Robust Human Activity Recognition using Generative Adversarial Imputation Networks

Dina Hussein[1], Aaryan Jain[2], and Ganapati Bhat[1]
[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99164
[2]Tesla STEM High School, Redmond, WA, 98052
Email: dina.hussein@wsu.edu, jain_aaryan@outlook.com, ganapati.bhat@wsu.edu

***Abstract*—Human activity recognition (HAR) is widely used in applications ranging from activity tracking to rehabilitation of patients. HAR classifiers are typically trained with data collected from a known set of users while assuming that all the sensors needed for activity recognition are working perfectly and there are no missing samples. However, real-world usage of the HAR classifier may encounter missing data samples due to user error, device error, or battery limitations. The missing samples, in turn, lead to a significant reduction in accuracy. To address this limitation, we propose an adaptive method that either uses low-power mean imputation or generative adversarial imputation networks (GAIN) to recover the missing data samples before classifying the activities. Experiments on a public HAR dataset with 22 users show that the proposed robust HAR classifier achieves 94% classification accuracy with as much as 20% missing samples from the sensors with 390 μJ energy consumption per imputation.**

I. Introduction

Research on human activity recognition (HAR) has grown significantly due to its applications in health monitoring, fitness, and defense applications [1, 2]. Development of a HAR algorithm typically involves four steps: data collection and labeling, segmentation, feature generation, and classifier training [1]. Data collection step uses a wearable device or a smartphone to obtain sensor data when the user is performing the activities of interest. The collected data is then labeled manually so that supervised learning algorithms can be trained. Next, raw data from the device is segmented such that each segment of data contains a single activity. The segments can either be fixed length [1, 3, 4] or variable length [5]. Data segments are then used to generate the features and train a supervised learning classifier [1]. Finally, the trained classifier is used at runtime to identify the activities of the user in real-world scenarios.

The vast majority of HAR algorithms ensure that all the sensors are working perfectly and there are no missing data samples during data collection. However, when the trained classifier is used in real-world scenarios, the sensors may malfunction, leading to missing data samples. The sensor positions may also drift from their original locations due to user error or long-term usage. Missing data leads to inaccuracies in the feature generation and classification. For instance, the accuracy of the HAR classifier proposed in [6] drops by 40% with only 10% missing data. Introducing examples with missing data in the training process also does not yield significant improvements since it cannot account for all missing data scenarios. Therefore, there is a strong need to develop robust HAR approaches that can recover missing data on the fly for accurate classification.

This paper presents an adaptive algorithm to recover missing sensor data in HAR classifiers. The algorithm starts by detecting missing data samples as the sensor data is received by the device before any pre-processing or segmentation. It is crucial to recover the data before segmentation to ensure that the missing data does not lead to inaccurate segmentation. Next, depending on the length of missing data sequence, the algorithm uses a generative adversarial imputation network (GAIN) [7] or a mean-based imputation technique to recover the missing data. Specifically, GAIN is used for longer missing data sequences while the mean-based imputation is used for shorter missing data sequences to reduce the power consumption of the device. After recovering the missing data, our robust HAR approach follows the common steps of HAR including segmentation, feature generation, and classification.

We validate the proposed data imputation algorithm on three publicly available datasets [3, 4, 6]. Our experiments show that we achieve accuracy that is comparable to the accuracy with clean data without retraining the classifier for all three datasets. Furthermore, segmentation with data imputation is able to identify, on average, 96% of the segments that are present in the baseline with clean data. Finally, measurements on the TI-CC2652 microcontroller (MCU) [8] show that the proposed approach consumes 390 μJ of energy per imputation. In summary, this paper makes the following contributions:

- An adaptive algorithm for data imputation in HAR.
- Introducing robustness into the segmentation algorithm to segment sensor data in the presence of missing samples.
- Experimental evaluation with three public datasets to demonstrate that the proposed approach achieves high accuracy with 390 μJ of energy consumption per imputation.

II. Related Work

HAR is receiving increased attention due to its applications in health monitoring, rehabilitation, and fitness [1, 2]. HAR classifiers are typically trained using clean data with no missing samples [1, 3, 4]. However, this assumption does not hold true in many real-world scenarios where samples can go missing [9]. The missing samples create the need for development of efficient and low-overhead methods for runtime data imputation.

Several techniques have been proposed for data imputation in HAR [10–13]. These methods include k-nearest neighbor (k-NN) and autoencoders [11, 12]. However, k-NN is not suitable for implementation on low-power devices as it imposes

978-3-9819263-6-1/DATE22/©2022 EDAA

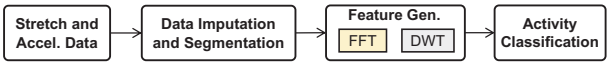

Fig. 1: Overview of the proposed robust HAR classifier

significant computational and power requirements. Moreover, prior approaches for data imputation in HAR recover features and do not recover the raw data. This can lead to inaccuracy in segmentation when variable-length segments are used. To address this issue, we propose an adaptive imputation method that recovers the raw data before activity segmentation. Experiments on three publicly available datasets show that the proposed method provides accuracy that is comparable to the baseline accuracy without any missing data.

III. Human Activity Recognition Preliminaries

This section describes the main steps in the proposed robust HAR algorithm, as shown in Figure 1. We use the w-HAR dataset [6] as the primary dataset in this work. The w-HAR dataset contains stretch sensor [14] and accelerometer data from 22 users while performing 8 activities. Data from the two sensors is used for activity classification, as described next.

Segmentation and Data Imputation: The first step in the HAR pipeline is dividing the streaming data from the sensors into distinct activity segments. Most approaches for HAR use fixed-length activity segments with 2 s to 10 s duration. However, fixed-length segments are not ideal since they can contain multiple activities in a single segment, leading to classification errors. Therefore, we adopt the variable-length segmentation algorithm proposed in [6] to generate segments that are 1 s to 3 s in length. However, the segmentation algorithm is susceptible to errors when there are missing samples.

To overcome the limitations of missing data, we integrate a data imputation block within the segmentation algorithm. Imputation is typically performed by analyzing blocks of data to ensure that the imputation algorithm is able to observe the distribution of the data. To this end, the imputation block stores the streaming sensor data in a fixed-sized buffer until it becomes full. Once the buffer is full, the proposed adaptive imputation algorithm (Section IV) recovers any missing data. In this work, we set the buffer length to 3 s (max. window size) while noting that other values can be used with a similar effect.

Feature Generation: The next step after segmentation is to generate features for classification. Since our focus is not feature generation, we re-use the feature set provided with the w-HAR dataset [6]. The feature set includes discrete wavelet transform (DWT) of the accelerometer data and fast Fourier transform (FFT) of the stretch sensor data.

Classification: The final step in the activity recognition pipeline is to use the features to identify the current activity. Any supervised learning algorithm can be used to obtain the activity classification. In our implementation, we use a low-overhead neural network to classify the activities. The neural network contains two hidden layers with ReLU activation and an output layer with softmax activation. The hidden layers contain four and eight neurons, respectively. The classifier is trained with clean baseline data and the same classifier is used to analyze the accuracy improvements provided by imputation.

IV. Proposed Adaptive Imputation Algorithm

This section describes two possible missing data types in HAR, our imputation method for each missing data type, and the adaptive algorithm that handles both types of missing data.

A. Missing Data Classification

Missing data in HAR can be classified into two main categories as follows. In the first case, we have isolated missing samples that are not clustered around any particular time instance. Causes of this type of missing data include limited communication bandwidth and buffer overflow in a sensor. We call this type of missing data as *random missing data* due to the randomness in the timing and number of missing samples.

The second type of missing data includes cases where a sequence of samples is missing. This can occur when one or more sensors have to go into a low-power state due to energy limitations. As a result, the sensor is unable to collect a sequence of samples for a period of time. We refer to this type of missing data pattern as *block missing data*.

B. Mean Imputation

HAR algorithms are typically executed on low-power wearable devices or smartphones, both of which are energy-constrained. Therefore, any imputation method must have low overhead in terms of energy and execution time. In this section, we propose a low-overhead imputation scheme for the random missing data scenario based on a key insight about the frequency of human activities. Specifically, studies have shown that human activities have frequency components in the order of few Hertz [6]. This means that sensors do not experience sudden changes in their values. Using this insight, we propose to impute the *random missing data* samples by taking the mean of the available samples around the missing sample as follows:

$$\hat{s}[k] = \frac{1}{2S} \sum_{i=-S}^{S} s[k+i] \tag{1}$$

where k missing sample index, $\hat{s}[k]$ is the imputed sample, S is the window length, and $s[k+i]$ are the available data.

Figure 2(a) shows an example of the mean imputation in the random missing data scenario for the stretch sensor. As expected, the sensor data experiences slow changes with time. It is seen that the imputed data closely matches the reference data. This shows that the mean imputation method is effective in recovering data in the *random missing data* case.

C. Generative Adversarial Imputation Network

The mean imputation method does not provide high accuracy when there is a sequence of missing data. In this case, the imputation algorithm must learn the underlying data distribution and use it to recover the data. To this end, we propose to use GAIN [7] for data imputation in the block missing data case.

GAIN is a class of adversarial networks that is well suited for data imputation [7]. Following the general GAN concept, GAIN consists of a generator and discriminator. The job of the generator is to accurately impute data while the discriminator's goal is to distinguish between imputed and observed data. In what follows, we provide an overview of GAIN.

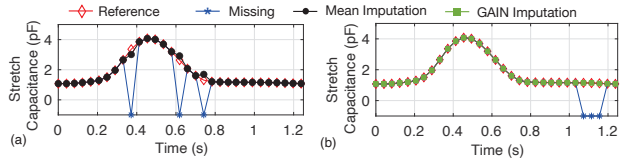

Fig. 2: Illustration of the (a) mean and (b) GAIN imputation methods. The missing samples are indicated with a fixed value of -1.

Generator: Consider that the samples in a given buffer are denoted by $\mathbf{X}$ and the number of samples is equal to N. We define a mask vector $\mathbf{M}$ to denote the missing samples such that $m_i = 1$ when the sample is observed and $m_i = 0$ otherwise. Given the values of $\mathbf{X}$ and $\mathbf{M}$ for any window, the generator G generates a vector $\bar{\mathbf{X}}$ as:

$$\bar{\mathbf{X}} = G(\mathbf{X}, \mathbf{M}, (1 - \mathbf{M}) \odot \mathbf{Z}) \tag{2}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{Z}$ is a noise variable used to initialize the missing samples. The vector $\bar{\mathbf{X}}$ includes generated samples even for data that is observed. We do not need to use the observed samples from $\bar{\mathbf{X}}$ since they are already present in $\mathbf{X}$. Therefore, we combine $\mathbf{X}$ and $\bar{\mathbf{X}}$ to obtain the complete data as:

$$\hat{\mathbf{X}} = \mathbf{M} \odot \mathbf{X} + (1 - \mathbf{M}) \odot \bar{\mathbf{X}} \tag{3}$$

where $\hat{\mathbf{X}}$ is the final data vector. The above equation ensures that the final data vector contains the observed samples where available and uses the imputed values when the data is missing.

Discriminator: The discriminator aims to determine the observed and imputed components of $\hat{\mathbf{X}}$. In other words, the discriminator tries to predict the mask vector $\mathbf{M}$.

Objective function: Since our primary goal is imputing data using the generator, we present the objective function for the generator and refer readers to [7] for details on the discriminator objective. Specifically, the generator objective function is:

$$\min_{G} \sum_{i=1}^{K} \alpha \mathbf{M}_i^T ||\bar{\mathbf{X}} - \mathbf{X}|| - (1 - \mathbf{M_i})^T \log(\hat{\mathbf{M}}_i) \tag{4}$$

where K is the number of training examples for the generator, $\hat{\mathbf{M}}_i$ is the prediction of the mask vector by the discriminator, and α is a hyperparameter. The first part of the objective evaluates how well samples from the generator match the observed values when the data is not missing ($m_i = 1$). It ensures that the generator is able to learn the underlying data distribution. The second part of the objective trains the generator to ensure that the discriminator is unable to distinguish between real and imputed samples. The hyperparameter α is used to control the relative importance of each part of the objective.

Figure 2(b) shows an illustration of the imputation achieved by applying GAIN on stretch sensor data. A portion of samples from 1 s and 1.2 s are missing. We use the trained GAIN to impute these samples. The imputed samples closely match the reference samples, showing the effectiveness of GAIN in imputing data in the *block missing data* scenario.

D. Adaptive Imputation Algorithm

HAR algorithms deployed in the real world can encounter missing data in either random or block patterns. To this end, we combine the mean and GAIN imputation methods to efficiently impute the data as a function of the missing data pattern. For each 3 s window, we determine the length of the longest sequence of missing data. If the length of the longest sequence is greater than a threshold, we apply GAIN to impute the data, otherwise we use mean imputation. The threshold must be chosen carefully to ensure that the execution time and energy overhead are minimized. This is because GAIN has a higher complexity, leading to higher execution time and energy overhead than the mean imputation. Therefore, we perform a design space exploration of the threshold values to determine the optimal threshold that minimizes imputation error and energy consumption. Once the data is imputed, it is passed to the other HAR blocks to recognize the activity.

V. Experimental Evaluation

A. Experimental Setup

Wearable device: We use a prototype based on the Texas Instruments CC2652R MCU [8] as the primary device for performing data imputation and activity recognition. We implement the adaptive imputation algorithm on the wearable device prototype to characterize its runtime and energy overhead.

Datasets: We use the w-HAR dataset described in Section III as the primary dataset. In addition to w-HAR, we also use the WISDM [3] and Shoaib et al. [4] datasets. The WISDM dataset contains accelerometer data for 29 users performing six activities while Shoaib et al. [4] dataset provides accelerometer data for ten users performing seven activities.

Missing Data Implementation: We introduce missing data in each 3 s window of the raw data by randomly choosing the missing samples indices in the 3 s window. The missing samples are set to zero for further processing. To evaluate the effectiveness of the proposed approach under different lengths of missing data, we use the following configurations of the missing data percentages: {2%, 5%, 10%, 20% and 30%}. In case of the *block missing data*, we randomly choose the start time of the missing data and then set the missing samples.

Hyperparameters: The hyperparameters in this work include the structure of the GAIN generator, the value of α, and the threshold for adaptive imputation. We use a neural network with three fully connected hidden layers with ReLU activation for the generator. The first hidden layer contains four neurons, while the second and third hidden layers contain eight neurons for the stretch sensor and 22 neurons for each accelerometer axis. The output layer with sigmoid activation produces 3 s of imputed data for the stretch sensor and accelerometer, respectively. The value of α is set to 100 such that the generator accurately imputes the data. The threshold for adaptive imputation is set to five because it gives the best trade-off between imputation accuracy and energy overhead.

B. Segmentation with Missing Data

We start our experimental evaluation by analyzing the effect of missing data on segmentation since it is the first step in HAR. We first obtain the baseline set of segments by using the dataset without any missing data. Then, we introduce the missing data and re-run the segmentation algorithm. Figure 3

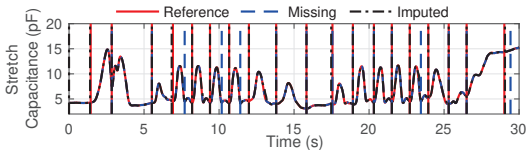

Fig. 3: Illustration of segmentation with the proposed imputation algorithm

TABLE I: Accuracy of segmentation with increasing missing data percentage

Missing Percentage	2%	5%	10%	20%	30%
Segments with missing data	96%	92%	86%	80%	78%
Segments with imputation	99%	98%	97%	95%	93%

shows an example of the segments identified after applying the imputation algorithm for a specific experiment in the dataset. The segments identified after applying the imputation closely match the segments with clean data. In strong contrast, the segments identified with missing data differ significantly from the reference. These errors, in turn, lead to lower classification accuracy. We summarize the results of segmentation accuracy for the entire dataset in Table I. The numbers in the table correspond to the percentage of baseline segments identified by each of the segmentation scenarios. We see that the imputation algorithm identifies more than 93% of segments while the segmentation with missing data only identifies 78% of the segments with 30% missing data. In summary, these results show that the proposed imputation algorithm is able to provide robust segmentation for HAR in the presence of missing data.

C. Accuracy Analysis with Robust Segmentation

The primary motivation of the proposed imputation algorithm is to achieve high recognition accuracy in the presence of missing data. To this end, we generate features from the segments identified in Section V-B. Figure 4 shows the classification accuracy as a function of the missing data percentage. The classifier used in this experiment is trained with baseline training data without missing samples. The segments obtained with missing data have a low accuracy due to the effect of missing data on both segments and the features. In contrast, the segments with imputed samples are able to provide a significantly higher accuracy. The accuracy reduces with increase in missing data percentage as we encounter higher degree of errors in the segmentation and imputation, which affect the classification accuracy as well. In summary, these results show that the proposed approach provides high accuracy in both segmentation and classification when data samples are missing.

D. Accuracy for WISDM and Shoaib Datasets

In this section, we apply the proposed adaptive imputation approach to the WISDM and Shoaib datasets. Since the default implementation of these datasets does not use variable-length segments, we use 10 s segments for both datasets to generate features and classify the activities. Table II summarizes the classification accuracy under no missing data, missing data, and imputation scenarios, respectively. The classifiers obtain 90% and 97% accuracy for the WISDM and Shoaib datasets when no missing data is present. The accuracy drops significantly with missing data due to inaccuracies in feature generation. The proposed imputation approach is able to improve the

Fig. 4: Classification accuracy as a function of the missing data percentage

TABLE II: Classification accuracy for WISDM and Shoaib datasets

Dataset	Reference	Without Imputation	With Imputation
WISDM	90%	80%	85%
Shoaib	97%	85%	97%

accuracy of classification to 85% and 97%, respectively, thus demonstrating its applicability to a wide range of HAR datasets.

E. Implementation Overhead

We measure the overhead of the proposed approach by implementing it on the TI-CC2652 MCU. The mean imputation takes about 0.1 ms of execution time, which results in an energy consumption of 0.7 µJ. Similarly, each invocation of GAIN takes 9 ms for the stretch sensor and 15 ms for each accelerometer direction. This amounts to 99 µJ and 168 µJ of energy for stretch and accelerometer, respectively. In our adaptive method, GAIN was used to impute 47% of the stretch segments and 68% of the accelerometer segments, resulting in an average energy consumption of 390 µJ per imputation.

VI. Conclusion

HAR is an essential component of personalized healthcare and activity monitoring. However, most classifiers designed for HAR do not consider missing data in their training process, which can lead to reduced accuracy in real-world usage. To address this issue, we proposed an adaptive imputation algorithm that recovers missing sensor data before activity segmentation. Experiments with three publicly available datasets showed that the proposed algorithm effectively recovers sensor data and achieves accuracy that is comparable to the accuracy with clean data while using the same classifier weights.

References

[1] O. D. Lara *et al.*, "A Survey on Human Activity Recognition Using Wearable Sensors," *IEEE Commun. Surveys & Tut.*, vol. 15, no. 3, pp. 1192–1209, 2012.
[2] A. J. Espay *et al.*, "Technology in Parkinson's Disease: Challenges and Opportunities," *Movt. Disorders*, vol. 31, no. 9, pp. 1272–1282, 2016.
[3] J. R. Kwapisz *et al.*, "Activity Recognition Using Cell Phone Accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
[4] M. Shoaib *et al.*, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014.
[5] A. Wang *et al.*, "A Comparative Study on Human Activity Recognition Using Inertial Sensors in a Smartphone," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4566–4578, 2016.
[6] 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.
[7] J. Yoon, J. Jordon, and M. Schaar, "GAIN: Missing Data Imputation Using Generative Adversarial Nets," in *ICML*, 2018, pp. 5689–5698.
[8] Texas Instruments Inc., "CC2652R Microcontroller," [Online] https://www.ti.com/product/CC2652R, accessed 1 November 2020, 2018.
[9] L. Kong *et al.*, "Data Loss and Reconstruction in Sensor Networks," in *IEEE INFOCOM*, 2013, pp. 1654–1662.
[10] T. Hossain *et al.*, "A Comparative Study on Missing Data Handling Using Machine Learning for Human Activity Recognition," in *ICIEV & icIVPR*, 2019, pp. 124–129.
[11] 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.
[12] A. Saeed *et al.*, "Synthesizing and Reconstructing Missing Sensory Modalities in Behavioral Context Recognition," *Sensors*, vol. 18, no. 9, p. 2967, 2018.
[13] O. M. Prabowo *et al.*, "Missing Data Handling Using Machine Learning for Human Activity Recognition on Mobile Device," in *Proc. ICISS*, 2016, pp. 59–62.
[14] B. O'Brien *et al.*, "Stretch Sensors for Human Body Motion," in *Proc. Electroactive Polymer Actuators and Devices*, vol. 9056, 2014, p. 905618.