Energy-Efficient Missing Data Imputation in Wearable Health Applications: A Classifier-aware Statistical Approach

Dina Hussein, **Taha Belkhouja**, **Ganapati Bhat** and **Janardhan Rao Doppa**
Washington State University
{dina.hussein, taha.belkhouja, ganapati.bhat, jana.doppa}@wsu.edu

Abstract

Wearable devices are being increasingly used in high-impact health applications including vital sign monitoring, rehabilitation, and movement disorders. Wearable health monitoring can aid in the United Nations social development goal of healthy lives by enabling early warning, risk reduction, and management of health risks. Health tasks on wearable devices employ multiple sensors to collect relevant parameters of user's health and make decisions using machine learning (ML) algorithms. The ML algorithms assume that data from all sensors are available for the health monitoring tasks. However, the applications may encounter missing or incomplete data due to user error, energy limitations, or sensor malfunction. Missing data results in significant loss of accuracy and quality of service. This paper presents a novel Classifier-Aware iMputation (CAM) approach to impute missing data such that classifier accuracy for health tasks is not affected. Specifically, CAM employs unsupervised clustering followed by a principled search algorithm to uncover imputation patterns that maintain high accuracy. Evaluations on seven diverse health tasks show that CAM achieves accuracy within 5% of the baseline with no missing data when one sensor is missing. CAM also achieves significantly higher accuracy compared to generative approaches with negligible energy overhead, making it suitable for wide range of wearable applications.

1 Introduction

Ensuring healthy lives and promoting well-being for all ages is one of the key goals of United Nations (UN) social development goals (SDG) [Sachs *et al.*, 2022]. Achieving this goal will require development of low-cost and reliable devices that provide early warning, risk reduction, and management of health risks. This is especially true for rural and underdeveloped communities with limited access to specialists and diagnostic facilities. For instance, at least one sixth of the population in sub-Saharan Africa lives at least two hours away from a public hospital [Falchetta *et al.*, 2020]. These challenges highlight the need for low-cost and reliable platforms for continuous health assessment.

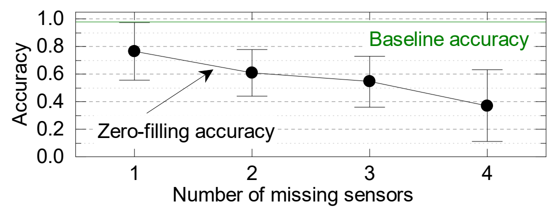

Figure 1: Classification accuracy (mean and standard deviation) of a deployed ML classifier for activity recognition task as a function of k missing sensors.

Wearable internet of things (IoT) devices are emerging as a transformative technology to tackle challenging healthcare applications [Espay *et al.*, 2016; Lara and Labrador, 2012; Maetzler *et al.*, 2013]. Wearable devices are able to continuously monitor parameters of interest in a healthcare task and improve patient outcomes while reducing clinical visits [Espay *et al.*, 2016; Bhat *et al.*, 2019]. Recent healthcare applications employ multiple sensors to improve data fidelity and accuracy. While multiple sensors help in improving the accuracy for complex tasks, they are also susceptible to errors due to missing sensor data. Specifically, data from one or more sensors could be missing due to energy limitations, user error, sensor malfunction, or other challenges [Liu *et al.*, 2020; Hossain *et al.*, 2020]. Missing data can be more severe in under-served communities with intermittent access to power or communication technologies. For instance, users may be able to recharge only a part of wearable sensors or operate them for part of a day due to energy availability constraints [Alam and Ben Hamida, 2014]. Data missingness can lead to significant drop in application performance, as shown for an activity recognition task in Figure 1. Consequently, there is a strong need for principled approaches that handle missing data while maintaining application accuracy.

Several approaches have been proposed to handle missing data [De Waal *et al.*, 2011; Guo *et al.*, 2019; Yoon *et al.*, 2018; Pires *et al.*, 2020]. One class of approaches train multiple classifiers to handle missing data scenarios. For instance, in a task with M sensors, we need $2^M - 2$ classifiers. This solution is not suitable for energy-constrained wearable devices used in health applications due to overhead of context switching during deployment. Moreover, accuracy with a classifier that uses a *subset* of sensors could be lower than using all sensors. The second class of imputation approaches use generative networks. These approaches are inspired by the success of generative networks in image and natural language processing [Creswell *et al.*, 2018; Xu *et al.*, 2018]. Application of generative networks to sensor data imputation suffers from two key limitations: **1)** they incur high memory and computational overhead, which is not suitable for energy-constrained wearables; and **2)** accuracy of imputation and health applications drops significantly when more than one sensor is missing, thus defeating the purpose of imputation. Therefore, there is an immediate need for approaches that impute data with low overhead (energy, execution time, and memory) and maintain high accuracy.

This paper presents a novel *Classifier-aware Imputation (CAM)* approach to impute missing data in wearable health applications while maintaining accuracy. CAM is based on the key insight that we do not need exact imputation of missing sensor data if we can maintain high accuracy for the health task. At the same time, sensor data may follow several repetitive patterns with some variations as a function of activities or demographics. Using a single imputation pattern for multiple distinct sensor data patterns may lower predictive accuracy. Therefore, we propose to utilize unsupervised clustering to obtain distinct patterns of sensor data and obtain an imputation pattern for each combination of clusters. For instance, assume that we have two clusters for each sensor in an application with three sensors. CAM maintains imputation patterns for four combinations of clusters across sensors two and three, if sensor one is missing. Multiple patterns allow CAM to customize imputation to changes in sensor data.

CAM achieves the desired goal of accurate imputations by searching the space of sensor data to obtain imputation patterns that represent an 'average' case of sensor observations. The imputation pattern is chosen such that the overall accuracy of the health task is maximized when using the pattern. The search algorithm in CAM is run for each cluster combination in 2^M-2 missing data scenarios to obtain a table of imputation patterns. Using this table for imputation instead generative networks aids in enabling energy-efficient imputation since the health application only needs to read from the table at runtime. Overall, CAM enables accurate health applications in the presence of missing data with low overhead.

We validate the proposed CAM approach on seven diverse health applications including human activity recognition, gesture monitoring, an assistive device for paralyzed patients, and affect detection [Shoaib *et al.*, 2014; Reiss and Stricker, 2012; Theodoridis, 2011; Wilhelm *et al.*, 2015; Birbaumer *et al.*, 2001; Villar *et al.*, 2016; Schmidt *et al.*, 2018]. We enumerate all possible scenarios of missing sensor data for each application and utilize CAM to obtain imputation patterns. Then, we test the accuracy of the machine learning (ML) classifier with imputed data from CAM and compare it against the accuracy with no missing data (upper bound). Our experiments show that CAM achieves accuracy within 10% of the upper-bound for up to two missing sensors. Moreover, comparisons against a generative approach show that CAM enables more than 15% higher accuracy for health tasks with lower computational cost. Measurements on an embedded device show that CAM imputes data with less than 10 mJ energy per imputation, showing significant promise towards real-world deployment. We are currently in the process of deploying CAM in gesture and activity recognition application in collaboration with the WSU School of Medicine. These deployments take us closer to achieving UN's goals.

Contributions. We make the following contributions:

- Characterization of adverse effects on health applications when data from one or more sensors is missing.
- Novel and energy-efficient approach, referred to as CAM, to obtain imputation patterns that preserve accuracy and quality of service for health tasks.
- Experimental evaluations with seven diverse health tasks to demonstrate that CAM is able to reliably impute missing sensor data with minimal energy and runtime overhead. Code for the CAM algorithm is available publicly at: https://github.com/gmbhat/cam.git.

2 Related Work

Wearable devices enable a number of interesting health applications as evidenced by recent literature [Espay *et al.*, 2016; Mosenia *et al.*, 2017; Limaye and Adegbija, 2018; Bhat *et al.*, 2020]. These applications use multiple sensors to enable more complex tasks or improve accuracy. However, multiple sensors also increase the probability of one or more sensors having missing samples, leading to drop in accuracy. Therefore, there is a great need to develop methods that impute sensor data in an energy-efficient manner [Hussein *et al.*, 2022b].

Several recent approaches in literature tackle the challenge of missing data in multi-sensor applications [Guo *et al.*, 2019; Yoon *et al.*, 2018; Pires *et al.*, 2020; Hussein *et al.*, 2022c; Hussein *et al.*, 2023]. Statistical methods such as mean, median, and kurtosis are used when there are isolated missing samples and reference data are available to evaluate the statistical measures. However, these methods fail when there are longer sequences of missing data. Generative networks have been successfully used for data imputation in recent literature [Yoon *et al.*, 2018; Talukder *et al.*, 2022]. Intuitively, the goal of these approaches is to learn the relationships between different sensors to impute longer sequences of missing data when one or more sensors are missing. However, the deep generative networks incur high overhead due to large number of weights and higher energy overhead to perform imputation. Consequently, they are not suitable for low-power wearable devices. CAM precisely addresses this challenge using a classifier-aware table of imputation patterns to handle missing data with low energy and memory overheads.

3 Background and Problem Setup

This section first provides the background on wearable devices and introduces the missing sensor data problem.

3.1 Wearable Devices Preliminaries

We consider healthcare tasks with multiple sensors monitoring physiological parameters, as shown in Figure 2. The sen-

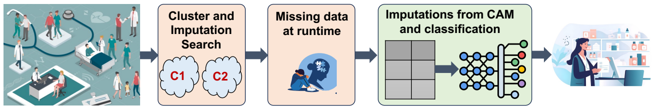

Figure 2: Overview of the proposed CAM approach for health tasks. Missing data in health tasks leads to reduction in quality of service and stress to users. CAM overcomes this by providing a table of imputations that can be used by the application (right part of figure). Imputations by CAM lead to recovery in quality of service.

sor data are used in a number of health applications including movement disorders, vital sign monitoring, and rehabilitation. We use a generic health task as a driver application in this paper since the principles of CAM are generally applicable to any sensor data processing task. The time-series sensor data are streaming in nature and go through the following steps for health assessment in real-time.

Data Segmentation. Health assessment of patients must be performed periodically (e.g., every few seconds or minutes) to track any changes in symptoms. Moreover, many health parameters, such as heart rate, experience variations throughout the day and must be continuously monitored. Therefore, the sensor data must be segmented into either equal or variable length windows for inference with ML classifiers. In general, for a system with M sensors and T samples in each window, we represent the sensor data with $X \in \mathbb{R}^{n\times T}$. A fixed-length segment with T samples is chosen for exposition while noting that each sensor can have variable samples.

Feature Generation and Classification. ML classifiers are commonly used by health applications to infer relevant parameters from sensor data. As such, the sensor data in each window $X \in \mathbb{R}^{n\times T}$ are used as inputs to feature generation and ML classifier blocks. Labeled pairs of sensor data windows and class labels y are used to train the classifiers and streaming sensor data are used with the ML classifier to infer health status at runtime. The ML classifier assumes that data from all sensors are available for classification at runtime.

3.2 Missing Data Challenges

Standard algorithms on wearable devices assume that data from all sensors are available at runtime. However, as noted earlier, sensor data may be unavailable due to energy constraints, communication challenges, or user error. Missing sensor data can either occur in isolated instances or in longer blocks spanning multiple segments. Isolated missing data are easier to handle since we can use available data around the missing samples for imputation. Indeed, a number of statistical approaches have been developed for handling isolated missing samples that occur at random [De Waal *et al.*, 2011; Pires *et al.*, 2020]. Longer missing sequences are more challenging since there are no reference samples for imputation. To this end, the goal of CAM is to impute data for long missing sequences by identifying an average case for missing sensors. This is a challenging problem for healthcare tasks since we must have highly accurate classification while avoiding expensive computations on resource-constrained devices.

3.3 CAM Problem Setup

Consider that a health task uses M sensors to monitor the user's symptoms. We can represent the time-series sensor data with $X \in \mathbb{R}^{n\times T}$ where T is the number of samples in each input window. Each sub-series in X consists of data from a single sensor. That is, the sensor data matrix can be decomposed into M vectors as $[X_1, \cdots, X_i, \cdots, X_M]$ where $X_i \in \mathbb{R}^T$ corresponds to data from i^{th} sensor. Several pairs of sensor data X and labels y are used to train a baseline ML classifier F_θ to perform the health task. Parameters θ denote weights of the ML classifier F_θ. The classifier takes data X with no missingness and predicts corresponding class $\hat{y}$.

The health application may encounter one or more missing sensors during inference in the field. Assume that the set of missing sensors is given by $\mathcal{S}$. Missing data from sensors leads to changes from the expected observations. The data matrix with missing sensors is denoted by $\tilde{X} \in \mathbb{R}^{n\times T}$:

$$\tilde{X} = \begin{cases} 0^T & \text{if } i \in \mathcal{S} \\ X_i & \text{if } i \notin \mathcal{S} \end{cases} \tag{1}$$

The missing data matrix substitutes zeros for missing sensors and uses observed data for sensors that are available. For instance, in a physical activity monitoring application, if one of the sensors is missing, all observations for the missing sensor are substituted with zeros. This change in data results in misclassifications by the health classifier and our goal is to impute data for missing sensors $\mathcal{S}$.

4 Classifier-aware Imputation

In this section, we provide details of the CAM approach for reliable health tasks. We start with an overview of CAM and the two guiding principles behind CAM. Then, we describe the key algorithmic steps involved in CAM.

Overview of CAM and Accuracy-Overhead Tradeoff. Imputation algorithms on wearable devices must trade-off overhead, reconstruction accuracy, and application performance. Generative networks typically achieve higher reconstruction accuracy, however, they incur high overhead due to the use of complex deep neural networks. On the other hand, we can store a *single* imputation pattern for each missing data scenario to significantly lower the overhead. However, this can impact application accuracy since a single pattern may not be able to capture all variations in a dataset. To this end, CAM explores a spectrum of overhead-accuracy trade-offs by using multiple sensor data clusters and storing cluster-specific imputation patterns. Clustering allows CAM to tailor the imputation patterns to variations in sensor data across different classes/activities and users. Moreover, varying the number of clusters allows us to control size of the look-up table since more clusters lead to higher memory requirements. CAM aims to balance the overhead and accuracy by performing design space exploration to uncover highly accurate imputation patterns with minimal number of clusters.

Figure 2 shows the key steps in CAM for health applications. We consider a generic health application that uses multiple sensing modalities, as shown on the left side of the figure. A baseline classifier is trained to process the sensor data for assessing health applications. The sensor data may go missing during real-world usage, which in turns leads to loss of accuracy for users. To handle the missing data, CAM provides a set of imputation patterns $\mathcal{I}$ in a table for each missing data scenario $\mathcal{S}$ and clusters $\mathcal{C}$. The imputation table is used to impute data and perform classification with the trained classifier. Imputation using CAM leads to accurate classification in the presence of missing data and improvement in quality of health applications, as seen on right side of the figure.

Guiding Principles behind CAM. CAM operates on two guiding principles to achieve effective imputation while preserving accuracy of the classifier. **1)** We do not need exact imputation of sensor data as long as the classifier accuracy is preserved. As such, we can search for an 'average' pattern for missing sensors configuration that provides high classification accuracy. **2)** Similarly, the ML classifier for health tasks can be trained to be robust to small deviations in sensor data. Robust training allows for accurate predictions even when the imputation patterns do not match exact sensor data.

4.1 Sensor Clustering and Imputation

Data from sensors for any given health task follows distinct patterns with variations across classes, users, and time. Following this, CAM uses unsupervised clustering to divide data from each sensor into distinct clusters. Specifically, CAM utilizes k-means clustering to obtain a set of clusters $\mathcal{C}$ for the M sensors. The number of clusters k is a hyperparameter that must be chosen such that clusters are well-separated and balanced [Hussein and Bhat, 2023]. Cluster centroids for each sensor are stored on the wearable device for runtime usage.

After obtaining the clusters, CAM obtains a unique imputation pattern for different combinations across sensors. This is based on the insight that there are inter-relationships between sensors. For instance, let us assume that an application has two sensors and CAM obtains two clusters for each sensor. Then, we can have a window with the cluster mapping as (1,2), which indicates that sensors belong to clusters 1 and 2, respectively. We can then learn an imputation pattern I for cluster mapping (1,2), such that if sensor one is missing and sensor two belongs to cluster 2, it can be imputed with I. These unique patterns ensure that imputation patterns are customized to combinations of clusters observed during training.

Algorithm 1 shows the procedure for obtaining imputation patterns as a function of sensor cluster mapping. Inputs to the algorithm include training data, clusters $\mathcal{C}$, classifier F_θ, and number of search iterations. The algorithm first identifies all 2^M-2 missing data combinations. Then, we iterate over each combination of clusters in a missing data scenario to build imputation patterns. In a health task with M sensors and k clusters for each sensor, we have $k^{(M-m)}$ cluster combinations for m missing sensors. CAM builds an imputation pattern for each of these cluster combinations in a missing data scenario. Specifically, all time-series windows corresponding to a cluster combination are passed through a gradient-descent algorithm to search for imputation patterns, as described in the next section. Imputation patterns are stored as a look-up table indexed by missing data scenario and cluster combinations. The table is queried at runtime to impute data as a function of missing sensors and clusters of available sensors.

Algorithm 1: CAM Training Procedure

Input: Training data $\mathcal{D}_{\text{train}}$, Sensor clusters $\mathcal{C}$, health classifier F_θ, MAX_G, maximum iterations for gradient descent; MAX, maximum iterations over all inputs
$\mathbf{C}_{\text{train}} \leftarrow$ Cluster information for each sensor in $\mathcal{D}_{\text{train}}$
$\mathbf{M}_s \leftarrow$ Enumerate $2^M - 2$ missing data scenarios
Initialize empty imputation table $\mathcal{I}$
for $m \in \mathbf{M}_s$ **do**
  Missing $\leftarrow$ Set of missing sensors
  Available $\leftarrow$ Set of available sensors in m
  $\mathbf{C_a} \leftarrow$ Combination of clusters for available sensors
  for $c_a \in \mathbf{C_a}$ **do**
    $\mathcal{X} \leftarrow$ All windows in $\mathcal{D}_{\text{train}}$ with clusters c_a for available sensors
    $\mathcal{I}_m \leftarrow$ Call Algorithm 2 with $\mathcal{X}, m, MAX_G$, and, MAX
    Update $\mathcal{I}$ with imputation $\mathcal{I}_m$, *key:* (m, c_a), *value:* $\mathcal{I}_m$
  end
end
Return: Imputation table $\mathcal{I}$

4.2 Algorithm to Find Imputation Patterns

High-level Overview of Algorithm. ML classifiers generally achieve high accuracy when all data are available for classification. Since we use multiple sensors for health applications, the classifier uses information spread across the M sensors. Loss of information from any one sensor leads to drop in accuracy. Prior studies have also shown that classifiers may rely on a subset of critical sensors to make many classification decisions [Belkhouja and Doppa, 2020]. CAM exploits this behavior by first pushing the classifier to rely more on the available sensors instead of the missing sensors. At the same time, information of missing sensors should not be completely ignored. Therefore, CAM utilizes a classifier-aware search to find the most likely 'average' case pattern for missing sensors to preserve accuracy of health tasks.

Algorithm. The goal of CAM is to obtain imputation patterns $\mathcal{I}_m$ that achieve same classification results as the ideal case with no missing data, where m is the set of missing sensors. We can represent it as a problem of searching for $\mathcal{I}_{i\in m} \in \mathbb{R}^T$ such that:

$$\mathcal{I}_m = \begin{cases} \mathcal{I}_i & \text{if } i \in m \quad \text{and} \quad F_\theta(X) \approx F_\theta(\mathcal{I}_m) \\ X_i & \text{if } i \notin m \end{cases} \tag{2}$$

where $F_\theta(X)$ and $F_\theta(\mathcal{I}_m)$ are the classifier outputs with original and imputed data, respectively. Intuitively, Equation 2 specifies that classifier outputs with imputed data are close to the output with real, observed data. The imputation patterns do not depend on the observed data from sensors. Instead, we search for a suitable 'average' pattern that maintains classifier accuracy. The problem of finding an imputation pattern for a given missing data scenario can be formulated as:

$$\text{Given } m \text{ and } \mathcal{X}: \text{ Find } \mathcal{I}_m \text{ s.t. } \forall \mathcal{X}, F_\theta(\mathcal{X}) \approx F_\theta(\mathcal{I}_m) \tag{3}$$

The search space must be appropriately traversed to ensure that we find effective patterns. To this end, we use the follow-

Algorithm 2: CAM Search Algorithm

Input: Windows $\mathcal{X}$; F_θ, pre-trained classifier; m, missing sensors configuration; MAX_G, maximum iterations for gradient descent; MAX, maximum iterations over all inputs
Output: $\mathcal{I}_m$, imputation pattern
Random initialization of the set $\{\mathcal{I}_{j\in m}\}$
for *i=1*, $\cdots$, MAX **do**
  for *each training example X* **do**
    for i_G=*1*, $\cdots$, MAX_G **do**
      $\mathrm{lg}X = \mathrm{Logits}(F_\theta(X))$
      $\mathrm{lg}I = \mathrm{Logits}(F_\theta(\mathcal{I}_m))$
      Estimate the loss $\mathcal{L}(\mathrm{lg}X, \mathrm{lg}I)$
      Estimate the gradient $\nabla_{\{\mathcal{I}_j\}}\mathcal{L}$ for $j \in m$
      Perform gradient descent and update $\mathcal{I}_m$
    end
  end
end
return Imputation pattern $\mathcal{I}_m$

ing minimization problem to solve Equation 3:

$$\min_{\mathcal{I}_{j\in m}} \mathcal{L}\Big(\mathrm{Logits}(F_\theta(\mathcal{X})), \mathrm{Logits}(F_\theta(\mathcal{I}_m))\Big) \tag{4}$$

where $\mathcal{L}$ is a loss function that compares logits of the classifier with imputed and original data. We interpret the logits as the unnormalized predictions of each class label in the health task. If we ensure that logits are close to the ideal case with no missingness, it means that the accuracy of classification is preserved. The loss function $\mathcal{L}$ can be any differentiable function that evaluates the difference between two logit values. In our implementation, we use mean squared error as the loss function. Overall, classification accuracy with imputed and original signals will be similar when $\mathcal{L} \to 0$.

CAM utilizes a gradient descent-based algorithm to solve the minimization problem of finding imputation patterns. Algorithm 2 shows the key steps of our solution to find imputation patterns $\mathcal{I}_m$. The number of gradient descent steps MAX_G is set to 100 to ensure that the algorithm finds the optimal pattern for each input example in $\mathcal{X}$. Similarly, the total iterations of optimization for the overall algorithm is set to 50 for finding optimal imputation patterns across the entire training data. The algorithm outputs imputation pattern $\mathcal{I}_m$ for a set of missing sensors m. Outputs of the algorithm are used to construct the look-up table that is used at runtime, as described in the previous section.

4.3 Robust Classifiers with CAM

CAM stores a single imputation pattern for each missing sensor configuration in a given health task. Using a single pattern allows CAM to have negligible overhead, however, it also increases the likelihood of classifier errors in a small number of instances. This is because the classifier has not seen the imputation pattern during training. We train robust classifiers [Belkhouja *et al.*, 2023] to overcome this challenge.

We train robust classifiers in CAM following the recent framework of augmented time-series data based on statistical features [Belkhouja and Doppa, 2022; Hussein *et al.*, 2022a]. The key idea is to generate small perturbations on the available training data while preserving the statistical features. We also include the imputation patterns from CAM as additional training data. The classifier F_θ is trained with these additional training examples to obtain a robust classifier F_θ^*.

5 Experiments and Results

This section evaluates the performance of CAM on seven healthcare tasks along multiple dimensions.

5.1 Experimental Setup

Wearable Device Setup. We employ the Odroid-XU3 board [Hardkernel, 2014] for sensor data processing and imputation. CAM utilizes the A7 cores on the Odroid-XU3 board for imputation since they are more energy-efficient. The Odroid board stores imputation patterns for all sensors and uses them as a function of the missing sensors.

Datasets. We employ seven diverse healthcare tasks to validate the efficacy of CAM. For each of these datasets we evaluate all possible missing data scenarios by varying the number of missing sensors from 1 to $M-1$.

Physical Activity Recognition [Shoaib et al.*, 2014]:* Activity recognition in health tasks is crucial since knowing what a patient is doing is important for movement disorders and rehabilitation [Maetzler *et al.*, 2016]. The Shoaib dataset provides accelerometer data for 10 users performing seven activities.

PAMAP2 [Reiss and Stricker, 2012]: PAMAP2 is another activity recognition dataset that provides data from three accelerometers for five activities with nine users.

EMG Physical Action (EMG) [Theodoridis, 2011]: The EMG dataset is another activity recognition dataset for users who may experience aggression during their tasks. The dataset is collected using EMG sensors interfacing four activities (Walking, Kicking, Jumping and Headering) with myoelectrical contractions. The dataset includes recordings from eight sensors placed on the upper arms and legs of the users.

Gesture Recognition [Wilhelm et al.*, 2015]*: Gesture recognition is an important in various healthcare tasks such as assistive devices and rehabilitation. To this end, we utilize the eRing dataset to evaluate CAM in gesture recognition settings. eRing is a smart health dataset that uses a finger ring to capture data along four dimensions using electric field sensing.

SelfRegulationSCP1 (SR-SCP1) [Birbaumer et al.*, 2001]*: Electroencephalograhy (EEG) is commonly used in brain-machine inference tasks. We use the SR-SCP1 dataset to evaluate CAM with EEG recordings. SR-SCP1 includes EEG data from six channels. The data are used in a control system that drives spelling devices for paralyzed patients.

Epilepsy [Villar et al.*, 2016]*: This dataset is collected from six participants using an accelerometer on the dominant wrist while conducting four different activities. The activities are walking, running, sawing, and seizure mimicking.

WESAD [Schmidt et al.*, 2018]*: WESAD is a multi-modal dataset using wearable sensors for affect detection. The data is collected from 15 participants 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). We note that in our experimental settings, we find that TEMP sensor's data is crucial to obtain a stable performance for all baselines. Therefore, we work under the assumption that the TEMP sensor is always available.

Evaluation Metrics. Application accuracy is crucial in healthcare tasks as it is important to maximize true positives for a condition. Therefore, we use accuracy as the primary metric in evaluation of CAM. Furthermore, energy and memory are important considerations due to resource constraints in wearable health applications.

Classifier Representation. CAM uses a classifier that is trained with data from all sensors for each dataset. Specifically, we use a 1-dimensional convolutional neural network (1-D CNN) to perform classification. The 1-D CNN uses two convolution and max-pool layers followed by two fully connected layers with the ReLU activation. All networks are trained with the Adam optimizer [Kingma and Ba, 2015] for 20 epochs. 60% of the data are used for training and 40% of data are used for cross-validation and testing.

CAM Training Overhead. We perform the imputation pattern search on a server with 32 Intel® Xeon® Gold 6226R cores with 192 GB of memory. Since the training is performed offline, it can use large-scale servers to obtain optimal imputation patterns. We note that the training overhead increases with number of sensors and clusters. This overhead can be minimized by utilizing parallel processing since patterns for each combination can be searched independently.

5.2 Baseline Methods for Comparison

We compare the accuracy of CAM against three baselines including a generative approach, zero-filling, and average-filling. The goal of these comparisons is to evaluate both accuracy and overhead when compared to the baselines.

GAIN. Generative approaches have been used for imputation. One specific instance of these approaches is GAIN that imputes data as a function of available observed data. Specifically, the generator in GAIN takes observed and missing data as inputs and outputs a data matrix that contains imputed values in missing instances. The primary disadvantage of GAIN is that accuracy degrades with increase in missing sensors.

Zero Filling. Missing sensor data will be typically filled with zeros when an imputation method is not available. This is because the sensor will not send any data to the processor. We use it as one of the baselines since it is easy to implement.

Average Filling. The missing data can also be substituted by imputation patterns that are determined from prior observations. One instance of this is average-filling, where missing data are substituted with average values from the training data. We use this as the third baseline for comparison.

5.3 Application Accuracy with Imputed Data

Accuracy of healthcare tasks is the primary consideration for CAM. Therefore, we first evaluate accuracy of each application with all four imputation approaches. We also obtain accuracy with no missing data since it provides the upper-bound on classification performance for each application. We also note that a single classifier is used with each application and we do not train new classifiers for each missing data scenario.

Label	**1**	**2**	**3**	**4**	**5**	**6**
1	**40**	0	4	3	3	0
2	1	**49**	0	0	0	0
3	0	0	**46**	3	1	0
4	0	0	3	**44**	3	0
5	3	0	0	15	**32**	0
6	0	0	0	0	0	**50**

Table 1: Confusion matrix for eRing dataset using CAM for missing data imputation

	Lie Down	**S**it	**W**alk	**R**un	**C**ycle
L	**355**	17	0	0	2
S	15	**686**	1	1	13
W	0	6	**453**	0	0
R	0	2	0	**368**	0
C	0	5	0	0	**309**

Table 2: PAMAP2 dataset confusion matrix using CAM for missing data imputation

Figure 3 shows the classification accuracy with CAM and other imputation approaches for all missing data scenarios. Each point on the figure shows the average accuracy and standard deviation with a fixed number of missing sensors. For instance, the point corresponding to one on the x-axis in Figure 3(a) shows accuracy and standard deviation over five cases of one sensor missing in the Shoaib et al. dataset. The default accuracy with no missingness for each task is shown with an orange line. The figures clearly show that CAM achieves higher accuracy than all three baselines. In particular, zero-filling and average-filling have significant accuracy drop for the majority of health tasks. For instance, zero-filling and average-filling show accuracy drops of about 15–20% for activity recognition. In WESAD dataset, we see that average-filling has comparable performance in some of the missing scenario cases when EDA sensor is missing. In these cases, we can utilize average-filling as the imputation pattern and avoid running CAM. At the same time, if any sensor other than EDA is missing, CAM achieves higher accuracy. There is a drop in accuracy with the increasing number of missing sensors which is due to the complexity of retrieving lost information. We note that even GAIN fails to improve the accuracy when a high number of sensors is missing.

CAM is also able to achieve better accuracy compared to the generative GAIN approach. GAIN is able to recover accuracy for up to one sensor, but it is unable to handle more than one missing sensor in-spite of being more complex and using larger number of parameters than CAM. This is a remarkable result for CAM since it is able to achieve higher accuracy compared to GAIN even when entire sequences of data are missing. Finally, Tables 1 and 2 show the confusion matrix of classifications for gesture and activity recognition tasks when one sensor is missing. The matrices show that CAM is able to equally recover all classes for both tasks. Additional confusion matrices are provided in an appendix available at: https://github.com/gmbhat/cam.git.

5.4 Implementation Overhead

The primary advantage of CAM over other baseline approaches is higher accuracy with minimal energy and memory overhead. To demonstrate this, Table 3 shows the memory requirements for CAM and GAIN. As expected, GAIN needs significantly higher memory on the device to store generator parameters. In strong contrast, overhead of CAM is less than 10% of the overhead of GAIN for all datasets except

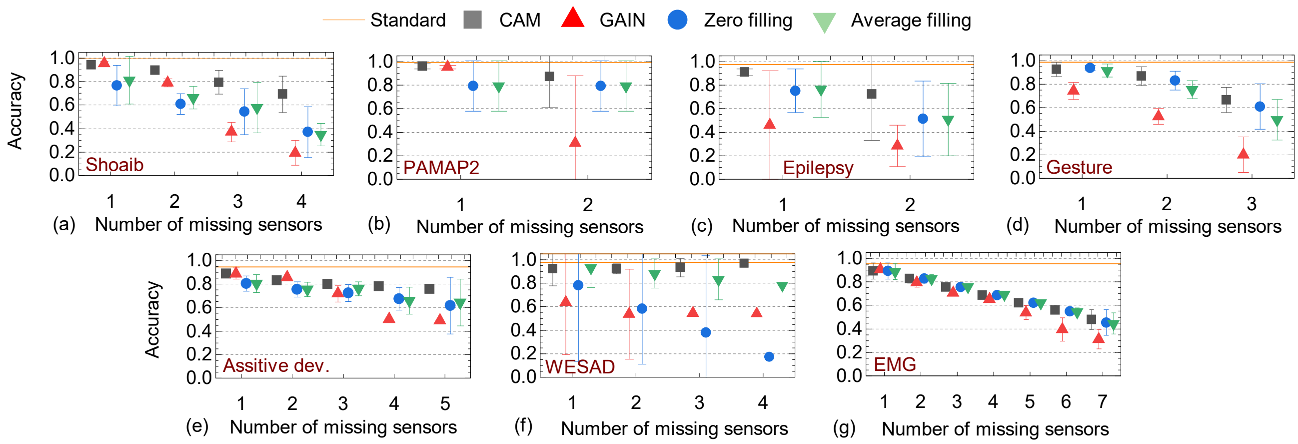

Figure 3: Accuracy (Mean and standard deviation) of ML classifier via different imputation methods on all combinations of missing sensors.

Dataset	CAM Memory (MB)	GAIN Memory (MB)
Shoaib	0.92	43.0
PAMAP2	0.15	102
Epilepsy	0.02	1.84
eRing	0.22	0.33
SR-SCP1	64.2	139
WESAD	0.31	4.82
EMG	0.78	12.3

Table 3: Summary of memory overhead of CAM and GAIN

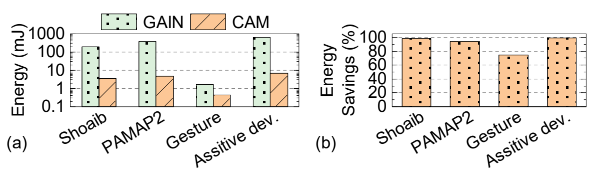

Figure 4: a) Comparison of energy consumption for GAIN and CAM. The y-axis is shown in log scale to represent the large range of values. b) Energy savings achieved by CAM compared to GAIN.

for eRing and SR-SCP1. The overhead is low for both CAM and GAIN for the eRing dataset. SR-SCP1 dataset has higher overhead since we must store more clusters and imputation patterns due to higher variability in data. The overhead is still less than GAIN while achieving higher accuracy.

We also compare the energy consumption of GAIN and CAM for the health tasks. Figure 4(a) shows the energy consumption while Figure 4(b) shows energy savings achieved by CAM compared to GAIN for four datasets, while noting that rest of the datasets show similar behavior. We see that CAM has less than 10 mJ energy for all four tasks while GAIN incurs higher energy. GAIN has almost 1 J energy consumption in the assistive devices task. We see similar behavior with energy savings where CAM achieves close to 100% savings in energy when compared to GAIN. In summary, CAM provides better performance while incurring significantly lower overhead.

6 Path to Deployment

Successful deployment and adoption of this project will lead to better health monitoring outcomes for the public, especially in rural areas and underdeveloped nations. This will be directly in line with UN's social development goal of healthy lives for all ages and communities. The authors of this paper have conducted successful user studies and deployments for gesture recognition, gait, and activity monitoring applications. Following on this experience, the authors are well-prepared to deploy CAM on a wide range of health subjects and patients. CAM is well-suited for deployment on resource-constrained devices in underdeveloped areas due to its low memory and energy overheads.

The authors also have active collaborations with the WSU School of Medicine and neurosurgeons for studies with patients. Through these collaborations, we will conduct studies with patients and evaluate performance of CAM in real-world settings. After these preliminary evaluations, we will work with community leaders across multiple nations to test CAM in the field. Specifically, we will develop smartphone or smartwatch applications so that users can monitor their health remotely. Results from health applications will be relayed to a healthcare dashboard that can be monitored by doctors. In addition to smartphones, we anticipate developing low-cost wearable devices that include necessary sensors and processors without additional features. This will bring the cost down to $10 or less, while having few mW of power consumption. The low-cost and low-power devices will be ideal for deployment in rural areas and underdeveloped nations. We anticipate widespread deployment of CAM in one to two years.

7 Conclusion

Several healthcare tasks are using wearable devices to enable continuous monitoring and to improve patient outcomes. However, multi-sensor wearable devices may suffer from missing data in the real-world, leading to loss in application quality. This paper presented a classifier-aware imputation approach to handle missing data. The proposed CAM approach enables low-overhead imputation by maintaining a table of 'average' case patterns for each missing scenario and cluster combinations. Experiments on a variety of health tasks showed that CAM achieves higher accuracy compared to more expensive baselines. Our immediate future work is to deploy CAM in variety of real-world settings.

Contribution Statement

Dina Hussein and Taha Belkhouja contributed equally to this work.

Acknowledgements

This work was supported in part by NSF CAREER award CNS-2238257 and in part by the AgAID AI Institute for Agriculture Decision Support, supported by the National Science Foundation (NSF) and United States Department of Agriculture (USDA) - National Institute of Food and Agriculture (NIFA) award #2021-67021-35344.

References

[Alam and Ben Hamida, 2014] Muhammad Mahtab Alam and Elyes Ben Hamida. Surveying Wearable Human Assistive Technology for Life and Safety Critical Applications: Standards, Challenges and Opportunities. *Sensors*, 14(5):9153–9209, 2014.

[Belkhouja and Doppa, 2020] Taha Belkhouja and Janardhan Rao Doppa. Analyzing Deep Learning for Time-Series Data Through Adversarial Lens in Mobile and IoT Applications. *IEEE Trans. on Comput.-Aided Design of Integr. Circuits and Syst.*, 39(11):3190–3201, 2020.

[Belkhouja and Doppa, 2022] Taha Belkhouja and Janardhan Rao Doppa. Adversarial Framework with Certified Robustness for Time-Series Domain via Statistical Features. *Journal of Artificial Intelligence Research*, 2022.

[Belkhouja *et al.*, 2023] Taha Belkhouja, Yan Yan, and Janardhan Rao Doppa. Dynamic Time Warping Based Adversarial Framework for Time-Series Domain. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7353–7366, 2023.

[Bhat *et al.*, 2019] Ganapati Bhat, Ranadeep Deb, and Umit Y Ogras. OpenHealth: Open-Source Platform for Wearable Health Monitoring. *IEEE Design & Test*, 36(5):27–34, 2019.

[Bhat *et al.*, 2020] Ganapati Bhat, Nicholas Tran, Holly Shill, and Umit Y Ogras. w-HAR: An Activity Recognition Dataset and Framework using Low-Power Wearable Devices. *Sensors*, 20(18):5356, 2020.

[Birbaumer *et al.*, 2001] N Birbaumer, H Flor, N Ghanayim, T Hinterberger, I Iverson, E Taub, B Kotchoubey, A Kbler, and J Perelmouter. A Brain-Controlled Spelling Device for the Completely Paralyzed. *Nature*, pages 297–298, 2001.

[Creswell *et al.*, 2018] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[De Waal *et al.*, 2011] Ton De Waal, Jeroen Pannekoek, and Sander Scholtus. *Handbook of Statistical Data Editing and Imputation*, volume 563. John Wiley & Sons, 2011.

[Espay *et al.*, 2016] Alberto J Espay, Paolo Bonato, Fatta B Nahab, Walter Maetzler, John M Dean, Jochen Klucken, Bjoern M Eskofier, Aristide Merola, Fay Horak, Anthony E Lang, et al. Technology in Parkinson's Disease: Challenges and Opportunities. *Movt. Disorders*, 31(9):1272–1282, 2016.

[Falchetta *et al.*, 2020] Giacomo Falchetta, Ahmed T Hammad, and Soheil Shayegh. Planning Universal Accessibility to Public Health Care in Sub-Saharan Africa. *Proceedings of the National Academy of Sciences*, 117(50):31760–31769, 2020.

[Guo *et al.*, 2019] Zijian Guo, Yiming Wan, and Hao Ye. A Data Imputation Method for Multivariate Time Series Based on Generative Adversarial Network. *Neurocomputing*, 360:185–197, 2019.

[Hardkernel, 2014] Hardkernel. Odroid-xu3. https://www.hardkernel.com/shop/odroid-xu3/ Accessed 5/29/2024, 2014.

[Hossain *et al.*, 2020] Tahera Hossain, Md Ahad, Atiqur Rahman, and Sozo Inoue. A Method for Sensor-Based Activity Recognition in Missing Data scenario. *Sensors*, 20(14):3811, 2020.

[Hussein and Bhat, 2023] Dina Hussein and Ganapati Bhat. CIM: A Novel Clustering-Based Energy-Efficient Data Imputation Method for Human Activity Recognition. *ACM Transactions on Embedded Computing Systems (TECS)*, 22(5s):1–26, 2023.

[Hussein *et al.*, 2022a] Dina Hussein, Taha Belkhouja, Ganapati Bhat, and Janardhan Rao Doppa. Reliable Machine Learning for Wearable Activity Monitoring: Novel Algorithms and Theoretical Guarantees. In *Proc. Int. Conf. Comput-Aided Des.*, pages 1–9, 2022.

[Hussein *et al.*, 2022b] Dina Hussein, Ganapati Bhat, and Janardhan Rao Doppa. Adaptive Energy Management for Self-Sustainable Wearables in Mobile Health. In *Proceedings of AAAI*, pages 11935–11944. AAAI Press, 2022.

[Hussein *et al.*, 2022c] Dina Hussein, Aaryan Jain, and Ganapati Bhat. Robust Human Activity Recognition Using Generative Adversarial Imputation Networks. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 84–87, 2022.

[Hussein *et al.*, 2023] Dina Hussein, Taha Belkhouja, Ganapati Bhat, and Janardhan Rao Doppa. Energy-Efficient Missing Data Recovery in Wearable Devices: A Novel Search-Based Approach. In *2023 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pages 1–6, 2023.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *The Int. Conf. on Learning Representations (Poster)*, 2015.

[Lara and Labrador, 2012] Oscar D Lara and Miguel A Labrador. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Commun. Surveys & Tut.*, 15(3):1192–1209, 2012.

[Limaye and Adegbija, 2018] Ankur Limaye and Tosiron Adegbija. HERMIT: A Benchmark Suite for the Internet of Medical Things. *IEEE Internet of Things Journal*, 5(5), 2018.

[Liu *et al.*, 2020] Shengzhong Liu, Shuochao Yao, Yifei Huang, Dongxin Liu, Huajie Shao, Yiran Zhao, Jinyang Li, Tianshi Wang, Ruijie Wang, Chaoqi Yang, and Tarek Abdelzaher. Handling Missing Sensors in Topology-Aware IoT Applications with Gated Graph Neural Network. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–31, 2020.

[Maetzler *et al.*, 2013] Walter Maetzler, Josefa Domingos, Karin Srulijes, Joaquim J Ferreira, and Bastiaan R Bloem. Quantitative Wearable Sensors for Objective Assessment of Parkinson's Disease. *Movement Disorders*, 28(12):1628–1637, 2013.

[Maetzler *et al.*, 2016] Walter Maetzler, Jochen Klucken, and Malcolm Horne. A Clinical View on the Development of Technology-Based Tools in Managing Parkinson's Disease. *Movement Disorders*, 31(9):1263–1271, 2016.

[Mosenia *et al.*, 2017] Arsalan Mosenia, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K Jha. Wearable Medical Sensor-Based System Design: A Survey. *IEEE Transactions on Multi-Scale Computing Systems*, 3(2):124–138, 2017.

[Pires *et al.*, 2020] Ivan Miguel Pires, Faisal Hussain, Nuno M Garcia, and Eftim Zdravevski. Improving Human Activity Monitoring by Imputation of Missing Sensory Data: Experimental Study. *Future Internet*, 12(9):155, 2020.

[Reiss and Stricker, 2012] Attila Reiss and Didier Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. In *International Symposium on Wearable Computers*, pages 108–109, 2012.

[Sachs *et al.*, 2022] Jeffrey D Sachs, Christian Kroll, Guillame Lafortune, Grayson Fuller, and Finn Woelm. *Sustainable Development Report 2022*. Cambridge University Press, 2022.

[Schmidt *et al.*, 2018] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. In *Proc. of the 20th ACM Int. Conf. on Multimodal Interaction*, pages 400–408, 2018.

[Shoaib *et al.*, 2014] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors*, 14(6):10146–10176, 2014.

[Talukder *et al.*, 2022] Sabera Talukder, Jennifer J Sun, Matthew Leonard, Bingni W Brunton, and Yisong Yue. Deep Neural Imputation: A Framework for Recovering Incomplete Brain Recordings. *arXiv:2206.08094*, 2022.

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

[Villar *et al.*, 2016] Jose R Villar, Paula Vergara, Manuel Menéndez, Enrique de la Cal, Víctor M González, and Javier Sedano. Generalized Models for the Classification of Abnormal Movements in Daily Life and its Applicability to Epilepsy Convulsion Recognition. *International Journal of Neural Systems*, 26(06):1650037, 2016.

[Wilhelm *et al.*, 2015] Mathias Wilhelm, Daniel Krakowczyk, Frank Trollmann, and Sahin Albayrak. eRing: Multiple Finger Gesture Recognition with One Ring Using an Electric Field. In *Proc. Int. Work. on Sensor-based Activity Recognition and Interaction*, pages 1–6, 2015.

[Xu *et al.*, 2018] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-Grained Text to Image Generation With Attentional Generative Adversarial Networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1316–1324, 2018.

[Yoon *et al.*, 2018] Jinsung Yoon, James Jordon, and Mihaela Schaar. GAIN: Missing Data Imputation Using Generative Adversarial Nets. In *Proc. Int. Conf. on Machine Learning*, pages 5689–5698, 2018.