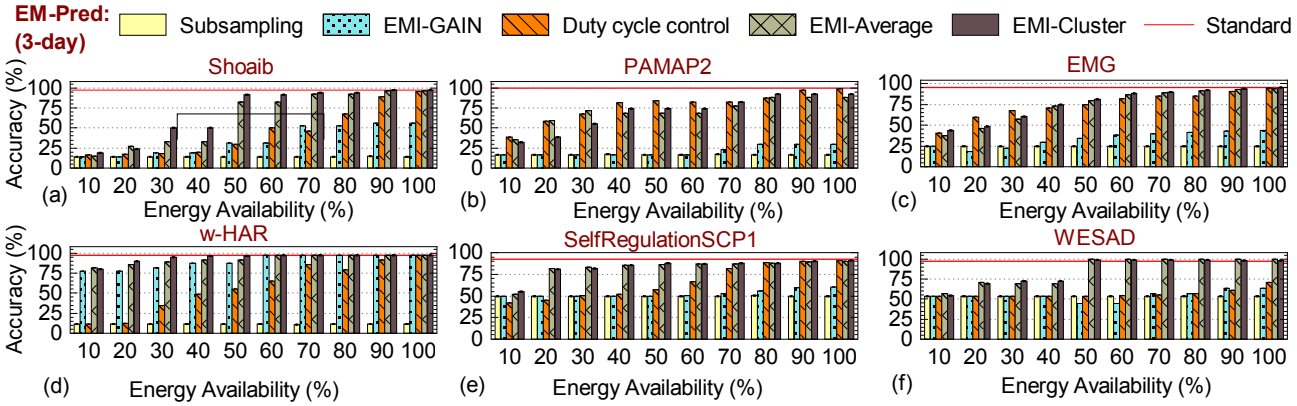

Fig. 9: Accuracy comparison of EMI with EM Energy Predictions (EM-Pred) and different imputation algorithms and baselines for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

Yearly Accuracy: Our goal with yearly accuracy evaluation over five years is to understand EMI performance over a long period of time. Specifically, Figure 7 shows the average accuracy in each year of operation from 2016 to 2020. We see that EMI-Cluster followed by EMI-Average are able to achieve accuracy close to the standard accuracy for all datasets. Duty cycle control in PAMAP2 has higher accuracy due to infrequent need for turning off sensors in the PAMAP2 dataset. This is not an issue for EMI since it uses the default EM decisions when sufficient energy is available. GAIN imputation has lower accuracy due to the higher overhead of executing it at runtime, thus leading to challenges in imputing multiple sensors. In summary, these results show that EMI provides significantly better accuracy when compared to EM baselines.

2) Accuracy with EM Iterative Gradient Projection (EM-IGP): Implementing the CVXPY solver with actual EH values is not feasible due to complexity of solvers and unavailability of future EH values. In this section, we replace the solver with the IGP algorithm. Figure 8 shows the accuracy distribution with varying energy availability to the sensors. For example, the initial set of bars in the graph illustrate the accuracy achieved when available energy falls within the range of 0 to 10% of the required energy. We can see that when we have lower percentages of available energy, EMI-Cluster and EMI-Average are outperforming duty cycle control and subsampling. The baselines have low accuracy in energy-deficient scenarios since they are unable to sample all sensors effectively. Similar to the previous section, we see that GAIN is unable to maintain accuracy due to higher overhead. The accuracy gap between EMI and baseline approaches reduces as energy increases. This is expected since duty cycle control and subsampling are able to sample more data.

3) Accuracy with EM Energy Predictions (EM-Pred): The previous sections' results provide insights into EMI accuracy with ideal EM decisions based on future EH knowledge. However, this is not practical for implementation in the field. Therefore, this section evaluates performance of EMI with EH estimates and the IGP algorithm. The EH estimates are provided using the mean of actual EH values in the past three days or hierarchical ML-based method, respectively. The three-day average EH estimates have a mean absolute error of about 4.9 J, while the ML-based prediction has an error of 1.9 J. Thus, the ML-based prediction provides higher accuracy with slightly higher overhead. Figures 9 and 10 show how the accuracy distribution changes in relation to the amount of available energy with three-day and ML-based estimates, respectively. The bars, ordered by energy availability, show that EMI-Cluster and EMI-Average consistently achieve higher or comparable accuracies to duty cycle control across datasets. The ML-based estimates improve accuracy in low-energy scenarios by minimizing EM errors, especially when energy is less than 50% of requirements. In general, accuracy achieved by EMI approaches are close to the accuracy with no missing data as long as 50% energy is available to the IoT device. Overall, EMI achieves higher accuracies for all energy availability scenarios with both energy prediction methods.

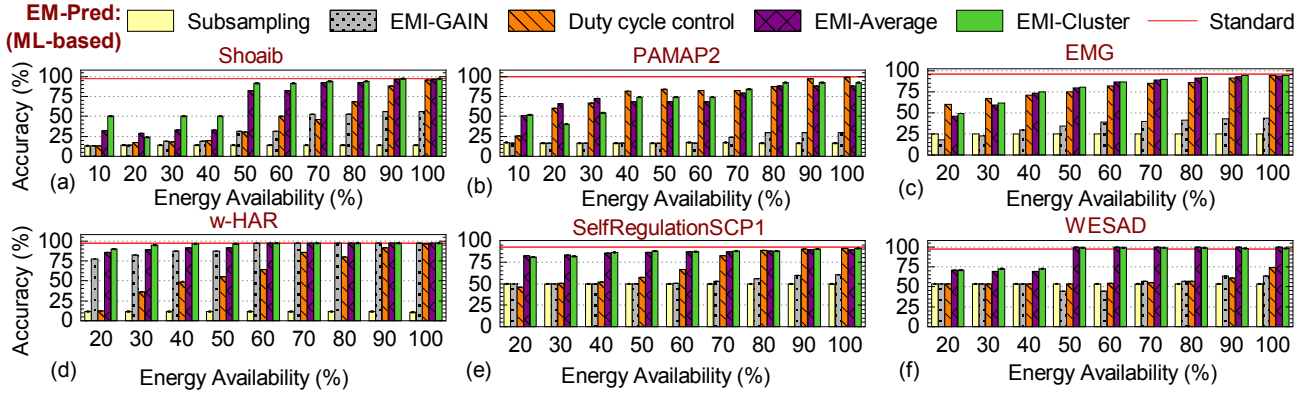

Fig. 10: Accuracy comparison of EMI with ML-based estimation and different imputation algorithms and baselines for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

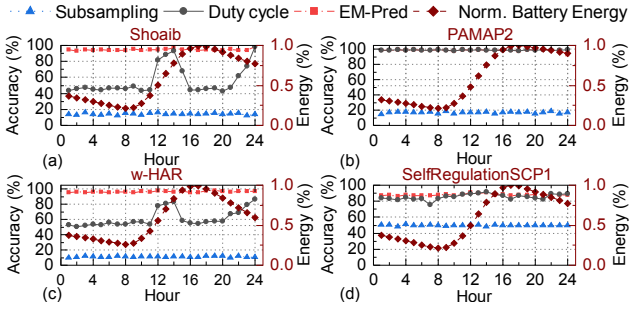

Fig. 11: Comparison of hourly classification accuracy and battery energy obtained by EM-Pred-Cluster with baseline methods for (a) Shoaib, (b) PAMAP2, (c) w-HAR, and (d) SelfRegulationSCP1 datasets on Sept. 17, 2017.

We also compare the performance of EMI-cluster on a specific day for four datasets as shown in Figure 11. The accuracies are obtained when using the EM-Pred with three-day average predictions for energy decisions. Similar to the previous results, we see that EMI-cluster is able to maintain higher accuracies compared to the baselines. The accuracy is lower in early hours of the day for PAMAP2 and Shoaib due to lack of EH. However, EMI-cluster accuracy increases as soon as more energy is available. This shows that EMI adapts to changes in EH patterns throughout the day. In summary, this evaluation shows that EMI achieves high application accuracy in real-world settings with estimates of future EH.

4) Accuracy with EM with Constraint Relaxation (EM-Heur): Finally, we obtain the accuracy with the EM-Heur algorithm using ML-based EH predictions. Figure 12 shows that EMI-Heur has performance within 1% of EM-Pred. This is an important result because it shows that turning off sensors and imputing them has high potential even when EM decisions may not be optimal. In summary, EM-Heur coupled with ML-based EH predictions, and cluster-based imputation offers the best trade-off for accuracy and overhead.

D. EMI with Random Energy Allocations

Evaluations with real-world EH may not expose EMI to all possible energy allocations across the spectrum of no energy to sufficient energy to meet sensor requirements. To this end, we generate random energy allocations over 1000 days for each sensor using a uniform distribution with limits of 0 and $E^{Rq}_{s_j}$, respectively. Figure 13 shows the accuracy distribution as a function of the available energy. As expected, we see that the accuracy gap between the baselines and EMI reduces as the available energy increases. Notably, EMI-Cluster and EMI-Average maintain close to 100% accuracy regardless of the energy budget. Shoaib dataset has lower accuracy when energy is below 50% of the requirement due to challenges in imputation. Overall, this analysis shows that EMI is able to seamlessly handle all energy allocation scenarios.

E. EMI Implementation Trade-Offs

EMI can be implemented with any of the EM and imputation approaches. Each combination offers a unique trade-off in terms of accuracy and overhead. For instance, IGP with small step size and tolerance values provides more accurate EM decisions with higher computations. Similarly, using GAIN for imputation provides raw data with higher energy cost. This trade-off must be managed for each application by selecting algorithms that meet accuracy and energy requirements. Overall, using IGP or EM-Heur with clustering-based imputation offers a balanced trade-off without incurring high overhead.

F. Implementation Overhead

The overhead of EMI with clustering-based imputation consists of energy and latency of imputation, and memory for storing representative data. We first implement EMI on the Odroid-XU3 board to measure energy and memory overheads. Specifically, we measure the power every 12 ms using on-board current sensors. Our measurements show that each cluster imputation consumes about 0.05 mJ to 1 mJ of energy, which results in less than 50 mJ to 1 J energy cost per hour. The overhead for average filling is similar since it uses a single imputation for entire dataset. EMI consumes on average 372 mJ per imputation when using the generative GAIN approach. At the same time, we note that imputation does not consume additional energy from the battery and uses part of the energy allocated to sensors. This allows us to maintain ENO under all energy conditions.

We also measure energy overhead of the imputation algorithms on the TI CC2652R MCU. The measurements show that the clustering-based imputation consumes less than 0.1 mJ per imputation for most datasets. However, the TI-CC2652R board has constraints on memory storage, which impacts its ability to handle larger datasets or GAIN.

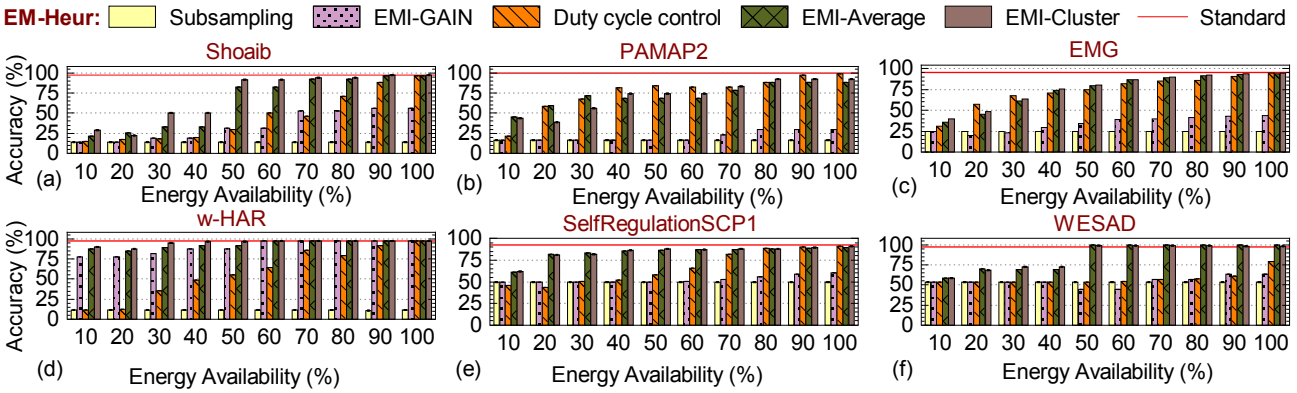

Fig. 12: Accuracy comparison of EMI with EM-Heur and different imputation algorithms and baselines for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

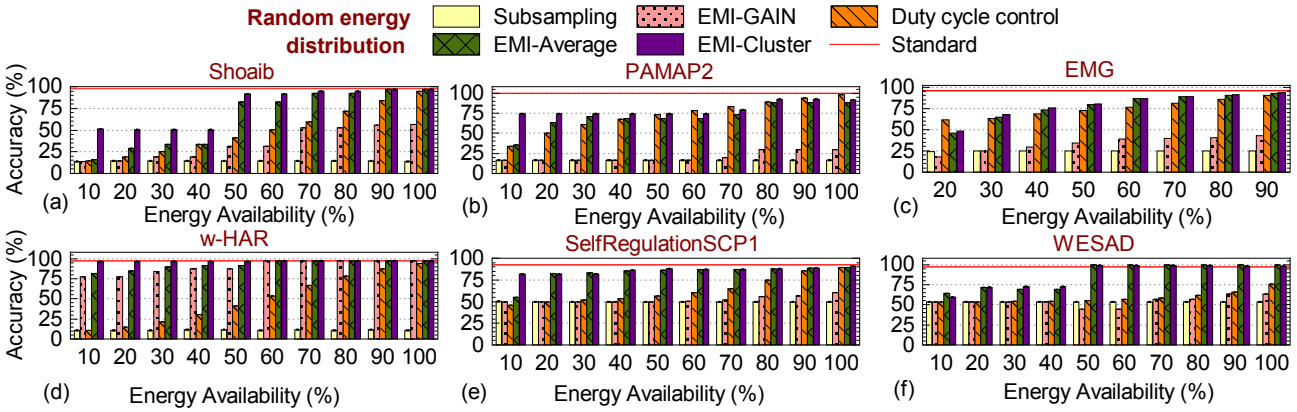

Fig. 13: Accuracy comparison of EMI with different imputation algorithms and baselines with random energy distribution for (a) Shoaib, (b) PAMAP2, (c) EMGPhysical, (d) w-HAR, (e) SelfRegulationSCP1 and (f) WESAD datasets.

Next, each EM decision takes about 10–30 mJ of energy every hour on the Odroid-XU3 and TI CC2652R boards. The EM overhead is obtained by running each EM algorithm multiple times on the board and taking the average. The reported EM overhead is minimal when compared to typical EH value of 15 J. The energy ratios reported in this section use sensor energy requirements from datasheets, EH data from NREL, and overhead measurements from the board. The EM energy is not included in the EMI overhead since the wearable device has to perform EM even when EMI is not used.

Memory overhead for storing representative data in clustering-based imputation is less than 220 KB for all datasets, which is negligible compared to memory available in modern wearable and IoT devices. The memory overhead for GAIN imputation is at most 105 MB to store parameters of the generator. In summary, EMI provides significant accuracy improvements with negligible overhead.

VI. Conclusion

Wearable and IoT devices have the potential to transform multiple facets of human life. However, they suffer from limited operating time due to small battery sizes. Energy harvesting and management has emerged as the most promising method to augment limited battery sizes. EM approaches typically aim to achieve ENO, which may lead to duty cycling or subsampling for sensors, thus reducing application accuracy. This paper proposed the novel EMI approach that smartly redistributes energy and turns on a subset of sensors. Then, EMI imputes sensors that are turned off so that accuracy is not affected. Experiments on six diverse datasets show that EMI achieves 25-55% accuracy improvement while maintaining ENO for wearable and IoT devices. Our future work will focus on analyzing correlations across sensors in EMI to achieve further energy savings by turning off redundant sensors. Other directions for future work include scalability improvements in large scale networks with distributed computing.

References

[1] A. J. Espay *et al.*, "Technology in Parkinson's Disease: Challenges and Opportunities," *Movt. Disorders*, vol. 31, no. 9, pp. 1272–1282, 2016.
[2] M. Shoaib *et al.*, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.
[3] D. Vasisht *et al.*, "FarmBeats: An IoT Platform for Data-Driven Agriculture," in *USENIX NSDI*, 2017, pp. 515–529.
[4] A. Valenzuela, "Energy Harvesting for No-Power Embedded Systems," 2008, https://bit.ly/3fnA6Vm, accessed 3/28/2024.
[5] A. Kansal *et al.*, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embedd. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.
[6] S. Nguyen and R. Amirtharajah, "A Hybrid RF and Vibration Energy Harvester for Wearable Devices," in *Applied Power Electron. Conf.*, 2018, pp. 1060–1064.
[7] M. Odema *et al.*, "Energy-Aware Design Methodology for Myocardial Infarction Detection on Low-Power Wearable Devices," in *ASPDAC*, 2021, pp. 621–626.
[8] M. Geisler *et al.*, "Human-Motion Energy Harvester for Autonomous Body Area Sensors," *Smart Materials and Structures*, vol. 557, no. 1, p. 012024, 2017.
[9] T. Hossain *et al.*, "A Method for Sensor-Based Activity Recognition in Missing Data Scenario," *Sensors*, vol. 20, no. 14, p. 3811, 2020.
[10] A. Reiss and D. Stricker, "Introducing a New Benchmarked Dataset for Activity Monitoring," in *ISWC*, 2012, pp. 108–109.

[11] G. Bhat *et al.*, "w-HAR: An Activity Recognition Dataset and Framework using Low-Power Wearable Devices," *Sensors*, vol. 20, no. 18, p. 5356, 2020.

[12] N. Birbaumer *et al.*, "A Brain-Controlled Spelling Device for the Completely Paralyzed," *Nature*, pp. 297–298, 2001.

[13] T. Theodoridis, "EMG Physical Action Data Set," UCI Machine Learning Repository, 2011, DOI: https://doi.org/10.24432/C53W49.

[14] P. Schmidt *et al.*, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection," in *Proc. Int. Conf. on Multimodal Interaction*, 2018, pp. 400–408.

[15] A. Andreas and T. Stoffel, "NREL Solar Radiation Research Laboratory (SRRL): Baseline Measurement System (BMS); Golden, Colorado (Data); NREL Report No. DA-5500-56488," 1981, accessed 11/20/2023.

[16] Hardkernel, "ODROID-XU3," https://www.hardkernel.com/shop/odroid-xu3/ Accessed 11/20/2023, 2014.

[17] A. Nozariasbmarz *et al.*, "Review of Wearable Thermoelectric Energy Harvesting: From Body Temperature to Electronic Systems," *Applied Energy*, vol. 258, p. 114069, 2020.

[18] G. Bhat *et al.*, "REAP: Runtime Energy-Accuracy Optimization for Energy Harvesting IoT Devices," in *Proc. Des. Autom. Conf.*, 2019, pp. 1–6.

[19] W. Jung and H. G. Lee, "Energy–Accuracy Aware Finger Gesture Recognition for Wearable IoT Devices," *Sensors*, vol. 22, no. 13, p. 4801, 2022.

[20] S. Wang *et al.*, "Neural Network Inference on Mobile SoCs," *IEEE Design & Test*, vol. 37, no. 5, pp. 50–57, 2020.

[21] ——, "High-Throughput CNN Inference on Embedded ARM Big.LITTLE Multicore Processors," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2254–2267, 2019.

[22] M. M. H. Shuvo *et al.*, "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," *Proc. IEEE*, vol. 111, no. 1, pp. 42–91, 2022.

[23] J. Yoon *et al.*, "GAIN: Missing Data Imputation Using Generative Adversarial Nets," in *ICML*, 2018, pp. 5689–5698.

[24] S. Talukder *et al.*, "Deep Neural Imputation: A Framework for Recovering Incomplete Brain Recordings," *arXiv:2206.08094*, 2022.

[25] L. Zhao *et al.*, "Local Similarity Imputation Based on Fast Clustering for Incomplete Data in Cyber-Physical Systems," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1610–1620, 2016.

[26] S. Liu *et al.*, "Handling Missing Sensors in Topology-Aware IoT Applications with Gated Graph Neural Network," *Proc. Interactive, Mobile, Wearable and Ubiquitous Tech.*, vol. 4, no. 3, pp. 1–31, 2020.

[27] D. Hussein *et al.*, "Energy-Efficient Missing Data Recovery in Wearable Devices: A Novel Search-Based Approach," in *2023 IEEE/ACM Int. Symp. on Low Power Electron. and Des. (ISLPED)*, 2023, pp. 1–6.

[28] D. Hussein and G. Bhat, "CIM: A Novel Clustering-based Energy-Efficient Data Imputation Method for Human Activity Recognition," *ACM Trans. on Embedd. Comput. Syst.*, vol. 22, no. 5s, pp. 1–26, 2023.

[29] ——, "SensorGAN: A Novel Data Recovery Approach for Wearable Human Activity Recognition," *ACM Trans. on Embedd. Comput. Syst.*, vol. 23, no. 3, pp. 1–28, 2024.

[30] D. Hussein *et al.*, "Robust Human Activity Recognition Using Generative Adversarial Imputation Networks," in *2022 Des., Automat. & Test in Europe Conf. & Exhib. (DATE)*, 2022, pp. 84–87.

[31] D. W. Ton *et al.*, *Handbook of Statistical Data Editing and Imputation*. John Wiley & Sons, 2011, vol. 563.

[32] I. M. Pires *et al.*, "Improving Human Activity Monitoring by Imputation of Missing Sensory Data: Experimental Study," *Future Internet*, vol. 12, no. 9, p. 155, 2020.

[33] T. Basaklar *et al.*, "tinyMAN: Lightweight Energy Manager using Reinforcement Learning for Energy Harvesting Wearable IoT Devices," in *tinyML Research Symposium*, 2022, pp. 1–7.

[34] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[35] T. Chen and G. B. Giannakis, "Bandit Convex Optimization for Scalable and Dynamic IoT Management," *IEEE Internet Things J*, vol. 6, no. 1, pp. 1276–1286, 2018.

[36] S. Rahim *et al.*, "A Convex Optimization Based Decentralized Real-Time Energy Management Model with the Optimal Integration of Microgrid in Smart Grid," *J. of Cleaner Prod.*, 2019.

[37] N.-T. Nguyen *et al.*, "Energy Management and Time Scheduling for Heterogeneous IoT Wireless-Powered Backscatter Networks," in *Proc. Int. Conf. Commun.*, 2019, pp. 1–6.

[38] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[39] N. Karakoç *et al.*, "Multi-Layer Decomposition of Network Utility Maximization Problems," *IEEE/ACM Trans. on Netw.*, vol. 28, no. 5, pp. 2077–2091, 2020.

[40] N. Yamin, , and G. Bhat, "Online Solar Energy Prediction for Energy-Harvesting Internet of Things Devices," in *Proc. ISLPED*, 2021, pp. 1–6.

[41] G. Bhat *et al.*, "Near-Optimal Energy Allocation for Self-Powered Wearable Systems," in *Proc. Int. Conf. on Comput.-Aided Design*, 2017, pp. 368–375.

[42] K. Y. Leong and S. M. Lim, "SurDis: A Surface Discontinuity Dataset for Wearable Technology to Assist Blind Navigation in Urban Environments," in *Adv. in Neural Info. Proc. Syst.*, vol. 35, 2022, pp. 24 117–24 129.

[43] M. D. Grammatikakis and M. Androulakis, "Comparison of a Medical-Grade and an Open ECG Biosensor Using a Soft Real-Time m-Health Platform," in *Int. Conf. on Apps. in Electronics Pervading Industry, Env. and Soc.* Springer, 2021, pp. 212–220.

[44] C. Tan *et al.*, "Locus: Low-Power Customizable Many-Core Architecture for Wearables," *ACM Trans. on Embedd. Comput. Syst.*, vol. 17, no. 1, pp. 1–26, 2017.

[45] LG, "LG Smart Watch Model," accessed 16 June 2024. [Online]. Available: https://www.lg.com/us/smart-watches/lg-W150-lg-watch-urbane

[46] Texas Instruments Inc., "CC2652R Microcontroller," [Online] https://www.ti.com/product/CC2652R, accessed 1 June 2024, 2018.

[47] T. Zhang *et al.*, "A Survey on Industrial Internet of Things (IIoT) Testbeds for Connectivity Research," *arXiv preprint arXiv*, 2024.

[48] B. O'Brien *et al.*, "Stretch Sensors for Human Body Motion," in *Electroactive Polymer Actuators and Devices*, vol. 9056, 2014, p. 905618.

[49] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR (Poster)*, 2015.

[50] N. Anish *et al.*, "Sensor-Classifier Co-Optimization for Wearable Human Activity Recognition Applications," in *Proc. Int. Conf. Embedd. Soft.Syst.*, 2019, pp. 1–4.