ParaLabel: Autonomous Parameter Learning for Cross-Domain Step Counting in Wearable Sensors

Parastoo Alinia, *Student Member, IEEE,* Ramin Fallahzadeh, *Member, IEEE,* Christopher P. Connolly, and Hassan Ghasemzadeh, *Senior Member, IEEE*

***Abstract*—Wearable step counters, also referred to as activity trackers, have been developed for health and activity monitoring, as well as for step tracking. These trackers, however, produce unreliable measurements during slow walking and when walking with assistive devices (i.e., aided walking). To address this challenge, in this article, we introduce, *ParaLabel*, a filter-based step counting algorithm that is reliable against various walking velocities and intensities. ParaLabel addresses this problem by learning a filter cut-off frequency autonomously in a new domain without the need for collecting sensor data and manually tuning the algorithm parameter for a different velocity and/or pattern of walking. We formulate this problem as a transfer learning problem in which the new filter cut-off frequency is transferred from a bank containing previously fine-tuned parameters from different domain(s). Our extensive analysis using real data collected from 15 participants while wearing an accelerometer sensor on their chest, wrist, or left pocket demonstrates the superiority of ParaLabel to two commercially available trackers worn on the same body location, and state-of-the-art techniques. ParaLabel achieves $96.3\% - 99.9\%$ accuracy during walking on a treadmill at three different velocities, $98.2\% - 99.9\%$ accuracy during walking with a shopping cart, and $89.3\%-97.3\%$ accuracy while walking with the aid of a walker.**

***Index Terms*—Step counting, wearable sensors, peak detection, low-pass filter, frequency components, K-nearest neighbor, time-domain features, cross-subject transfer learning.**

I. INTRODUCTION

Internet-of-things (IoT) has emerged as a promising paradigm for interconnecting computing devices such as wearable sensors and smartphones with cloud computing platforms for seamless interactions, health monitoring, and clinical decision making. Wearable trackers, as a rapidly growing component of IoT, have gained much attention recently due to their potentials to improve the health and well-being of their users. Promoting physical activity levels is an important factor to sustain and improve cardiovascular health [1]. An effective direction to evaluate the activity level of individuals is to continuously track the number of steps they take daily [2]. Jawbone, Fitbit, Misfit, and Garmin are among many acceleration-based tracking devices that are used to measure and monitor physical activity. These trackers are small, non-obtrusive, user friendly, and provide an objective indicator of physical activity behavior. Furthermore, they intend to avoid common sources of error in subjective measurement (e.g., self-report) [1], [3]. However, these activity trackers produce unreliable and inconsistent measurements when deployed in uncontrolled environments in particular during irregular, low-intensity walking [4]. These issues can potentially lead to unsustainable utilization of wearable activity tracking devices.

P. Alinia, R. Fallahzadeh, and H. Ghasemzadeh are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 99164.

Chris Connolly is with the Education Department, Washington State University, Pullman, WA, 99164.

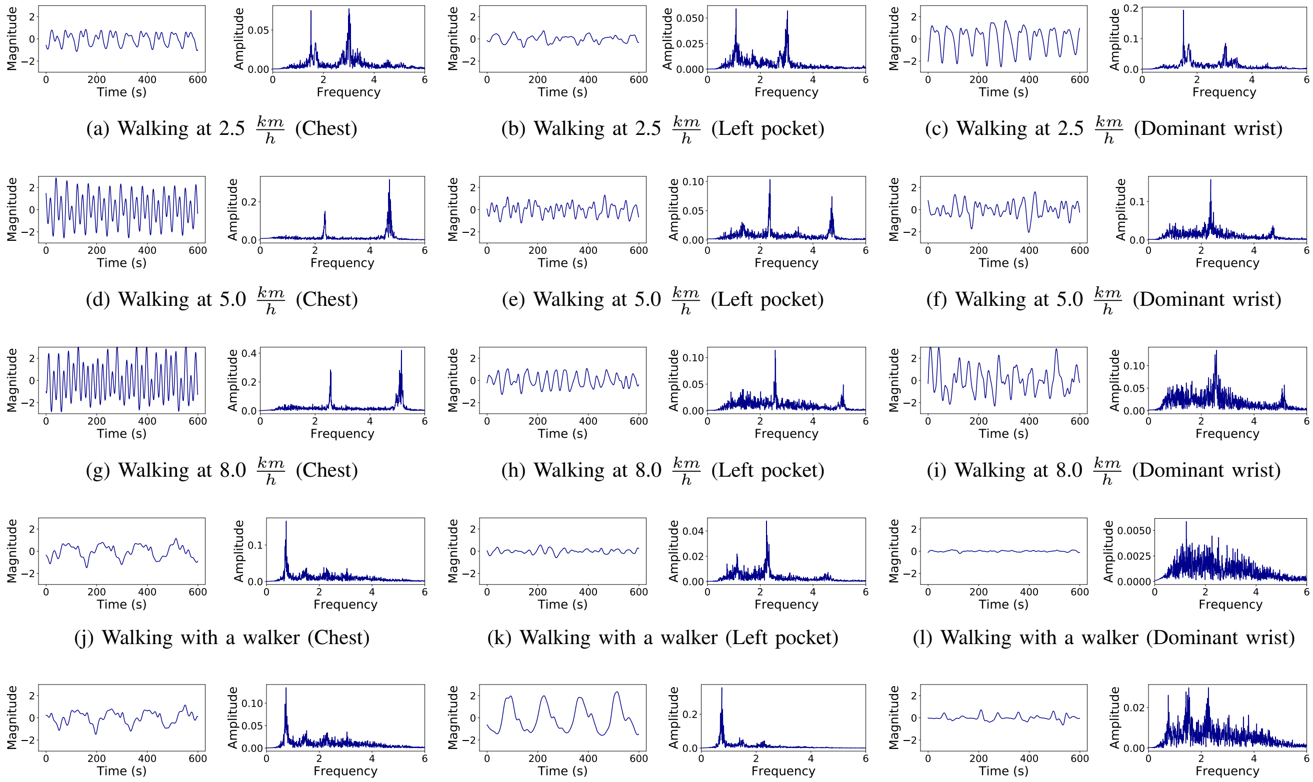
Researchers have proposed several approaches to detect the steps from signals captured by wearable inertial sensors [5]–[8]. These algorithms detect the steps by capturing the periodic patterns from the sensor data such as acceleration and angular velocity. A major challenge with existing step-counting algorithms is that they experience substantial accuracy drop during slow (< 0.8 m/s) walking [9], [10], and those requiring an assistive tool such as walking with a walker. Although few studies propose step-counting algorithms for free walking [11] and wheeled walking frame [12], to the best of our knowledge, none of the previous studies have proposed a step-counting algorithm that generalizes to various speed and pattern of walking such as low-intensity walking, walking with a walker in addition to regular walking. The problem with low-intensity walking becomes more evident knowing that directly applying machine learning algorithms will not result in generalizable models to different subjects and various activities. Therefore, achieving high precision in detecting steps (e.g., $> 95\%$ accuracy) in such situations requires manual tuning of the algorithm parameters. As elaborated in section Section II-A, even, the error rate of popular activity trackers such as Fitbit increases significantly (e.g., from 3.5% to 39.1%) when transitioning from normal walking to low intensity or aided walking. We propose ParaLabel, an autonomous filter-based step-counting algorithm that transfers the filter parameters learned in one setting to a new setting, without manually fine-tuning the parameters. Therefore, ParaLabel generalizes to various patterns and speeds in walking without the need to collect new data and tune the parameters.

II. PRELIMINARIES

In this section, we discuss the motivations for our research followed by a presentation of the related research.

A. Motivation

The common approach to detect steps using motion signals such acceleration is to identify the occurrence of a particular feature in the signal. For example, peaks in the magnitude of the accelerometer signal show the steps taken by the user while walking [13]. However, variations in the pattern and the velocity of walking can easily mislead a traditional step



(a) Walking at 2.5 $\frac{km}{h}$ (Chest) (b) Walking at 2.5 $\frac{km}{h}$ (Left pocket) (c) Walking at 2.5 $\frac{km}{h}$ (Dominant wrist)

(d) Walking at 5.0 $\frac{km}{h}$ (Chest) (e) Walking at 5.0 $\frac{km}{h}$ (Left pocket) (f) Walking at 5.0 $\frac{km}{h}$ (Dominant wrist)

(g) Walking at 8.0 $\frac{km}{h}$ (Chest) (h) Walking at 8.0 $\frac{km}{h}$ (Left pocket) (i) Walking at 8.0 $\frac{km}{h}$ (Dominant wrist)

(j) Walking with a walker (Chest) (k) Walking with a walker (Left pocket) (l) Walking with a walker (Dominant wrist)

(m) Pushing a shopping cart (Chest) (n) Pushing a shopping cart (Left pocket) (o) Pushing a shopping cart (Dominant wrist)

Fig. 1. (a) Magnitude signal and (b) Spectrum Amplitude from walking with a walker, walking with a shopping cart, walking on a treadmill at 2.5 $\frac{km}{h}$, 5.0 $\frac{km}{h}$, and 8.0 $\frac{km}{h}$ for one subject.

counting algorithm, which operates on pre-defined parameters [14]. In what follows, we further elaborate on this challenge by examining real-data collected with inertial sensors during various walking activities. Figure 1 shows the magnitude and frequency spectrum of a tri-axial accelerometer worn on the chest, left pocket, and dominant wrist of a subject while walking under different conditions. In particular, each sub-figure corresponds to one specific activity performed by the same subject. The frequency spectrum chart represents the frequency components of a periodic signal. The spikes in the amplitude show the cycle frequency of the signal.

As shown in Figure 1, velocity and pattern of walking, and the location of the senor impact the shape of the signal in the time domain. Walking at a higher velocity and regular walking such as walking on a treadmill at 5.0 $\frac{km}{h}$ and 8.0 $\frac{km}{h}$, as shown in Figure 1-(d–i), exhibit more clear periodic signal readings than walking at lower velocity and those performed with assistive devices such as walking with a walker and shopping cart in Figure 1-(d–i). Moreover, wearing the sensor on the dominant wrist adopt noisier and less periodic signal comparing the sensor on the chest and the left pocket, specifically in aided walking such as walking with a walker, shown in Figure 1-(l) and shopping cart, shown in Figure 1-(o). These issues become more challenging considering that wearable sensors need to be deployed for long-term health monitoring where the signals captured with inertial sensors become irregular, and noisy over time. Therefore, static algorithms fail to accurately detect the steps in such conditions.

As shown in Figure 2, a trivial solution with respect to the noisy signals is to use a low-pass filter to minimize the effect of the high-frequency noise [15]. The peaks of the smoothed signal represent the occurrence of the steps. On the basis of spectrum information, the maximum amplitude in the frequency spectrum corresponds to the step frequency while walking. Filtering the signal with a low-pass filter with a cut-off frequency equal to the step frequency results in a smoothed periodic signal which peaks represent steps. However, directly assigning the frequency with the largest amplitude to the step frequency leads to inaccurate results because of the noise in the accelerometer signal particularly for low-intense and irregular walking activities [16]. To address these issues, we develop ParaLabel, a frequency-based step-counting algorithm that autonomously learns its parameter based on the intensity and pattern of walking. The proposed algorithm is implemented in the frequency domain to mitigate the influence of random noises in time-domain.

B. Related Work

One can categorize existing step-counting algorithms into time-domain, frequency-domain, and feature-based clustering. The most common and simple approach to detect steps from inertial sensors is to focus on a particular feature on the signal and use a threshold value to identify steps [17]–[19]. In threshold-based methods, each time the sensory data meet a

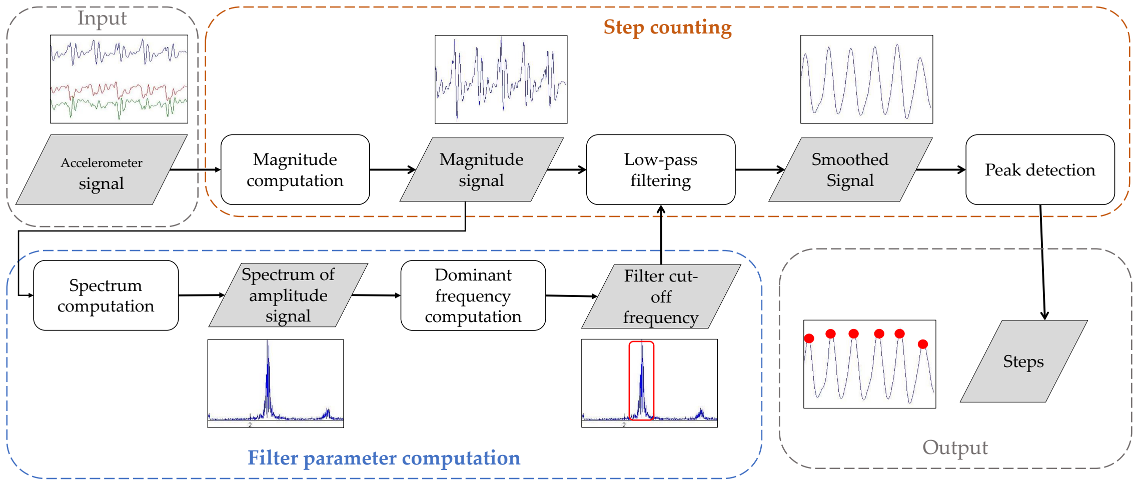


Fig. 2. Trivial step detection algorithm based on peak detection: extracting magnitude of raw tri-axial accelerometer signal (top), and smoothing the signal by low-pass filtering (bottom).

predefined threshold, the number of steps is increased by one. Despite the simplicity, the main issue with these methods is the need to estimate optimal thresholds for uncontrolled walking such as slow-walking or aided-walking and in unconstrained environments [20].

Several time-domain step-counting algorithms are implemented based on the cyclic nature of walking. The auto-correlation approach detects cyclic periods directly in the time domain through evaluating auto-correlation [18], [21], [22]. Peak detection [5] or zero-crossing counting [6] on low-pass filtered accelerometer signals can also be used to find specific steps. For smartphone data, which is not firmly attached to the body, the signal magnitude is computed instead of the orthogonal axis of the sensor. In [7], [15], the authors apply low-pass filtering to remove interference. In [23], the authors limit the time interval between two peaks to reduce misjudgments. In [24], the authors apply two filters to reduce jitters in accelerations. In [11], vertical acceleration data are used to infer steps for unconstrained smartphones, but sensor fusion is used to obtain vertical acceleration data. Other time-domain algorithms include template matching techniques such as Dynamic time warping [8], which attempts to match an off-line-processed template of one step to the input signal to identify each step.

The frequency-domain approaches identify the steps according to the frequency components of successive windows of measurements based on short-term Fourier transform (STFT) [25], fast Fourier transform (FFT) [20], and continuous/discrete wavelet transforms (CWT/DWT) [25], [26]. In [20], steps are identified by extracting frequency domain features in acceleration data through FFT, and the accuracy of 87.52% is achieved. Additionally, FFT is employed in [27] to smooth acceleration data and then peak-detection is used to count the steps. In [16], steps are identified by multiplying the step frequency extracted from the FFT of the time-domain gyroscope signal and walking duration, and an accuracy of 95.7% is achieved. These methods can generally achieve high accuracy but suffer from either resolution issues or computational overheads [16].

The feature clustering approaches apply machine learning algorithms, e.g., Hidden Markov models (HMMs) [28], to identify gait segments from a signal and count the steps from the segments based on both time-domain and frequency-domain features extracted from sensory data [29]. However, the utility of these techniques in low-intensity activities and across individuals has not been investigated [16].

Based on a comprehensive comparison among existing step-counting algorithms in [4], windowed peak detection, HMM and, CWT are considered the best step-counting algorithms. The proposed step-counting algorithm in this paper, ParaLabel, benefits from the advantages of both time-domain and frequency-domain algorithms and also devises machine learning models to enhance the generalizability of the algorithm. ParaLabel is designed based on a simple peak detection algorithm on a low-pass filtered signals. It estimates the frequency parameter of the filter using time- and frequency-domain features extracted from the sensor data. The process of parameter estimation is autonomous and does not involve human inputs.

The technique introduced in this paper can be viewed as a transfer learning method for use with step counting algorithms in wearable sensor systems. Prior research on designing transfer learning techniques for physical activity monitoring focused primarily on classification tasks such as human activity recognition [30], [31]. To the best of our knowledge, our work in this paper is the first study to design an approach for autonomous reconfiguration of signal processing algorithms for wearable-based step counting.

C. Contributions

In this section, we summarize the contributions of this paper. We propose ParaLabel, a robust filter-based step counting algorithm using accelerometer signals. We propose a transfer learning algorithm that transfers the optimal parameters from one setting to another using a kNN classification based on

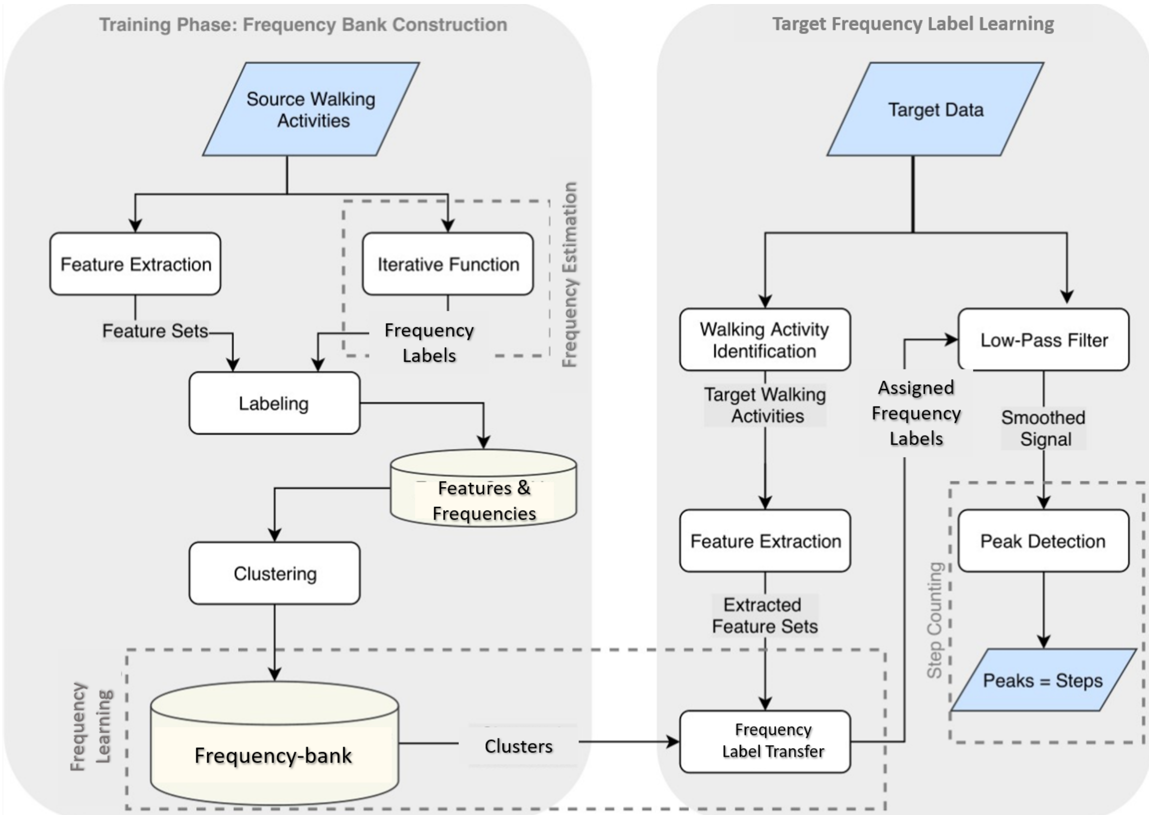


Fig. 3. Step-counting approach in ParaLabel. The process consists of two main phases including *training* and *label transfer*.

Euclidean distance in feature space without manually fine-tuning the parameters in the new setting. We propose to smooth the noisy input signal using a low-pass filter and detect the steps by identifying the peaks in the magnitude of the smoothed signal. We evaluate ParaLabel using a dataset of accelerometer signals collected from three smartphones on the chest, the dominant wrist, and the left pocket of 15 subjects. We compare the performance of ParaLabel against the Fitbit trackers and state-of-the-art frequency-based step counting algorithms during medium to high intensity walking (e.g., walking on a treadmill at 5.0 $\frac{km}{h}$, and 8.0 $\frac{km}{h}$), low-intensity (e.g., walking on a treadmill at 2.5 $\frac{km}{h}$, and aided walking (e.g., walking with a walker and walking with a shopping cart).

III. ParaLabel Algorithm Design

A. Problem Statement

As mentioned in Section II-A, the step detection task is to identify peaks in the smoothed magnitude of the tri-axial accelerometer signal such that the accuracy of the resulting step-counts is bounded by a given threshold (i.e, 95%). We devise an approach based on a low-pass filter with a variable cut-off frequency parameter to smooth the norm signal prior to peak detection. To achieve a high step-counting accuracy, the signal needs to be filtered with appropriate cut-off frequency (i.e., step frequency as noted in Section II-A). Therefore, the main focus in the design of ParaLabel is on an optimization of the filter parameter to guarantee a high accuracy in detecting steps. Figure 3 shows the overall process of algorithm design in ParaLabel.

Therefore, ParaLabel aims to estimate the filter parameter (step frequency) of a given signal which is recorded in an arbitrary setting (i.e., target), based on the knowledge from a different but similar setting (source). Based on the common analogy in machine learning, we refer to the previous domain with known filter parameters as the *source* and to the new setting as the *target*. The target data contains signals collected with a new user. The source includes data from a different set of users than that of the target. We assume that the signal recorded in the target setting, also referred to as the target signal, S_t, can be segmented into p smaller signals, $\{w_1, \ldots, w_p\}$, each of which represents walking with a particular step frequency. We assume that the source, S_s, consists of n walking activities $\{a_1, a_2, ..., a_n\}$ with known step frequencies $\{f_1, f_2, ..., f_n\}$. The problem is to learn the step frequency for the target segments autonomously based on the training data available in the source.

B. Problem Solution

We propose a transfer learning approach that transfers the step frequencies from a frequency-bank which is built based on the knowledge from the source dataset, to the target signal. The proposed approach learns step frequencies for each segment of the target signal based on two main phases: (i) Training phase, in which step frequencies are computed for the activities in the source setting; and (ii) Target frequency learning, which assigns step frequencies from the source dataset to the target segments.

1) Training Phase in Source Setting: We construct a frequency-bank from the source setting during an off-line process. The frequency-bank is a database of features and step frequency labels corresponding to signal segments that represent different walking activities in the source setting. We extract a set of representative features and estimate the step frequency values given the step-counts, for each walking activity in the source. Therefore, a set of observations in feature space and a step frequency in the frequency-bank represents each walking activity in the source setting.

Feature extraction: we extract a k-dimensional feature vector, $Fs_i = \{fs_{i1}, fs_{i2}, ..., fs_{ik}\}$ from each activity a_i, in the source dataset. The feature vector is the mean value of the feature sets extracted using a sliding window of size 2 seconds (200 samples) with a 50% overlap. As shown in Table I, the extracted features include time- and frequency-domain features such as amplitude, median, mean, variance, energy, and fundamental frequency of the signal that represent the pattern, and intensity of the walking activities. As discussed previously fundamental frequency of a periodic signal recorded while walking also represents the step frequency. Moreover, prior research has shown that the time-domain features such as mean, amplitude, median, and variance are effective in physical activity recognition using wearable motion sensor data such as accelerometer sensors [32], [33].

TABLE I
EXTRACTED FEATURES FROM EACH ACCELEROMETER DATA SEGMENT

Feature	Description
AMP	Amplitude of Signal Segment
MED	Median of the Signal
MNVALUE	Mean of the Signal
VAR	Variance
FRQ	Fundamental Frequency
ENG	Energy of the Signal

Step frequency estimation: we estimate the step frequency of each walking activity a_i in the source dataset, given the number of steps taken during that activity. To this end, we compute the step frequency by iterating over all the possible step frequency values (with $0.1Hz$ increments) between 2Hz to 12Hz. We select those frequency values that result in more than a given lower bound (e.g., 90%) on the accuracy of step-counting matching between the steps taken and peaks extracted from the smoothed magnitude signal.

2) Frequency-Label Learning in the Target Setting: The goal of frequency-label learning phase is to assign frequency values to a segment of signal such that the error of assignment is minimized. We estimate the error of assigning f_i label from a_i in frequency-bank to walking segment w_j in the target based on Euclidean distance between the representative features. The error, therefore, is given by:

$$\epsilon_{ij} = \sqrt{\sum_{k=1}^{n} (fs_{ki} - ft_{ki})^2} \quad (1)$$

where k represents the dimension of the feature vector. Symbol n refers to the number of walking activities in the source dataset, therefore, the number of the feature sets in the frequency-bank. Symbols fs_{ki} and ft_{ki} show the k_th element of the feature vector corresponding to the signal segment i from the source and signal segment j from the target domain, respectively.

The process of label learning in the target setting includes three steps: (i) walking activity identification by segmenting the target signal; (ii) step frequency transfer from frequency-bank to the target segments; and (ii) step detection.

Walking activity identification: because gait speed changes over time, we assume that the target data does not necessarily contain a single-walking activity. We extract a set of features which are representative of the intensity and pattern of the walking, from a sliding window over the magnitude signal in the target setting. We classify each window into the type of walking such as walking with a walker, walking with a shopping cart, walking on the treadmill at 2.5 $\frac{km}{h}$, walking on the treadmill at 5.0 $\frac{km}{h}$, and walking on the treadmill at 8.0 $\frac{km}{h}$.

Algorithm 1: PARALABEL STEP-COUNTING ALGORITHM

Input: $S_t = \{w_1, w_2, ..., w_p\}$ walking segments in the target setting, feature sets $S_s = \{Fs_1, Fs_2, ..., Fs_n\}$ in frequency-Bank, estimated step frequency labels $F = f_1, f_2, ..., f_n$

```
min ← +Inf;
foreach segment w in S_t do
    c ← FeatureExtraction(w);
    foreach entry fs in S_s do
        if Distance(c, fs) < min then
            min ← Distance(c, fs);
        freq ← fs.f ;
    end
end
Output: freq /* the transferred step
    frequency labels */
```

Step frequency transfer: we develop a transfer learning approach, shown in Algorithm 1, which performs 5-nearest-neighbor classification to transfer the step frequency labels from the frequency-bank to the target segments. ParaLabel computes the Euclidean distance between the target segments and the frequency-bank entries in the feature space. The function $Distance(.,.)$ computes the Euclidean distance between two k-dimensional feature vectors. We label each target segment by the step frequency of the most similar entry from the frequency-bank.

Step detection: ParaLabel applies a low-pass filter to each segment of the target signal to remove the high-frequency noises. The cut-off frequency of the low-pass filter is set to the step frequency of the signal segment which is learned from the previous step. The peaks of the smoothed signal represent the steps while walking. More information on the design and implementation of the low-pass filter of the proposed algorithm is provided in Section IV-B.

The time complexity of Algorithm 1 is O($n \times p$), where n and p denote the number of entries in the frequency-bank in the source setting, and the number of signal segments in the target setting, respectively. While this approach is feasible for small source datasets and short target signals, it may limit the scalability of ParaLabel for deployment with large datasets. Therefore, we also propose an extension to ParaLabel, called Augmented ParaLabel that improves the time complexity of the original algorithm.

The design process of Augmented ParaLabel is shown in Algorithm 2. We cluster the frequency-bank entries using the K-means clustering algorithm [34], [35]. Each cluster is represented by the centroid which is a mean value of the member of the cluster according to their representation in the feature space. Augmented ParaLabel, classifies each new segment in the target, into an existing cluster with the closest centroid to the feature vector representing that segment. The closeness is computed as the Euclidean distance between the centroid and feature vector. The algorithm assigns the frequency label of the closest member of the chosen cluster from the previous step, as the label for the new signal segment.

Algorithm 2: AUGMENTED PARALABEL STEP-COUNTING ALGORITHM

```
Input: S_t = {w_1, w_2, ..., w_p} walking segments in the
       target setting, feature sets
       S_s = {Fs_1, Fs_2, ..., Fs_n} in frequency-Bank,
       estimated step frequency labels
       F = f_1, f_2, ..., f_n
minDist ← +Inf; /* the minimum distance
   among the pairs of the instances
   in the feature set of the target
   and source sensors                    */
C ← Cluster(FrequencyBank)
foreach walking segment w in S_t do
    c ← FeatureExtraction(w);
    Ĉ ← ClusterSelection(c, C)
    foreach entry fs in Ĉ do
        if Distance(c, fs) < minDist then
        minDist ← Distance(c, fs)
        freq ← fs.f ;
    end
end
Output: freq /* the dataset of the
   target data unit with estimated
   step frequency labels                 */
Function ClusterSelection(f, C)
    maxCor ← −Inf; /* the maximum
       correlation value between the f
       and the clusters' feature set in
       set of clusters C                 */
    foreach cluster c in C do
        if Distance(f, c) > maxCor then
        maxCore ← Distance(f, c);
        c_selected ← c ;
    end
    return c_selected
```

IV. VALIDATION APPROACH

This section describes the experimental setup, dataset, and methodologies used to validate the ParaLabel step-counting algorithm.

A. Experimental Design

We conducted a study with healthy adults who performed multiple physical activities while wearing various data collection wearable devices.

1) Recruitment and Participant: We recruited 15 subjects, seven females and eight males, between 21 and 31 years old to participate in the pre defined protocol. Washington State University (WSU) Institutional Review Board (IRB) approved the study protocol. Our selection criteria included the absence of conditions that might cause gait abnormalities such as fractures and broken bones, as well as neurological impairments. The protocol exclusion criteria included inability to walk on a treadmill, inability to walk with an assisting device, and inability to perform 30 minutes of light to moderate physical activity (MET < 6) with multiple rests in between. We informed all the participants about the goal of the study, methodology, and testing procedure before the data collection. The participants were asked to complete a questionnaire including age, gender, and physical conditions. We recruited the participants through direct contact and advertisement in EECS school at WSU. Table II shows the physical statistics and demographic information of the participants in the study.

2) Collection Devices: Each participant wore three Fitbit trackers including Zip, One, and Flex as well as three Samsung Galaxy smartphones S6 on three different body-locations simultaneously. The Fitbit Zip, One, and Flex were worn on the chest, left pants pocket, and dominant wrist of the participants. The Fitbit trackers were selected as the widely-used off-the-shelf activity trackers and step counting devices to compare against the proposed step counting algorithm based on the ground truth step counts from the video data. We note that the results from a prior study suggested no significant difference in the number of steps recorded by the Fitbit trackers Zip, One, and Flex when a subject wore them on the same location of the body [36]. The smartphones were placed on the same location as the Fitbit trackers for a fair comparison. The smartphones were used to collected tri-axial raw accelerometer data with sampling frequency @100Hz during the experiment using a custom Android 5.0.2 application. We also recorded the feet

TABLE II
PHYSICAL AND DEMOGRAPHIC CHARACTERISTICS OF PARTICIPANTS.

Variable	All Subjects (15)	Female (7)	Male (8)
Age (y)	21-31	24-26	23-31
Height (cm)	155-189	155-178	175-189
Body Mass (kg)	47-86	47-75	65-86
BMI (kgm-2)	19-24.8	19-24.8	20.1-24.8

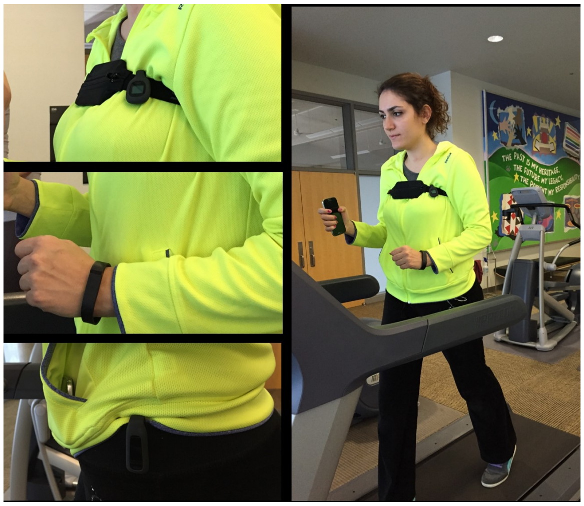


Fig. 4. Placement of the Fitbit trackers and Galaxy smartphones during the experiment on one of the participants.

of the participants during the experiment using a camera. The ground truth step counts were manually counted by looking at the recorded videos from the feet of the participants during the experiment.

3) Physical Activities: We included two categories of physical activities in this study: (1) walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, and walking on a treadmill at 8.0 $\frac{km}{h}$; (2) walking with a shopping cart, and walking with a walker. We asked the participants to perform each activity for 5 minutes.

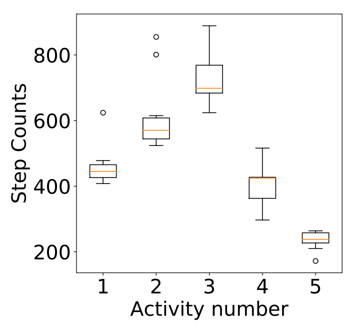


Fig. 5. The actual number of steps taken by participants during different walking activities. Activities 1 to 5 refer to walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a shopping cart, and walking with a walker, respectively.

Figure 5 compares the number of steps during each activity where activities labeled 1 to 5 on the x-axis refer to walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a shopping cart, and walking with a walker, respectively. The average step-counts for the users were 464.7, 641.0, 777.5, 242.5, and 430.1 with standard deviations of 57.4, 118.8, 81.4, 16.3, 71.3, during activities 1 to 5, respectively.

B. Comparison Method and Implementation Details

1) Activity Type Classification: To identify different walking activities, described in Section III-B, we used an ensemble tree classifier that took features extracted from a sliding window of size 2 seconds (e.g., 200 samples) and overlap of 50% over magnitude signal as input and produced the walking activity class. We labeled the windows with 6 types of walking including walking with the treadmill at 2.5 km, 5.0 $\frac{km}{h}$, and 8.0 $\frac{km}{h}$, walking with a walker, walking with a shopping cart, and not walking. The signals that were identified as walking were used for further analysis, namely step-counting.

2) Low-Pass Filter: Filtering is a traditional method of improving response in noisy conditions [37]. Low-pass filters are designed based on two parameters, order and cut-off frequency. The order of a filter represents the complexity of its transfer function; The less order value the more efficient the filter would be in terms of computational complexity. On the other hand, the cut-off frequency of a filter represents the threshold for eliminating higher frequency components of the signal [38]. Out of several implementations of the low-pass filter, we selected the *Butter-worth* implementation for our study [39], [40]. We implemented a low-pass filter with order 10 and a variable cut-off frequency parameter. We computed the cut-off frequency of the filtered as:

$$cf = \frac{f \times 2}{F_s} \tag{2}$$

where f denotes the step frequency and F_s refers to the sampling rate of data collection device. Each signal segment in the target setting was filtered by the cut-off frequency values estimated based on the step frequency numbers from the frequency-bank. As soon as the signal is filtered based on a given filter parameter set (cut-off frequency and order), each peak on the filtered signal represents one step taken by the user while walking.

3) Comparison Approach: We compared the performance of the proposed algorithm against three other methods. (1) The results obtained with a commercially available step-counter (i.e., Fitbit). We chose Fitbit trackers in our study because of their popularity and that they are shown to be among the most accurate step-counting devices [3], [41] (2) A conventional step-counting algorithm, which uses the dominant frequency of the signal as the filter parameter. The dominant frequency is computed based on the Discrete Fourier Transform of the target signal. We refer to this approach as *frequency-based*. Implementation details of the frequency-based approach are described in Section II-A. (3) A conventional step-counting algorithm that uses the median point of the frequencies in Figure 7 with respect to the activity type and location of the sensor, as the cut-off frequency value of the low-pass filter prior to peak detection. We refer to this method as the *Median*. Although, computation of the median frequency requires pre-processing of a training dataset, the purpose to implement such a method is to compare ParaLabel against the case when we assign an average filter parameter to each type of walking.

4) Performance Metrics: We computed the absolute error (error rate) of step-counting to demonstrate the need for developing new step-counting algorithms that take into account various walking activities. The development of ParaLabel is motivated by the results obtained in this analysis (see the Results section). The absolute error, ϵ, can be computed as:

$$\epsilon = 1 - \frac{Steps_{estimated}}{Steps_{actual}} \quad (3)$$

We computed the accuracy of step-counting, α, as the ratio of correctly counted steps by an algorithm to the ground truth step-counts. This accuracy measure is given by:

$$\alpha = \frac{Steps_{estimated}}{Steps_{actual}} \quad (4)$$

We compute the CoV for each algorithm over the subjects based on the equation below.

$$CoV = \frac{\sigma}{\mu} \quad (5)$$

where σ and μ are respectively the standard deviation and mean of the F1-Score in lying posture detection over different folds.

5) Complexity Analysis & Design Specifications: Constructing the source filter bank and training the ensemble tree classifier on the signal for walking activity detection are performed offline on a server computer. These offline data processing and system design efforts do not affect the complexity of ParaLabel in the target domain.

We use a moving window (e.g., of 2 seconds length) to segment the input signal into distinct walking types (e.g., walking at 2.5 $\frac{km}{h}$). To detect the walking types, the features in Table I are extracted from each window and classified into a walking class using an ensemble tree classification model. To transfer the filters from the filter bank to the target signal segments, for each segment of the target signal, we compute the Euclidean distance between its representative feature vector and all the filter bank entries. Since ParaLabel extracts a constant number of the features, both feature extractions and distance computation run in linear time.

As described in the methodology sections of the paper, the entries in the filter bank are labeled with optimal filter frequency values. To detect steps in the target domain, we use the label of the entry with the smallest distance as our filter parameter. To reduce the complexity of filtering, we use a Butterworth filter of order 2 to implement the low-pass filter. We detect the peaks on the magnitude of the smoothed signal using a peak detection algorithm. In general, step counting methods based on peak-detection are shown to be computationally simple [42]. The run-time complexity of filtering and peak detection is a function of the segment length. It is reasonable to assume that the signal length is bounded by a constant value, $Length_{max}$. Therefore, we can assume that the algorithmic computations for each signal segment are limited to a constant upper-bound.

We also assume that the input signal is partitioned into m walking segments and there are n entries in the filter bank. Therefore the time complexity of the entire algorithm is $O(m \times n)$. Moreover, we reduce the computational complexity of ParaLabel by clustering the filter bank entries offline in the source setting when reducing the entries significantly by replacing a group of features with the centroid of their corresponding cluster.

We implemented our algorithms using MATLAB signal processing toolbox on a Laptop computer with a 16 GB RAM and an Intel Core i7 processor running at 2.5 GHz. As discussed previously, the time complexity of ParaLabel and ParaLabel+ running in the target setting is low enough to be implemented on common wearable devices such as a smartphone or a smartwatch. For example, an Apple iPhone 11 consists of two 5.65 GHz Lightning and four 1.8 GHz Thunder CPU cores, a storage capacity of up to 256 GB, and a 6 GB of RAM. Also, a Samsung Galaxy S20 contains two Mongoose M5, two Cortex-A76, and four Coretx-A55 processing units, 128 GB of storage capacity, and up to 12 GB of RAM. As another example, Apple Watch 5 has an Apple S5 processor with 32 GB of RAM, and a Samsung Galaxy watch has a Dual-core 1.15 GHz processor with a 4 GB of storage. As a future direction to this study, we will investigate the strengths and limitations of deploying ParaLabel for in-the-field step counting to compare its performance against existing algorithms in terms of step counting accuracy and technology acceptance.

V. RESULTS

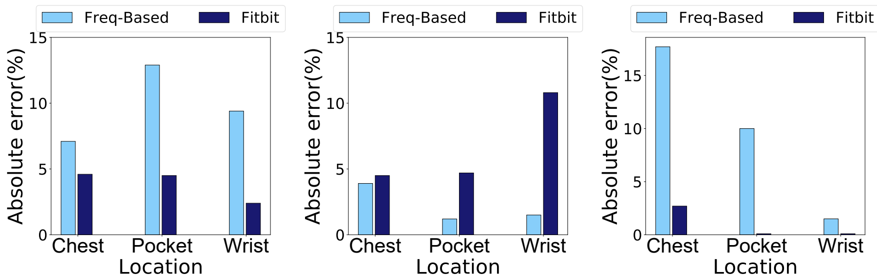
This section presents results on the performance of the proposed reliable step-counting approach. In the first subsection, we present results that derive our motivation for developing an accurate step-counting algorithm for low-intensity and aided walking activities. In the second subsection, we investigate the results of the estimated step frequencies in the source setting. The third subsection contains the results on the walking activity identification in the target setting. The last subsection of the results provides the evaluation results of the proposed step-counting algorithm, ParaLabel, and compare it to the state-of-the-art step-counting approaches.

A. Performance of state-of-the-art

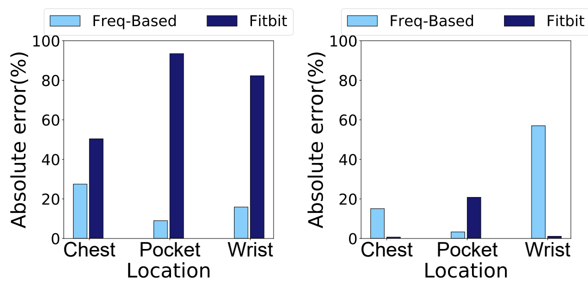
Figure 6 compares the step-counting error of a Fitbit tracker worn on the chest, and the frequency-based algorithm. For the frequency-based step-counting, we used the accelerometer data collected using a smartphone worn approximately on the same body-location as the Fitbit tracker.

The Fitbit tracker miscounted 4.6%, 1.4%, 5.7% of the steps during walking on a treadmill at 8.0 $\frac{km}{h}$, 5.0 $\frac{km}{h}$, and 2.5 $\frac{km}{h}$, respectively. The absolute errors of the tracker increase to 10.1% and 83.8% for walking with a shopping cart and a walker as shown in Figure 6. The average error of the Fitbit tracker during moderate-to-high-intensity walking activities is 2.9%. However, it grows to 33.2% and 46.9% during low-intensity and aided walking, respectively.

Compared to the Fitbit, the frequency-based algorithm achieved a better performance during walking with a walker. Specifically, the frequency-based algorithm obtained an absolute error of 19.4% during walking with a walker. However, the algorithm did not maintain a high accuracy during walking on a treadmill and walking with a shopping cart. Specifically, the algorithm achieved 16.6%, 18.9%, 11.5%, 67.5% error in counting the steps during walking on a treadmill at 8.0 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 2.5 $\frac{km}{h}$, and walking with a shopping cart, respectively.



(a) Walking at 2.5 $\frac{km}{h}$ (b) Walking at 5.0 $\frac{km}{h}$ (c) Walking at 8.0 $\frac{km}{h}$



(d) Walking with a walker (e) walking with a shopping cart

Fig. 6. The step-counting error of different approaches: Models used in an activity tracker (i.e., Fitbit) and a machine-learning-based model (i.e., frequency-based step-counting).

These results necessitate the design of a reliable and re-configurable step-counting algorithm that maintains the high accuracy level of the current trackers in moderate-to-high-intensity walking, while achieving a high accuracy in low-intensity and aided walking activities as well. The remainder of this section presents our evaluation of ParaLabel.

B. Step Frequency Estimation

Given a signal segment from source dataset collected from a participant while walking, We computed the step frequency by a brute-force search process and the ground-truth step counts. Since human step frequency can range from $0.1Hz$ when standing still to $12Hz$ for vigorous running. We applied low-pass filters with cut-off frequency values from $0.1Hz$ to $12Hz$ with a step size of $0.01Hz$ and recorded the accuracy of detecting steps in the smoothed signal. Figure 7 shows the distribution of the cut-off frequency values that resulted in step counting accuracy of more than 95% for different walking activities and sensor locations. In each sub-figure, activity numbers 1 to 5 refer to walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a walker and walker with a shopping cart, respectively.

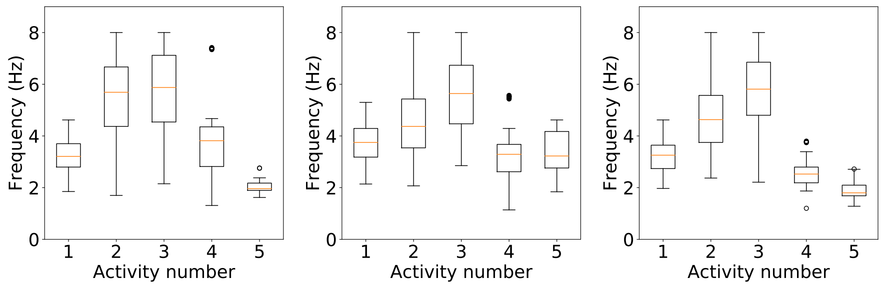
The estimated step frequencies exhibit a similar range during the same activity for all three locations. For the chest sensor, mean value and range of estimated step frequencies are 4.04Hz (2.5Hz to 5.75Hz), 5.36Hz (3Hz to 8.25Hz), 7.01Hz (4.25Hz to 11Hz), 2.86Hz (2Hz to 4Hz) and 4.02Hz (2Hz to 6.5Hz) for walking activities 1 to 5, respectively. For pocket sensors, mean value and range of estimated step frequencies are 4.26Hz (2.5Hz to 6.5Hz), 5.86Hz (3Hz to 9.75Hz), 6.56Hz (3.75Hz to 9Hz), 2.21Hz (1.75Hz to 3Hz) and 3.81Hz (2.5Hz to 5.75Hz) for walking activities 1 to 5,respectively. For wrist sensor, mean value and range of estimated step frequencies are 4.25Hz (2.75Hz to 6.25Hz), 5.86Hz (3.5Hz to 9.75Hz), 6.98Hz (4.5Hz to 10.75Hz), 2.10Hz (1.5Hz to 3Hz) and 3.75Hz (2.5Hz to 5.5Hz) for walking activities 1 to 5, respectively.

C. Target Walking Activity Identification

We apply an ensemble-tree classifier on the magnitude signal to detect the type of walking activity. The red squares show the identified walking activities in the sample target signal. The ensemble-tree classifier achieves high accuracy of 93.3%, 90.7%, and 88.6% in detecting the type of walking from the data collected by the sensor on the chest, pocket, and wrist, respectively. The reason for the decline in the accuracy of the wrist sensor is that subjects might exhibit extra hand movements with different patterns that are a source of the noise to the collected signal. The sensor in the pocket is not as accurate as of the sensor on the chest, since, the smartphone is free in the pocket while it is firmly attached to the chest.

D. Step Detection Performance Evaluation

In this section, we compare the accuracy and running time of the ParaLabel and augmented ParaLabel in the target setting. Then, we evaluate the accuracy of the ParaLabel against state-of-the-art approaches including frequency-based, and Fitbit trackers mode for the chest, dominant wrist, and left pocket locations. The step-counting accuracies are reported for each walking activity including walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a shopping cart, and walking with a walker. The values reported in the results section are averaged over

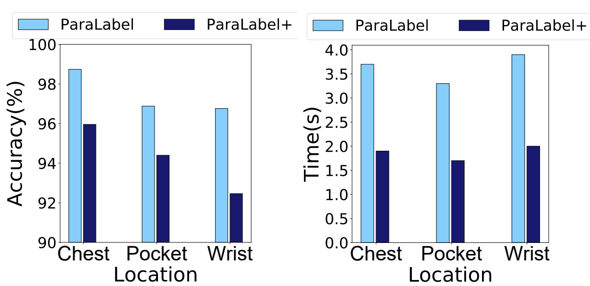


(a) Chest (b) Left pocket (c) Dominant wrist

Fig. 7. Estimated step frequency range during different walking activities: (1) treadmill walking at 2.5 $\frac{km}{h}$, (2) treadmill walking at 5.0 $\frac{km}{h}$, (3) treadmill walking at 8.0 $\frac{km}{h}$, (4) walking with a walker, and (5) walking with a shopping cart for a sensor on chest (a), pocket (b) and wrist (c).

all the subjects and all the analyses are done by the leave-one-subject-out validation technique.

1) Performance of Augmented ParaLabel: We compare the accuracy of step-counting inFigure 8a and the running time in Figure 8b, of the ParaLabel and augmented ParaLabel algorithms. We refer to the augmented ParaLabel as ParaLabel+ in Figure 8. ParaLabel+ shows a negligible accuracy drop while it decreases the running time by 33.1% on average, compared to ParaLabel. We expect the gap between the running time of the two algorithms to increase as the source dataset becomes larger, which means the frequency-bank contains more entries. Therefore, we report the result of both ParaLable and augmented ParaLabel methods against state-of-the-art step-counting approaches, in the remainder of the results section.



(a) Accuracy (b) Execution Time

Fig. 8. Average step-counting accuracy and running time of the ParaLabel and ParaLabel+ algorithms for the chest, left pocket and dominant wrist sensor locations.

2) Comparison against State-of-the-art: Table III shows the average accuracy and coefficient of variation for the proposed algorithms ParaLabel, ParaLabel+, and comparison models, Fitbit, filtering with the median frequency per walking activity (Median), and filtering with an adaptive frequency based on the signal (frequency-based) during different walking activities when the sensor was worn on the chest, dominant wrist, and in the left pocket.

Chest: Fitbit algorithm correctly identified 95.4%, 95.5%, 97.3%, and 99.3% of the steps taken during treadmill activities walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, and walking with a shopping cart, respectively. The step-counting accuracy of Fitbit drops to 49.6% when walking with a walker. The proposed algorithm, ParaLabel, could maintain the high performance across all the activities dependent of the intensity and pattern; it achieved 99.1%, 98.4%, 99.0%, 99.8%, and 97.3% during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a shopping cart and walking with a walker. Overall, Augmented ParaLabel achieves higher accuracy than the Fitbit algorithm and slightly lower accuracy than the ParaLabel. The frequency-based algorithm obtains competitive results during the walking on a treadmill at 2.5 $\frac{km}{h}$, and walking on a treadmill at 8.0 $\frac{km}{h}$, while its accuracy drops to 72.5%-87.1% for the other activities. While setting the filter parameter to the median frequency for each activity drops the accuracy significantly to a range of 34.6%-87.1%. These results demonstrate that pre-defining a frequency for each activity obtains poor performance because subjects might perform the same activity with a different pattern and intensity. While adaptive filtering frequency for the signal of activities from a different intensity might not be an optimal solution to recognize the steps as well.

As shown in Table III, ParaLabel obtains the lowest accuracy variation to mean ratio comparing to the state-of-the-art methods. ParaLabel achieves 0.00, 0.05, 0.03, 0.00, 0.08 CoV during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$, walking on a treadmill at 8.0 $\frac{km}{h}$, walking with a shopping cart, and walking with a walker. These results demonstrate the consistency of the results across all 15 participants in the experiments.

Left pocket: Fitbit trackers achieve $> 95.3\%$ step counting accuracy during the treadmill activities, while their accuracy drops to 79.2% and 6.5% during walking with a shopping cart and walking with a walker. The frequency-based algorithm obtains $> 96.7\%$ step counting accuracy for all the activities except walking with a walker in which the accuracy slightly drops to 91.0%. Predefined filtering frequency (median value) for each activity decreases the step counting accuracy to 59.6%, 71.8%, and 45.0% for walking on the treadmill at 2.5 $\frac{km}{h}$, 5.0 $\frac{km}{h}$, and 8.0 $\frac{km}{h}$, while increases to 88.5%, and 57.2% for walking with a shopping cart and walking with a walker. On the other hand, the promising results of the frequency-based method compared to the Fitbit tracker is

TABLE III
COMPARISON BETWEEN THE MEAN ACCURACY AND COEFFICIENT OF VARIATION OF STEP DETECTION ACCURACY OVER DIFFERENT PARTICIPANTS FOR VARIOUS WALKING ACTIVITIES AND BODY LOCATIONS.

Location	Activity	Mean Value (%)					Coefficient of Variation				
		Median	Fitbit	Freq-Based	ParaLabel	ParaLabel+	Median	Fitbit	Freq-Based	ParaLabel	ParaLabel+
Chest	2.5 $\frac{km}{h}$ walking	50.8	95.4	92.9	99.1	96.0	0.48	0.04	0.30	0.00	0.21
	5.0 $\frac{km}{h}$ walking	70.3	95.5	87.1	98.4	97.5	0.15	0.06	0.24	0.05	0.11
	8.0 $\frac{km}{h}$ walking	34.6	97.3	90.6	99.0	95.9	0.32	0.10	0.32	0.03	0.15
	Pushing a cart	68.9	99.3	84.9	99.8	99.0	0.39	0.24	0.46	0.00	0.05
	Using a walker	87.1	49.6	72.5	97.3	91.4	0.13	0.96	0.69	0.08	0.12
Left pocket	2.5 $\frac{km}{h}$ walking	59.6	97.6	96.1	99.4	97.0	0.32	0.10	0.03	0.01	0.11
	5.0 $\frac{km}{h}$ walking	71.8	95.3	98.8	98.5	96.2	0.10	0.09	0.11	0.00	0.08
	8.0 $\frac{km}{h}$ walking	45.0	99.2	98.5	99.9	96.4	0.30	0.13	0.04	0.00	0.06
	Pushing a cart	88.5	79.2	96.7	98.2	92.1	0.11	0.36	0.29	0.07	0.20
	Using a walker	57.2	6.5	91.0	95.4	90.3	0.16	1.90	0.33	0.10	0.23
Dominant wrist	2.5 $\frac{km}{h}$ walking	66.8	94.1	82.3	96.3	93.9	0.42	0.12	0.35	0.07	0.20
	5.0 $\frac{km}{h}$ walking	78.4	99.9	90.0	99.1	94.0	0.13	0.01	0.15	0.05	0.09
	8.0 $\frac{km}{h}$ walking	43.8	99.7	98.5	99.9	96.8	0.40	0.05	0.15	0.05	0.21
	Pushing a cart	72.9	98.9	43.0	99.2	95.5	0.17	0.06	1.21	0.03	0.10
	Using a walker	67.1	17.7	84.1	89.3	82.1	0.12	1.87	0.23	0.01	0.23

because the accelerometer signal readings from left pocket represent the motion of the left foot while walking, therefore the frequency component of such signal highly correlates with the step frequency of the users. ParaLabel improves the performance even more to $> 95.4\%$ step counting accuracy during the treadmill activities, walking with a shopping cart, and walking with a walker. In particular , ParaLabel achieves 99.4%, 99.4%, 99.4%, 99.4%, and 99.4% during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, walking while pushing a walker and walking with a walker, respectively.

Dominant wrist: the step-counting accuracies of the Fitbit tracker are 94.1%, 99.9% and 99.7% during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, respectively, because the hands are free to swing alongside the body while walking, therefore, represent the step frequency. The frequency-based algorithm could detect $> 90.0\%$ of the steps taken during walking on a treadmill at $5.0\frac{km}{h}$, and $8.0\frac{km}{h}$, however, the step counting accuracy drops to 82.3% while walking on a treadmill at a slower pace of $2.5\frac{km}{h}$. We note that the accuracy of the frequency-based method drops significantly to 43.0% while walking with a shopping cart because both hands are holding the cart, therefore the frequency components of the signal from the dominant wrist are not an accurate representation of the step frequency. Low step counting accuracy (e.g., 43.8%-78.4%) when using predefined frequencies based on the type of walking in the Median method demonstrates the cross-subject variations in the pattern of performing the same walking activity. On the other hand, ParaLabel achieves 99.4%, 98.5% and 99.9% step-counting accuracy during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, respectively. Moreover, it improves the accuracy of the frequency-based algorithm and Fitbit model to 89.3% during walking with a walker and 99.9% during walking with a shopping cart, respectively. However, the accuracy drops to 17.7% during walking with a walker for the Fitbit tracker on the dominant wrist.

VI. CONCLUSIONS

We proposed ParaLabel, a step-counting algorithm based on peak detection techniques, the frequency component of the signal and other frequency, and time-domain features extracted from the input signal. The proposed approach exhibits kNN classification to transfer the frequency labels from frequency-bank to the target. ParaLabel achieved high step counting accuracy during the walking activities regardless of the intensity and pattern of the walking. We evaluated the proposed algorithm using data collected from three locations including the chest, the left pocket, and the dominant wrist on the 15 users. ParaLabel achieved step counting accuracy of $96.3\% - 99.4\%$, $98.4\% - 99.1\%$, $99.8\% - 99.9\%$, $98.2\% - 99.9\%$, and $89.3\% - 97.3\%$ during walking on a treadmill at 2.5 $\frac{km}{h}$, walking on a treadmill at 5.0 $\frac{km}{h}$ and walking on a treadmill at 8.0 $\frac{km}{h}$, walking while pushing a walker and walking with a walker. Moreover, ParaLabel achieved the lowest variation to mean accuracy ratio (ranged from 0.00 for walking on the treadmill at 2.5 $\frac{km}{h}$ to 0.10 for walking with a walker) comparing to the state-of-the-art. These results demonstrate the consistency of the performance of the proposed algorithm across all 15 participants in the experiments. We further, proposed an upgraded version of ParaLabel to reduce the time and computational complexity of the algorithm by clustering the frequency-bank prior to label transfer. We were able to maintain high accuracy while decreasing the running time by 33.1%.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation under grant CNS-1750679. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

REFERENCES

[1] M. L. Dontje, M. H. van der Wal, R. P. Stolk, J. Brügemann, T. Jaarsma, P. E. Wijtvliet, C. P. van der Schans, and M. H. de Greef, "Daily physical activity in stable heart failure patients," *Journal of Cardiovascular Nursing*, vol. 29, no. 3, pp. 218–226, 2014.

[2] G. Mammen, S. Gardiner, A. Senthinathan, L. McClemont, M. Stone, and G. Faulkner, "Is this bit fit? measuring the quality of the fitbit step-counter," *Health Fit J Can*, vol. 5, no. 4, pp. 30–9, 2012.

[3] J.-M. Lee, "Validity of consumer-based physical activity monitors and calibration of smartphone for prediction of physical activity energy expenditure," 2013.

[4] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 2013, pp. 225–234.

[5] Y. Huang, H. Zheng, C. Nugent, P. McCullagh, N. Black, W. Burns, M. A. Tully, and S. M. McDonough, "An orientation free adaptive step detection algorithm using a smart phone in physical activity monitoring," *Health and Technology*, vol. 2, no. 4, pp. 249–258, 2012.

[6] S. Jayalath, N. Abhayasinghe, and I. Murray, "A gyroscope based accurate pedometer algorithm," in *International Conference on Indoor Positioning and Indoor Navigation*, vol. 28, 2013, p. 31st.

[7] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic step detection in the accelerometer signal," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*. Springer, 2007, pp. 80–85.

[8] M. E. Micó-Amigo, I. Kingma, E. Ainsworth, S. Walgaard, M. Niessen, R. C. van Lummel, and J. H. van Dieën, "A novel accelerometry-based algorithm for the detection of step durations over short episodes of gait in healthy elderly," *Journal of neuroengineering and rehabilitation*, vol. 13, no. 1, p. 1, 2016.

[9] S. M. Korpan, J. L. Schafer, K. C. Wilson, and S. C. Webber, "Effect of actigraph gt3x+ position and algorithm choice on step count accuracy in older adults," *Journal of aging and physical activity*, vol. 23, no. 3, pp. 377–382, 2015.

[10] J. B. Martin, K. M. Krč, E. A. Mitchell, J. J. Eng, and J. W. Noble, "Pedometer accuracy in slow walking older adults," *International journal of therapy and rehabilitation*, vol. 19, no. 7, p. 387, 2012.

[11] X. Yang and B. Huang, "An accurate step detection algorithm using unconstrained smartphones," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. IEEE, 2015, pp. 5682–5687.

[12] G. Bieber, M. Haescher, P. Hanschmann, and D. J. Matthies, "Exploring accelerometer-based step detection by using a wheeled walking frame," in *Proceedings of the 5th International Workshop on Sensor-based Activity Recognition and Interaction*, ser. iWOAR '18. New York, NY, USA: ACM, 2018, pp. 8:1–8:6. [Online]. Available: http://doi.acm.org/10.1145/3266157.3266212

[13] D. Trong Bui, N. D. Nguyen, and G.-M. Jeong, "A robust step detection algorithm and walking distance estimation based on daily wrist activity recognition using a smart band," *Sensors*, vol. 18, no. 7, p. 2034, 2018.

[14] Y. Liu, K. Lu, S. Yan, M. Sun, D. K. Lester, and K. Zhang, "Gait phase varies over velocities," *Gait & posture*, vol. 39, no. 2, pp. 756–760, 2014.

[15] Y. Cho, H. Cho, and C. Kyung, "Design and implementation of practical step detection algorithm for wrist-worn devices," *IEEE Sensors Journal*, vol. 16, no. 21, pp. 7720–7730, Nov 2016.

[16] X. Kang, B. Huang, and G. Qi, "A novel walking detection and step counting algorithm using unconstrained smartphones," *Sensors*, vol. 18, no. 1, p. 297, 2018.

[17] C. Arcidiacono, S. M. Porto, M. Mancino, and G. Cascone", ""a threshold-based algorithm for the development of inertial sensor-based systems to perform real-time cow step counting in free-stall barns"," *"Biosystems Engineering"*, vol. "153", pp. "99 – 109", "2017". [Online]. Available: "http://www.sciencedirect.com/science/article/pii/S1537511016300873"

[18] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar, "Strap-down pedestrian dead-reckoning system," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, 2011, pp. 1–7.

[19] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *Positioning*, vol. 1, no. 08, p. 0, 2004.

[20] A. C. DIrican and S. Aksoy, "Step counting using smartphone accelerometer and fast fourier transform," *Sigma J. Eng. Nat. Sci*, vol. 8, pp. 175–182, 2017.

[21] N. Capurso, T. Song, W. Cheng, J. Yu, and X. Cheng, "An android-based mechanism for energy efficient localization depending on indoor/outdoor context," *IEEE Internet of Things Journal*, vol. 4, no. 2, p. 299, 2017.

[22] B. Huang, G. Qi, X. Yang, L. Zhao, and H. Zou, "Exploiting cyclic features of walking for pedestrian dead reckoning with unconstrained smartphones," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 374–385.

[23] G. Chen, L. Fei, and Y. Zhang, "Pedometer method based on adaptive peak detection algorithm," *J. Chin. Inert. Technol*, vol. 23, pp. 315–321, 2015.

[24] K.-C. Lan, W.-Y. Shih *et al.*, "Using smart-phones and floor plans for indoor location tracking." *IEEE Trans. Human-Machine Systems*, vol. 44, no. 2, pp. 211–221, 2014.

[25] P. Barralon, N. Vuillerme, and N. Noury, "Walk detection with a kinematic sensor: Frequency and wavelet comparison," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006, pp. 1711–1714.

[26] J.-H. Wang, J.-J. Ding, Y. Chen, and H.-H. Chen, "Real time accelerometer-based gait recognition using adaptive windowed wavelet transforms," in *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on*. IEEE, 2012, pp. 591–594.

[27] N.-H. Ho, P. H. Truong, and G.-M. Jeong, "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone," *Sensors*, vol. 16, no. 9, p. 1423, 2016.

[28] A. Mannini and A. M. Sabatini, "A hidden markov model-based technique for gait segmentation using a foot-mounted gyroscope," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 4369–4373.

[29] W. Dargie, "Analysis of time and frequency domain features of accelerometer measurements." in *ICCCN*, vol. 9, 2009, pp. 1–6.

[30] P. Alinia, R. Saeedi, R. Fallahzadeh, A. Rokni, and H. Ghasemzadeh, "A reliable and reconfigurable signal processing framework for estimation of metabolic equivalent of task in wearable sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 5, pp. 842–853, 2016.

[31] P. Alinia, I. Mirzadeh, and H. Ghasemzadeh, "Actilabel: A combinatorial transfer learning framework for activity recognition," *arXiv preprint arXiv:2003.07415*, 2020.

[32] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[33] R. Saeedi, B. Schimert, and H. Ghasemzadeh, "Cost-sensitive feature selection for on-body sensor localization," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 2014, pp. 833–842.

[34] K. Alsabti, S. Ranka, and V. Singh, "An efficient k-means clustering algorithm," 1997.

[35] T. Barton, T. Bruna, and P. Kordik, "Chameleon 2: An improved graph-based clustering algorithm," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 1, pp. 1–27, 2019.

[36] P. Alinia, C. Cain, R. Fallahzadeh, A. Shahrokni, D. Cook, and H. Ghasemzadeh, "How accurate is your activity tracker? a comparative study of step counts in low-intensity physical activities," *JMIR Mhealth Uhealth*, vol. 5, no. 8, p. e106, Aug 2017. [Online]. Available: http://mhealth.jmir.org/2017/8/e106/

[37] A. R. Pratama, R. Hidayat *et al.*, "Smartphone-based pedestrian dead reckoning as an indoor positioning system," in *System Engineering and Technology (ICSET), 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[38] L. Zhongshen, "The design of butterworth lowpass filter based on matlab [j]," *Heilongjiang Electronic Technology*, vol. 3, p. 017, 2003.

[39] B. Yu, D. Gabriel, L. Noble, and K.-N. An, "Estimate of the optimum cutoff frequency for the butterworth low-pass digital filter," *Journal of Applied Biomechanics*, vol. 15, no. 3, pp. 318–329, 1999.

[40] P. Podder, M. M. Hasan, M. R. Islam, and M. Sayeed, "Design and implementation of butterworth, chebyshev-i and elliptic filter for speech signal analysis," *International Journal of Computer Applications*, vol. 98, no. 7, 2014.

[41] T. J. Kooiman, M. L. Dontje, S. R. Sprenger, W. P. Krijnen, C. P. van der Schans, and M. de Groot, "Reliability and validity of ten consumer activity trackers," *BMC sports science, medicine and rehabilitation*, vol. 7, no. 1, p. 24, 2015.

[42] H.-h. Lee, S. Choi, and M.-j. Lee, "Step detection robust against the dynamics of smartphones," *Sensors*, vol. 15, no. 10, pp. 27 230–27 250, 2015.