Unsupervised Detection and Analysis of Changes in Everyday Physical Activity Data

Gina Sprint[a,*], Diane J. Cook[a], Maureen Schmitter-Edgecombe[b]

[a]*School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA*
[b]*Department of Psychology, Washington State University, Pullman, WA*

Abstract

To be written later.

Keywords: Physical activity monitoring, Wearable sensors, Unsupervised learning, Change point detection, Data mining

1. Introduction

In recent years, sensors have become ubiquitous in our everyday lives. Sensors are ambient in the environment, embedded in smartphones, and worn on the body. Data collected from sensors form a time series, where each sample of data is paired with an associated timestamp. This sensor-based time series data is valuable when monitoring human behavior to detect and analyze changes in behavior. Such analysis can be used to detect seasonal variations, new family or job situations, or health events. Analyzing sensor-based time series data can also be used to monitor changes in human behavior as a person makes progress toward a fitness goal. Making a significant lifestyle change often takes weeks or months of establishing new behavior patterns [1], which can be challenging to sustain. Automatically detecting and tracking behavior changes from sensor data can provide a valuable motivating and monitoring tool.

Recently, wearable sensors have increased in popularity as people aspire to be more conscientious of their physical health. Many consumers purchase

*Corresponding author

Email addresses: `gsprint@eecs.wsu.edu` (Gina Sprint), `cook@eecs.wsu.edu` (Diane J. Cook), `schmitter-e@wsu.edu` (Maureen Schmitter-Edgecombe)

a pedometer or wearable fitness device in order to track their physical activity, often in pursuit of a goal such as increasing cardiovascular strength, losing weight, or improving overall health. Physical activity is estimated by pedometers and fitness trackers in terms of the steps taken by the wearer [2]. To track different types of changes in physical activity data, two or more time periods, or windows, of physical activity data can be quantitatively and objectively compared. If the two time windows contain significantly different sensor data then this may indicate a significant behavior change. Existing off-the-shelf change point detection methods are available to detect change in time series data, but the methods do not provide context or *explanation* regarding the detected change. For physical activity data, algorithmic approaches to change detection require additional information about what type of change is detected and its magnitude to potentially report progress to users for motivation and encouragement purposes. Furthermore, existing approaches often do not provide a method for determining if a detected change is *significant*, meaning the magnitude of change is high enough to suspect it likely resulted from a lifestyle alteration. A personalized, data-driven approach to significance testing for fitness tracker users is a necessary feature of physical activity change detection.

Currently, there is no clear consensus regarding which change detection approaches are best for detecting and analyzing changes in physical activity data. Consequently, we aim to formalize the problem of unsupervised physical activity change detection and analysis and address the problem with our Behavior Change Detection (BCD) approach. BCD is a framework that 1) segments time series data into time periods, 2) detects changes between time periods 3) determines significance of the detected changes and 4) analyzes the significant changes. We review recently proposed change detection methods and analyze their appropriateness for BCD.

We demonstrate the approaches on sample FitBit data collected for two weeks from an individual with documented daily activity information [TODO: which data is this? not presently included]. Next, we evaluate the ability of alternative change detection approaches to capture pattern changes in synthetic physical activity data. Finally, we illustrate how change approaches are used to monitor, quantify, and explain behavior differences in Fitbit data collected from older adults who participate in a brain health behavior intervention. We conclude with discussions about the limitations of current approaches and suggestions for continued research on unsupervised sensor-based change detection.

2. Related Work

In the literature, a few studies have aimed to detect change specifically in human behavior patterns. These approaches have quantified change statistically [3, 4], graphically [4, 5, 6], and the change detection algorithms that we incorporate into our method, BCD. [5, 7, 8, 9]. Recently, Merilahti *et al.* [3] extracted features derived from actigraphy data collected for at least one year. Each feature was individually correlated with a component of the Resident Assessment Instrument for insights into how longitudinal changes in actigraphy and functioning are associated. While this approach provides insight into the relationship between wearable sensor data and clinical assessment scores, this study does not directly quantify sensor-based change.

Wang *et al.* [5] introduced another activity-based change detection approach in which passive infrared motion sensors were installed in apartments and utilized to estimate physical activity in the home and time away from home. This data were converted into gray-level co-occurrence matrices for computation of image-based texture features. Their case studies suggest the proposed texture method can detect lifestyle changes, such as knee replacement surgery and recovery. Though the approach does not provide explanation of the detected changes over time, visual inspection of the data is suggested with activity density maps. More recently, Tan *et al.* [6] applied the texture method to data from Fitbit Flex sensors for tracking changes in daily activity patterns for elderly participants. Another approach for activity monitoring is the Permutation-based Change Detection in Activity Routine (PCAR) algorithm [7]. PCAR researchers modeled activity distributions for time windows containing at least one day. Changes between windows were quantified with a probability of change acquired via hypothesis testing.

The change detection algorithms described previously are intended for monitoring human activity behavior. There are several additional approaches that are not specific to activity data, but instead represent generic statistical approaches to detecting changes in time series data. Change point detection, the problem of identifying abrupt changes in time series data [10], constitutes an extensive body of research as there are many applications requiring efficient, effective algorithms for reliably detecting variation. There are many families of change detection algorithms that are suitable for different applications [11], for example: single point or two sample (window), univariate or multivariate, labeled (supervised) or unlabeled (unsupervised), streaming or offline, etc. Algorithms appropriately handling two sample, univariate, and unlabeled data are most relevant to the current study due to their data-driven change score computation and no need for ground truth information. Univariate unsupervised change detection approaches include subspace models and likelihood ratio methods [8]. One particular subgroup of likelihood ratio methods, direct density ratio estimator methods, are used in various applications [12, 13]. Relative Unconstrained Least-Squares Importance Fitting (RuLSIF) [8] is one such approach used to measure the difference between two samples of data surrounding a candidate change point. Other recent change point detection research includes work on multivariate [14, 15] and streaming time series data [11].

The above approaches are effective methods for detecting change between two samples of data; however, they are not explanatory methods as they only identify if two samples are different and do not provide information on how the samples are different. Once a change is detected and determined significant, additional analyses are required to explain the change that occurred. Hido *et al.* [9] formalized this problem as change analysis, a method of analysis beyond change detection to explain the nature of discrepancy. Hido and colleagues' solution to change analysis utilizes supervised machine learning algorithms to identify and describe changes in unsupervised data. Research by Ng and Dash [16] and Yamada *et al.* [10] have also explored methods for detecting and explaining change in time series data.

The aforementioned methods provide several options for change detection and analysis, each with their own suitability for various applications (i.e. univariate vs. multivariate, change significance testing available or not, etc.). In this paper, the appropriateness of 1) RuLSIF [8], 2) texture-based dissimilarity [5], 3) PCAR [7], 4) our proposed adaptation of PCAR for small window sizes (sw-PCAR), and 5) change analysis [9] for unsupervised change detection and analysis in wearable sensor time series data is evaluated.

3. Methods

Physical activity is often defined as any bodily movement by skeletal muscles that results in caloric energy expenditure [17]. Physical activity consists of bouts of movement that are separated by periods of rest. Physical activity bouts are composed of four dimensions [17]:

1. Frequency: the number of bouts of physical activity within a time period, such as a day.

2. Duration: the length of time an individual participates in a single bout.
3. Intensity: the physiological effort associated with a particular type of physical activity bout.
4. Activity type: the kind of exercise performed during the bout.

To add exercise throughout the day, individuals can increase their number of bouts (frequency), increase the length of bouts (duration), increase the intensity of bouts, and vary the type of physical activity performed during the bouts. These four components of physical activity represent four distinct types of changes that can reflect progress towards many different health goals, such as increasing physical activity or consistency in one's daily routine.

We study the problem of detecting and analyzing change in physical activity patterns. More specifically, we introduce methods to determine if a significant change exists between two windows of time series step data. Algorithm 1, BehaviorChangeDetection, outlines this process. Let m denote the number of equal sized time intervals in a day and t_{mins} denote the number of minutes per time interval. For example, if the sampling rate of the wearable sensor device is one reading per minute, $t_{mins} = 1$ minute and $m = 1440$ minutes / t_{mins}. Now, let $D = \{x_1, x_2, ..., x_t, ..., x_m\}$ be one day of time series data where x_t is a scalar number of steps taken at time interval $t = 1, 2, ..., m$. Let $W_i = \{D_i, D_{i+1}, ..., D_n\}$ be a window of n days with $1 \leq i \leq n$. Suppose we have two windows of data, W_i and W_j $(i \leq j)$, sampled from a time series X. For change detection and analysis, a function F computes a change score, $CS = F(W_i, W_j)$ between two windows, W_i and W_j. Furthermore, an aggregate window, $\hat{W}$, represents the average of all days within the window W:

$$\hat{W} = \frac{1}{n} \sum_{i=1}^{n} D_i, D_i \in W \tag{1}$$

We can compare windows of data within time series data X. These windows may represent consecutive times (e.g., days, weeks, months), a baseline window (e.g., the first week) with each subsequent time window, or overlapping windows. Windows W_i and W_j can be formed as subsets of X based on an *offset* denoting the start of W_j as a function of the start of W_i and iteration advancements adv_i and adv_j to move windows W_i and W_j respectively for the next comparison. Two windows can be compared in either baseline or sliding window mode. For a baseline window comparison, the first window is a reference window that occurs at the beginning of the time series (i is initialized to 1) and is used in each comparison, so $adj_i = 0$. All subsequent windows are compared to the baseline window. Thus j is initialized to $1 + \mathit{offset}$ and is subsequently advanced by adv_j. In the case of a sliding window comparison, both windows used for comparison are advanced through the time series data. Typically $adv_i = adv_j$ for consistently spaced comparisons. In Algorithm 1, BehaviorChangeDetection, i is initialized to 0 and j is initialized to offset. In steps 17 and 18, i is advanced to $i + adv_i$ and j is advanced to $j + adv_j$.

Algorithm 1 BehaviorChangeDetection(X, n, offset, adv_i, adv_j)

1: Input: X = time series data
2: Input: n = window length in days
3: Input: offset = number of days separating windows
4: Input: adv_i = number of days to advance the first window
5: Input: adv_j = number of days to advance the second window
6: Output: Change score vector V
7: Initialize: $i = 1$ and $j = 1 + \mathit{offset}$
8: **for each** pair of windows to compare, W_i and W_j of time series X:
9:     $W_i = X[i : i + n]$
10:     $W_j = X[j : i + n]$
11:     Compute $CS = F(W_i, W_j)$
12:     Determine if CS is significant
13:     Identify the type of change that is exhibited
14:         Manual inspection of change
15:         Unsupervised inspection (change analysis)
16:     Append CS to change score vector V
17:     i = i + adv_i
18:     j = j + adv_j
end for
19: **return** Change score vector V

The choice of window size, n, limits the algorithms that can be applied to the data. For example, the PCAR algorithm [7] is designed for longitudinal data comprising several months; consequently sensitivity decreases with small window sizes. For this study, we categorize window size n choices into the following descriptors:

1. Small window (n = 1 day). Suitable for performing day-to-day comparisons (ex: $D_{Monday1}$ compared to $D_{Monday2}$, $D_{Tuesday1}$ compared to

$D_{Tuesday2}$, ...) or aggregate day comparisons (ex: $\hat{W}_1$ compared to $\hat{W}_2$, $\hat{W}_2$ compared to $\hat{W}_3$, ...).

2. Medium window (2 days $\leq n \leq$ 5 days). Suitable for performing weekday-to-weekday (ex: W_1 compared to W_2 where $W_1 = \{D_{Monday1}$, $D_{Tuesday1}$, $D_{Wednesday1}$, $D_{Thursday1}$, $D_{Friday1}\}$ and $W_2 = \{D_{Monday2}$, $D_{Tuesday2}$, $D_{Wednesday2}$, $D_{Thursday2}$, $D_{Friday2}\}$ or weekend-to-weekend comparisons.
3. Large window ($n > 5$ days). Suitable for performing week-to-week or month-to-month comparisons.

3.1. Change Detection Algorithms

In the following sections, we describe five different algorithmic options for the window-based change score function, F. A summary and comparison of the algorithms is listed in Table 1.

3.1.1. RuLSIF

Non-parametric approaches to change point detection include a family of methods comparing the probability distributions of two time series samples to determine the corresponding dissimilarity. A greater difference between the two distributions implies a higher likelihood that change occurred between the two samples. Instead of estimating the probability distributions, their ratio can be estimated and used to detect changes in the underlying probability distributions. Direct density ratio estimation between two windows of time series data is substantially simpler to solve than computing the windows' probability densities independently and then using these to compute the ratio. Unconstrained Least-Squares Importance Fitting (uLSIF) [8] is one such ratio estimation approach that measures the difference between two samples of data surrounding a candidate change point. For this approach, the density ratio between two probability distributions is estimated directly with the Pearson divergence dissimilarity measure. Depending upon the data, the Pearson divergence can be unbounded. Consequently, a modification to uLSIF, relative uLSIF (RuLSIF), utilizes an alpha-relative Pearson divergence to bound the change score by $1/\alpha$, $0 \leq \alpha < 1$ [8].

3.1.2. Texture-based Change Detection

For this approach, two windows of physical activity data, W_i and W_j, are converted into gray-level co-occurrence matrices (GLCM) [5]. Rows in the resulting GLCM correspond to time intervals while columns correspond

Table 1: Window-based change detection algorithms.

Approach	Window size	Preprocessing	Change score	Change significance test
RuLSIF [8]	Any	Optional Hankel matrix [8]	Probability density ratio estimation with Pearson divergence	Threshold learning in supervised applications. N/A for unsupervised applications
Texture-based [5, 6]	Any	Gray-level co-occurrence matrix, texture features	Weighted normalized Euclidean distance	N/A
PCAR [7]	Large	$m \times N$ KL divergence permutation matrices	Count of time intervals with significant changes (proportion of permuted KL distances greater than observed window)	N/A
sw-PCAR	Small, medium	N KL divergence permutation vectors	KL divergence distance	Non-parametric outlier detection based on Boxplot analysis
Virtual classifier [9]	Large	Physical activity features (intraday and interday if window size > 1)	Cross validation prediction accuracy of binary classifier	Hypothesis testing based on prediction accuracy exceeding a threshold

N = number of permutations

to days. Each cell of the GLCM contains normalized step values symbolized by a gray scale value [TODO: possibly add example (see Figure 4 from Tan)]. Next, the texture features of contrast, dissimilarity, homogeneity, angular second moment (ASM) energy, and correlation are computed from each GLCM [18], producing feature vectors T_i and T_j. To compare two windows W_i and W_j for changes, a weighted normalized Euclidean distance measure is used as a change score to quantify the differences between the corresponding feature vectors T_i and T_j. The smaller the Euclidean distance between these two vectors, the more similar the two windows of data are. The texture-based approach can operate on small or large window sizes; however, the method lends itself more appropriately to large window sizes (Wang *et al.* [5] used window size of one month).

3.1.3. PCAR

PCAR utilizes smart home sensor data to detect changes in behavioral routines [7]. This approach assumes that an activity recognition algorithm [19] is available to label the sensor data with corresponding activity names $A = \{A_1, A_2, ..., A_a\}$. The algorithm is based on the notion of an activity curve C, a compilation of m probability distributions $R_t = \{d_{t,1}, d_{t,2}, ..., d_{t,a}\}$ of a activities per time interval t in a day (t = 1, 2, ..., l, ..., m). The l^{th} element $d_{t,l}$ represents the probability of activity A_l during time interval t. Windows of time spanning multiple days are averaged into an aggregate activity curve $\hat{A}$. To compute a change score CS between two aggregated activity curves $\hat{A}_i$ corresponding to window W_i and $\hat{A}_j$ corresponding to window W_j, the two activity curves are first maximally aligned with dynamic time warping (DTW). Next, the symmetric Kullback-Leibler (KL) divergence is used to compute the distance between each pair of activity distributions in A_i and A_j at time interval t [7]:

$$KL_{symmetric} = KL(D_i, D_j) + KL(D_j, D_i) \tag{2}$$

where

$$KL(D_i, D_j) = \sum_{k=1}^{a} d_{i,k} \cdot log \frac{d_{i,k}}{d_{j,k}} \tag{3}$$

The total distance between the two curves is calculated as the sum of each time interval distance. To test significance of the activity curve distance value, W_i and W_j are concatenated to form a window W of length $2n$ days.

Next, all days within W are shuffled. The first half of the shuffled days form a new first window, W_i^*, while the second half form a new second window, W_j^*. KL distances for each DTW-aligned time interval pairs in W_i^* and W_j^* form a vector that is inserted into a matrix. This shuffling procedure is repeated N times, producing a $N \times m$ permutation matrix, M. If N is large enough, M forms an empirical distribution of the possible permutations of activity data within the two windows of time. Next, for each time interval t, the number of permuted KL distances that exceed the original change score CS is divided by N to form a p-value. After computing a p-value for each time interval t, the Benjamini-Hochberg correction [20] is applied for a given α ($\alpha = 0.01$ or 0.05). Finally, the remaining significant p-values are counted to produce the change score. For this study, we additionally normalize the change score. To do this, we divide the score by the result of DTW-alignment pairs and multiplying by 100 to yield a percentage value.

While the algorithm is intended for activity distribution data available from activity recognition algorithms, in this paper we adapt the PCAR algorithm to analyze the physical activity change detection as part of our BCD method. Instead of activity distribution vectors, we use scalar step counts. Additionally, PCAR is suitable for only large window sizes due to the requirement of permuting daily time series data. The approach originally was intended for correlating change scores with standardized clinical assessments to determine if ambient smart home sensor-based algorithms can detect cognitive decline [7]. Consequently, there is not a test for significance of the change score. In the following section we propose a version of PCAR that is more suitable for small windows (sw-PCAR) as required by BCD and in Section 3.2 we propose an accompanying significance test for sw-PCAR.

3.1.4. sw-PCAR

We propose a small window adaptation of PCAR to allow permutation-based change detection for window sizes of one week or less. Algorithm 2 outlines the sw-PCAR approach. For sw-PCAR, two windows W_i and W_j are collapsed into aggregate windows $\hat{W}_i$ and $\hat{W}_j$ (see Equation 1). A change score CS is derived by computing the KL divergence between the average number of steps taken in $\hat{W}_i$ and the average number of steps taken in $\hat{W}_j$. Next, $\hat{W}_i$ and $\hat{W}_j$ are concatenated to form a window W of length two days. All time intervals within W are shuffled. The first half of the shuffled intervals form a new first window, W_i^*, while the second half form a new second window, W_j^*. W_i^* and W_j^* are each averaged to produce two step values. The KL distance between the two values is computed and inserted into a vector. This is repeated N times to produce a N-length vector V of KL distances. Vector V is later used for change score significance testing (see Section 3.2).

Algorithm 2 sw-PCAR(W_i, W_j, N)

```
Input: W_i, W_j = two windows of time series data
Input: N = number of permutations
Output: Change score CS and Boolean sig
Initialize: k = 0
Initialize: V as a vector of length N
Compute Ŵ_i, Ŵ_j aggregate windows
Compute CS, the KL divergence between Ŵ_i and Ŵ_j
while k < N:
    Shuffle the time intervals of Ŵ_i and Ŵ_j
    Generate new aggregate windows W*_i and W*_j
    Compute KL divergence between W*_i and W*_j
    Store resulting distance in V
    k = k + 1
end while
sig = BoxplotOutlierDetection(CS, V) (see Algorithm 3)
return CS, sig
```

3.1.5. Virtual Classifier

Change analysis, as proposed by Hido *et al.* [9], utilizes a virtual binary classifier to detect and investigate change. We apply the virtual classifier (VC) approach to the physical activity change problem for large window sizes. First, a feature extraction step reduces two windows W_i and W_j into two $n \times z$ feature matrices, M_i and M_j, where n is the window size (in days) and z is the number of features that are extracted. Next, each daily feature vector of M_i is labeled with a positive class and each daily feature vector of M_j is labeled with a negative class. VC trains a decision tree to learn the decision boundary between the virtual positive and negative classes. The resulting average prediction accuracy based on k-fold cross validation represented as p_{VC}. If a significant change exists between W_i and W_j, the average classification accuracy p_{VC} of the learner should be significantly higher than the accuracy expected from random noise, $p_{rand} = 0.5$, the binomial maximum likelihood of two equal length windows.

3.2. Change Significance Testing

Significance testing of change score CS is necessary to interpret change score values. For the VC approach, Hido *et al.* [9] proposed a test of significance to determine if p_{VC} is significantly greater than p_{rand}. For this test, the inverse survival function of a binomial distribution is used to determine a critical value, $p_{critical}$, at which n Bernoulli trials are expected to exceed p_{bin} at α significance. If $p_{vc} \geq p_{critical}$, a significant change exists between the two windows, W_i and W_j.

The PCAR approach does not have an accompanying test of significance. We address this with our proposed sw-PCAR technique. sw-PCAR computes change significance by comparing CS to the permutation vector V with boxplot-based outlier detection (see Algorithm 3). An outlier can be defined as an observation which appears to be inconsistent with other observations in the dataset [21]. For this method, the interquartile range (75^{th} percentile - 25^{th} percentile) of V is computed. Values outside of the $1.5 \cdot 75^{\text{th}}$ percentile are considered outliers [22]. If CS is determined to be an outlier of V, then the change score is considered significant. There are alternative approaches to test membership of an observation (i.e. CS) to a sample distribution (i.e. V) other than boxplot outlier detection. If the sample is normal, statistical tests such as Grubb's test for outliers [23] can be applied. However, the assumption of normality does not hold for all samples of human behavior data. More advanced alternatives include data mining techniques relevant to outlier detection [21, 24]. Exploration and testing of such data mining techniques are outside the scope of this paper.

RuLSIF does not explicitly provide a method to determine a cutoff threshold for values of the Pearson divergence function are considered significant change scores. In supervised applications where ground truth change labels are available, a threshold parameter is typically learned by repeated training and testing with different parameter values. For unsupervised applications, domain knowledge and/or alternative data-driven approaches are necessary. Like RuLSIF, the texture-based method also does not provide a test of change significance. For RulSIF and texture-based approaches, we propose a medium to large window change significance test based on intra-window variability and outlier detection.

Our proposed change significance test utilizes the existence of day-to-day variability in human activity patterns [25]. In order to consider a change between two windows significant, the magnitude of change should exceed the day-to-day variability within each window. To illustrate, consider two

Algorithm 3 BoxplotOutlierDetection(CS, V)

Input: CS = change score between two windows
Input: V = sample distribution vector
Output: Boolean sig
Arrange V in ascending order
Compute Q_1, the 25$^{\text{th}}$ percentile of V
Compute Q_3, the 75$^{\text{th}}$ percentile of V
Compute the interquartile range of V, $IQR = Q_3 - Q_1$
if $CS > 1.5 \cdot IQR$:
    $sig = True$
else:
    $sig = False$
return sig

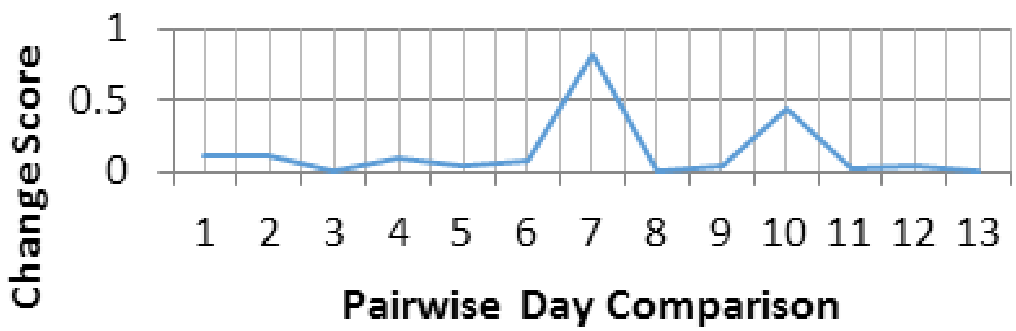


Figure 1: Pairwise sliding window RulSIF change scores. [TODO: MAKE THIS A MATPLOTLIB PDF]

adjacent, non-overlapping windows W_1 and W_8, each of length $n = 7$ days. Now run a pairwise sliding window change algorithm over W_1 concatenated with W_8. If there is a significant change between the windows, the magnitude of change should be higher for the inter-window comparison (between days 7 and 8) than any other intra-window comparison. Fig. 1 shows an example plot of RulSIF change scores for real Fitbit data illustrating this phenomenon. There are small, noisy day-to-day changes for all comparisons except the largest maximum occurring for inter-window comparison (7$^{\text{th}}$ change score) and a potential anomaly between days 3 and 4 of W_8.

Based on the assumption that a significant inter-window change should exceed intra-window change, we propose an intra-window change significance test (see Algorithm 4). Given a change score CS between two windows, the task is to determine if CS is significant. To do this, first compute a list of all possible daily change scores, DCS, within each window. DCS contains $2 \cdot \text{Combination}(n, 2)$ change scores (see Algorithm 5). For example, a week to week comparison ($n = 7$) would generate an intra-window daily change score sample of 42 day-to-day variations. Next, apply the outlier detection method (see Algorithm 3) from sw-PCAR to test if CS is an outlier score when compared to the distribution of intra-window daily change scores DCS. Advantages of the proposed test include it is non-parametric and can be coupled with any small window change algorithms. Furthermore, the approach lends itself well to online change detection algorithms, since only a vector of the previous window's (baseline or sliding) change scores need to be retained. Finally, the candidate change score, CS, can be computed based on any window size (i.e. Monday to Monday, aggregate to aggregate, week to week, etc.).

Algorithm 4 Intra-windowChangeSignificanceTest(W_1, W_2, n, CS, F)

1: Input: W_1, W_2 = two windows of time series data
2: Input: n = window size
3: Input: CS = change score between W_1 and W_2
4: Input: F = change score function
5: Output: Boolean sig
6: Initialize: Vector of daily change scores DCS
7: Append W_1 intra-window daily change scores to DCS (see Algorithm 5)
8: Append W_2 intra-window daily change scores to DCS (see Algorithm 5)
9: Compute sig = BoxplotOutlierDetection(CS, DCS) (see Algorithm 3)
10: **return** sig

3.3. Change Analysis

If a change significance test concludes a change score is significant, the next step is to determine the source of change (see Algorithm 1 for an overview of the change detection and analysis process). Often this step requires the computation of features that summarize the data and provide a meaningful context for change. For example, the number of daily steps taken is an example of a simple physical activity feature. The change between daily steps from one window of time to the next can be quantified and used for an explanation of change. Several approaches exist to capture change across

Algorithm 5 Intra-windowDailyChangeScores(W, n, F)

```
Input: W = window of time series data
Input: n = window size
Input: F = change score function
Output: Vector of daily change scores DCS
Initialize: i = 0, j = 0
while i < n − 1:
    curr = W[i]
    j = i + 1
    while j < n:
        next = W[j]
        CS = F(curr, next)
        Append CS to DCS
        j = j + 1
        end while
    i = i + 1
end while
return DCS
```

time in individual metrics. A straightforward method is to compute the percent change for a feature f from a previous window W_1 to a current window W_2:

$$\Delta\% = \frac{f_{W2} - f_{W1}}{f_{W1}} \tag{4}$$

Statistical approaches such as two sample tests or effect size analyses can also be applied to quantify change; however, in applying repeated statistical tests, the multiple testing problem should be accounted for with a method such as the Bonferroni or Benjamini-Hochberg correction [20].

One of the advantages of the virtual classifier approach over other change point detection algorithms is it includes an explanation of the source of change without reliance on statistical tests. Upon significant change detection, retraining a decision tree on the entire dataset and inspecting the tree reveals which features are most discriminatory in learning the differences between two windows. Naturally, this approach requires a pre-processing step to compute relevant features from the windowed time series data. The following section presents relevant features utilized for physical activity change analysis.

3.4. Feature Extraction

The following features are grouped together based on the number of days required for computation: 1) one day (24 hour window of time) or less, 2) at least one day, or 3) two or more days. Daily features include intraday physical activity summaries based on intensity, frequency, duration, variability of steps and walking bouts. Sequences of time series data with steps greater than t_{mins}. t_{mins} represents the minimum number of steps per time interval to be considered physical activity. This assumes physical activity is characterized by at least one step per minute. If ground truth activity labels, such as walking, biking, chores, etc., are available from the device user and/or an activity recognition algorithm, physical activity type can be inferred and thresh can be set dynamically for different activities. For this study, we assume such labeled information is not available and set thresh to t_{mins}.

- Daily PA intensity
    - Bout steps: Mean and SD of number of steps per bout
    - Period steps [4]: Mean and SD per period 1) 24 hour period (full days), 2) Day (9am-9pm), 3) Night (12am-6am). Day and night normalized by 24 hour mean
    - Ratio of mean night and day steps [4]: See period steps definition
- Daily PA frequency
    - Number of bouts: Count of detected PA bouts
- Daily PA duration
    - Bout minutes: Mean and SD of duration of bouts
    - PA intensity percentage: Mean percentage of 1) sedentary (<5 steps/min), 2) low ($5 \leq$ steps/min < 40), 3) moderate ($40 \leq$ steps/min < 100), 4) high (≥ 100 steps/min) activity levels
    - Rest minutes: Mean and SD of duration of rest periods

Features computed on window sizes of at least one day include an adaptation of relative amplitude from Merilahti *et al.* [3] and texture features from the texture-based change detection approach [18] (see Section 3.1.2).

- Relative amplitude: Normalized ratio between the most active 8 hours and the least active 4 hours activity periods (not required to be consecutive). If sleep data is available, awake hours are used for least active periods. $RA = \frac{M8-L4}{M8+L4}$
- Texture features: See section 3.1.2

Features requiring at least two days of data summarize activity across or between days or quantify the users circadian rhythm (the periodicity from day-to-day [25]). Poincare-plot analysis [4] provides an additional set of useful physical activity features. Poincare plots depict how activity patterns repeat themselves based on a time delay, d. Time series data at time t, $A(t)$, is plotted as a function of previous data, $A(t-d)$. From the resulting Poincare plot, two measures of dispersion can be computed 1) SD1, the standard deviation of the data against the axis $x = y$ and 2) SD2, standard deviation of the data against the axis orthogonal to x=y and crosses this axis at the mean value of the data (center of mass). Delay values of $d = 24hours$ and $d = 12hours$ plot the data as a function of the previous data and the counter phase respectively. The day-to-day circadian rhythm preservation (CRP) feature is based on dispersion values from these two delays. [TODO: include sample poincare plots?]

- Inter-daily stability (IS) [3]: Quantifies stability between the days $IS = \frac{n\sigma h=1,p(x_h-x)^2}{p\sum_{i=1}^{n}(x_i-x)^2}$
- Intra-daily variability (IV) [3]: Quantifies the fragmentation of rhythm and activity $IV = \frac{n\sum_{i=2}^{n}(x_i-x_{i-1})^2}{(n-1)\sum_{i=1}^{n}(x\_i-x)^2}$
- Circadian rhythm strength (CRS) [3]: Divides average night-time activity (11pm- 5am) by the average activity of the previous day (8am-8pm) $CRS = \frac{steps_{11pm-5am}}{steps_{prev8am-8pm}}$
- Cosinor mesor: Time series mean from fitting a cosinor functional model with a 24 hour period to time series data via least squares method [3, 4]
- Cosinor amplitude: Difference between the mesor and peak (or trough) of the fitted waveform
- Cosinor acrophase: Time of day at which the peak of a rhythm occurs

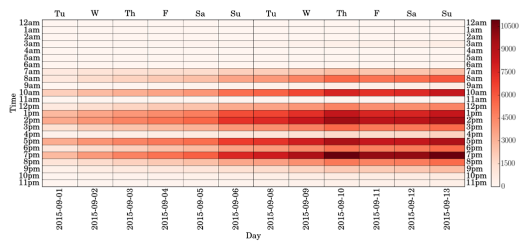
- Poincare plot SD1 [4]: Standard deviation of Poincare data against the axis $x = y$
- Poincare plot SD2 [4]: Standard deviation of Poincare data against the axis orthogonal to $x = y$ and crosses this axis at the mean value of the data (center of mass)
- Poincare plot circadian rhythm preservation (CRP) [4]: Day-to-day circadian rhythm preservation based on dispersion values from SD1 and SD2 with delays of 24 hours and 12 hours $CRP = SD2_{24h} + SD1_{12h} - SD1_{24h} - SD2_{12h}$

3.5. Datasets

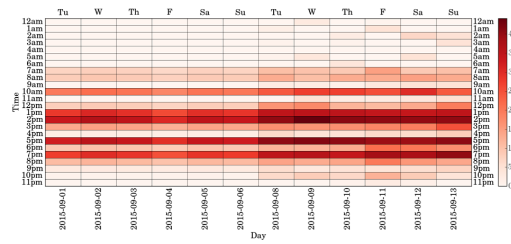
To evaluate the change detection algorithms, two datasets are presented, Hybrid-synthetic (HS) and B-Fit (BF). The HS dataset comprises synthetic data and the BF dataset comprises real-world data collected from a Fitbit study. To generate the HS dataset, step data collected from a volunteer wearing a Fitbit Charge Heart Rate fitness tracker was re-sampled and modified to produce five different synthetic physical activity profiles, each exhibiting a different type of change. The length of HS profiles was set to 12 days, resulting in two equal size windows of 6 days for comparison. Twelve days was chosen for similarity to the real-world BF dataset. The HS profiles with their profile identification (HS0-4) and a description are as follows:

1. HS0: No significant daily or window change. Data is subject to small daily variation.
2. HS1: Medium daily change and consequently significant window change. Increase bout duration and intensity from day-to-day.
3. HS2: No significant daily change but significant window change. Increased activity for days 7-12.
4. HS3: Medium daily change and consequently significant window change. Increase activity variability from day-to-day.
5. HS4: No significant daily change for days 1-6. Significant daily change for days 7-12. Consequently significant window change.

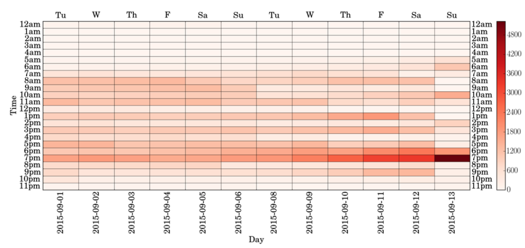
Figure 2 shows the associated activity density maps (ADMs) for HS1-4 profiles. An ADM is a heat map proposed by Wang *et al.* [5] to visualize daily activity (shade of color) as a function of 24 hour time (Y-axis) and window time (X-axis).



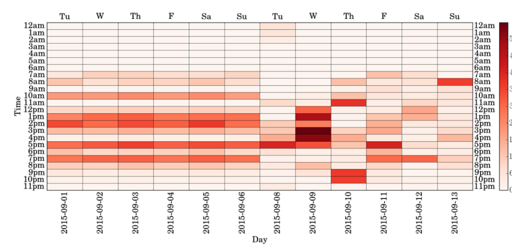
(a) HS1



(b) HS2



(c) HS3



(d) HS4

Figure 2: Hybrid-synthetic (HS) activity density maps for HS1-4.

The BF dataset consists of data collected from 11 older adults (57.09 ± 8.79) participating in a 10-week health intervention called B-Fit (see Table 2 for participant characteristics). For this study, participants pre-intervention physical activity profiles were assessed with wrist-worn Fitbit Flex fitness trackers for one week (six full 24 hour days) before the intervention. During weeks two through nine, the participants were educated in eight different subjects related to health: exercise, cardiovascular risk factors, sleep, stress reduction, cognitive engagement, nutrition, social engagement, and compensatory strategy use. Each week was devoted to education, goal setting, and goal tracking for each of the eight categories. Example goals the participants set included “brisk walking, four times a week for a half hour or more” and “drinking more water at work.” To track goal achievement, individuals rated themselves from 0 to 3 (0: did not meet goal, 1: partly met goal, 2: completely met goal, 3: exceeded goal). Table 2 shows self-ratings for the BF participants for the categories of exercise and cardiovascular risk factors. After the intervention, participants' post-intervention physical activity profiles were assessed for one week (six full 24 hour days) with the same Fitbit devices.

Table 2: B-Fit study participant characteristics

ID	Sex	Age	Height	Weight	S1 dates	S2 dates	Exercise	Cardio
BF1	F	65	5'4"	210 lbs	9/13-9/20	11/15-11/22	1.8 Brisk walking	2.5
BF3	F	68	5'3"	207 lbs	9/13-9/20	11/15-11/22	3 Walk more	2.86
BF18	F	65	5'2"	165 lbs	9/23-9/30	12/2-12/9	1 Take stairs	2
BF20	M	64	6'0"	190 lbs	9/23-9/30	12/2-12/9	1.75 Walking	2
BF23	F	57	5'5"	135 lbs	9/24-9/30	12/2-12/9	2.17 Yoga	1
BF24	F	53	5'6"	130 lbs	9/23-9/30	12/2-12/9	0.6 Walking	2
BF26	F	65	5'8"	145 lbs	9/23-9/30	12/2-12/9	1.67 Crossfit class	0
BF27	M	42	5'8"	145 lbs	9/23-9/30	12/2-12/9	2 Walking	3
BF28	M	48	6'4"	380 lbs	9/23-9/30	12/2-12/9	0 No exercise	3
BF29	F	52	5'6"	180 lbs	9/23-9/30	12/2-12/9	1 Walking	1
BF30	F	49	5'6"	262 lbs	9/23-9/30	12/2-12/9	1.92 Yoga	2

BF = B-Fit, ID = identification, lbs = pounds.

HS and BF data were subject to cleaning prior to serving as input to the change detection algorithms. First, missing data were identified and handled for BF data. Since donned/doffed information is not available for Fitbit Flex fitness trackers, days with zero steps taken during the day (9am-9pm) were considered missing data. By this method, three days exhibited missing data, each belonging to a different participant (11/18/15: BF3; 9/28/15: BF27; 12/7/15: BF29, see Table 2 for participant information). Zero steps during the day is likely due to removing the Fitbit to charge it, then forgetting to put it back on until much later.

Algorithm 6, WindowedFillMissingData, summarizes the missing data filling process. First, to fill a missing day ($D_{missing}$), the same day of the week (i.e Monday) of $D_{missing}$ is identified in the opposite window (D_{other}). As an example, P27 exhibited missing data on 9/28/15, which was a Monday in the pre-intervention window. Next, the Monday of the post-intervention window is identified (12/7/15 in this case). Euclidean distance-based clustering is then applied to find the k nearest neighbors NN_{other} of D_{other}, $3 \leq k \leq 5$. The days of the week for each day in NN_{other} are then identified. These are used to select days, $NN_{missing}$ in the original window containing $D_{missing}$. The k days of $NN_{missing}$ are averaged and used to fill $D_{missing}$ from 9am to 9pm. Three nearest neighbors is chosen as a minimum value for k in case one of the days of the week is not available in the original six day window.

Additional pre-processing of Fitbit data includes down sampling the data for a given time interval length, t_{mins}, by summing the steps every t_{mins} minutes. Furthermore, for the case of PCAR and sw-PCAR, add one smoothing was applied to avoid a division by zero during KL divergence computations. Finally, while the Fitbit Flex also provides distance traveled and calories burned, for this study we only consider steps due to the high inter-metric redundancy between steps and these two Fitbit metrics. For the BH data, Pearson correlations of $r > 0.99$ for distance and $r > 0.90$ for calories were measured.

4. Results

All data are processed with the Python programming language. Unless otherwise stated, the following parameter values are used: k (fill missing data): 3; n (window size): 6 days; *offset* (window offset): 6 days; α (RuLSIF): 0.1; RuLSIF cross validation folds: 5; GLCM distance: 2; N (number of PCAR and sw-PCAR permutations): 1000; α (PCAR): 0.05; virtual classifier

Algorithm 6 WindowedFillMissingData($D_{missing}$, $W_{missing}$, W_{other}, k)

1: Input: $D_{missing}$ = day with missing data to fill
2: Input: $W_{missing}$ = window of time series data containing $D_{missing}$
3: Input: W_{other} = window of time series data
4: Input: k = number of nearest neighbors
5: Output: D_{fill} day data to fill $D_{missing}$
6: D_{other} = day in W_{other} with same day of week as $D_{missing}$
7: NN_{other} = k nearest neighbors of D_{other} in W_{other}
8: $NN_{missing}$ = k days in $W_{missing}$ with same days of week as days in NN_{other}
9: $D_{missing}$[9am:9pm] = average of k days in $NN_{missing}$[9am:9pm]
10: **return** D_{fill}

cross validation folds: 4; virtual classifier prediction threshold $p_{critical}$: 0.75. The time interval aggregation size t_{mins} is tested with values of t_{mins} = 1, 5, 10, 15,..., 60 minutes.

Table 3 shows RuLSIF, Texture-based, sw-PCAR, and VC significant change results for each HS profile for each time interval length t_{mins}. Table 4 shows PCAR change scores for each HS profile and BF participant for each time interval length. The contextual features of number of bouts, bout minutes, bout steps, daily steps, and sedentary minutes percent are listed in Table 5 with window one and window two values (mean and SD). Results in Table 5 have time interval length t_{mins} = 1 minute in order to report the most-detailed feature values. For further change analysis, decision trees are shown in Figure 3 for HS profiles HS1-4. Similarly for the BF dataset, each participant's change scores and change significance testing results are presented in Table 6. The five contextual features (number of bouts, bout minutes, bout steps, daily steps, and sedentary percent) pre and post-intervention values are listed in Table 7. Finally, decision trees are shown in Figure 4 for select BF participants with a significant virtual classifier change score (BF3, BF26, and BF29).

5. Discussion

We investigate unsupervised change detection and analysis for step-based time series. We compare five change detection approaches, four from the literature and one proposed algorithm. We also implement change significance testing and compute several features for explaining detected changes. The

Table 3: Hybrid-synthetic (HS) significant change detection as a function of time interval size t_{mins} for each HS profile. Results are in the form Count: Boolean (is change significant? 0: false, 1: true) {HS0, HS1, HS2, HS3, HS4}.

t_{mins}	RulSIF	Texture-based	sw-PCAR	Virtual classifier	Total
1	2:0,0,1,0,1	1:0,0,1,0,0	3:0,1,1,0,1	4:0,1,1,1,1	10
5	3:0,1,1,0,1	2:0,0,1,0,1	3:0,1,1,0,1	4:0,1,1,1,1	12
10	2:0,0,1,0,1	2:0,0,1,1,0	2:0,1,1,0,0	3:0,0,1,1,1	9
15	3:0,1,1,0,1	3:0,1,1,1,0	1:0,1,0,0,0	3:0,0,1,1,1	10
20	3:0,1,1,0,1	4:0,1,1,1,1	1:0,1,0,0,0	3:0,0,1,1,1	11
25	1:0,0,1,0,0	2:0,0,1,1,0	1:0,1,0,0,0	3:0,0,1,1,1	7
30	3:0,1,1,0,1	1:0,0,0,1,0	1:0,1,0,0,0	4:0,1,1,1,1	9
35	2:0,1,1,0,0	1:0,0,0,1,0	1:0,1,0,0,0	3:0,1,1,0,1	7
40	3:0,1,1,1,0	0:0,0,0,0,0	1:0,1,0,0,0	3:0,0,1,1,1	7
45	2:0,0,1,0,1	0:0,0,0,0,0	1:0,1,0,0,0	2:0,0,1,0,1	5
50	2:0,0,1,0,1	0:0,0,0,0,0	1:0,1,0,0,0	4:0,1,1,1,1	7
55	3:0,1,1,1,0	1:0,0,0,1,0	0:0,0,0,0,0	3:0,1,1,0,1	7
60	4:0,1,1,1,1	0:0,0,0,0,0	1:0,1,0,0,0	4:0,1,1,1,1	9
Total	33:0,8,13,3,9	17:0,2,6,7,2	17:0,12,3,0,2	43:0,7,13,10,13	110

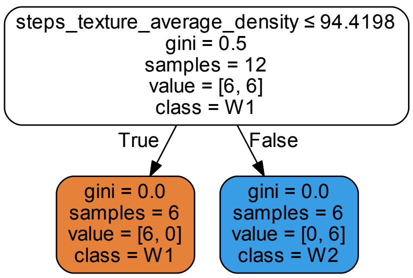
0 = insignificant, 1 = significant

Table 4: Hybrid-synthetic (HS) and B-Fit (BF) significant change detection as a function of time interval size t_{mins} for each HS profile. HS esults are in the form PCAR change values {HS0, HS1, HS2, HS3, HS4}. BF results list the top 3 participants with the highest PCAR change scores.

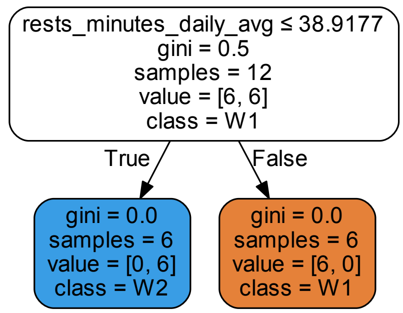
t_{mins}	Hybrid-synthetic PCAR (%)	B-Fit PCAR (%)
1	0.67,48.10,29.89,43.43.2.89	BF27:1.52,BF30:0.73,BF20:0.04
5	32.95,53.33,38.07,44.67,13.77	BF27:3.39,BF30:1.87,BF20:1.62
10	31.22,50.00,52.06,48.00,23.45	BF3:10.04,BF27:7.30,BF26:5.83
15	0.00,50.71,56.34,45.80,10.00	BF3:36.84,BF27:9.02,BF1:4.73
20	0.00,53.92,65.66,61.29,17.31	BF3:34.68,BF27:24.51,BF30:9.47
25	0.00,55.00,66.22,59.46,12.05	BF3:38.95,BF30:11.39,BF23:5.88
30	0.00,70.77,65.15,60.66,27.27	BF3:41.18,BF27:17.91,BF1:7.58
35	0.00,56.36,80.85,46.15,22.95	BF3:52.23,BF27:24.14,BF20:22.22
40	0.00,60.87,58.70,55.32,20.76	BF3:52.83,BF27:9.26,BF29:8.70
45	0.00,65.91,63.42,51.28,25.00	BF3:51.06,BF1:15.23,BF27:8.33
50	22.86,63.16,57.14,50.00,27.5	BF3:48.98,BF27:23.81,BF1:22.73
55	0.00,62.86,63.64,54.55,42.11	BF3:50.00,BF27:31.58,BF1:15.39
60	0.00,70.00,72.41,53.33,33.33	BF3:51.43,BF1:18.75,BF27:14.29

Table 5: Hybrid-synthetic (HS) feature results (mean ± standard deviation) with t_{mins} = 1 minute. Window one and window two values are separated by a comma.

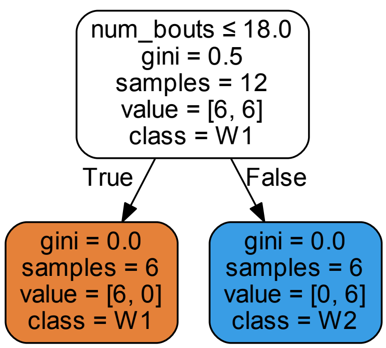
ID	Number of bouts	Bout minutes	Daily steps	Sedentary %
HS0	70.33, 70.00	5.10 ± 9.91, 5.13 ± 9.92	20601.65, 21274.32	75.65, 75.56
HS1	34.50, 14.17	19.39 ± 23.93, 46.82 ± 56.78	36409.49, 72769.11	64.57, 54.59
HS2	71.50, 62.50	5.07 ± 9.83, 7.63 ± 11.13	20755.53, 30037.48	75.62, 67.44
HS3	54.83, 102.83	18.71 ± 51.74, 6.49 ± 14.49	14395.85, 14746.43	45.22, 63.72
HS4	53.50, 81.33	8.14 ± 12.61, 4.56 ± 8.48	22048.02, 17327.00	70.66, 77.86



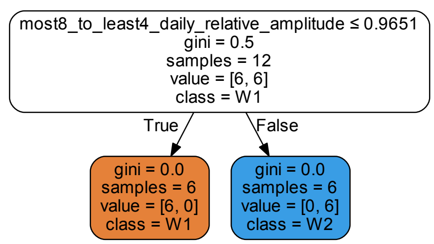
(a) HS1



(b) HS2



(c) HS3



(d) HS4

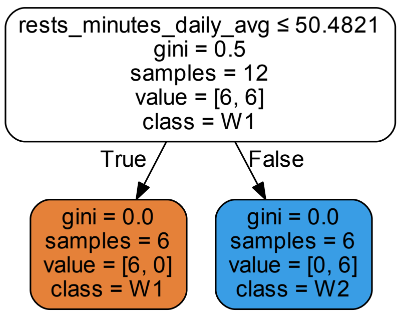
Figure 3: Decision trees for HS profiles with significant virtual classifier change scores for t_{mins}=5 minutes.

Table 6: B-Fit (BF) significant change detection as a function of time interval size t_{mins}. Results are in the sparse form Count:IDs: (Boolean is change significant? 0 false, 1 true).

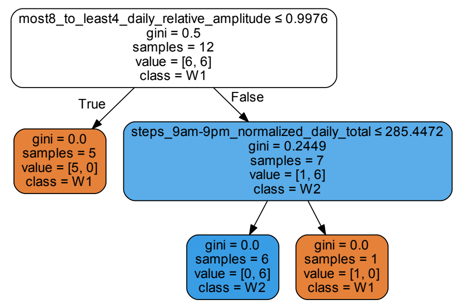
t_{mins}	RulSIF	Texture	sw-PCAR	Virtual classifier	Total
1	1:BF3:1	0	10:BF29:0	5:BF3,24,26,29,30:1	16
5	2:BF3,26:1	0	5:BF1,3,18,20,27:1	6:BF3,24,26,27,29,30:1	13
10	1:BF3:1	0	4:BF1,3,20,27:1	6:BF3,18,24,26,27,29:1	11
15	1:BF3:1	0	4:BF1,3,20,27:1	4:BF3,26,29,30:1	9
20	2:BF3,26:1	0	3:BF3,20,27:1	4:BF3,26,29,30:1	9
25	1:BF3:1	0	3:BF3,20,27:1	2:BF26,29:1	6
30	2:BF3,27:1	0	3:BF3,20,27:1	2:BF24,29:1	7
35	3:BF3,20,27:1	0	3:BF3,20,27:1	5:BF3,24,26,29,30:1	11
40	1:BF3:1	0	3:BF3,20,27:1	3:BF3,26,29:1	7
45	4:BF3,18,20,24	0	1:BF3:1	2:BF3,29:1	7
50	1:BF3:1	0	1:BF3:1	5:BF3,24,26,28,29:1	7
55	1:BF29:1	0	1:BF3:1	3:BF3,20,29:1	5
60	2:BF20,28:1	1:BF27:1	1:BF3:1	4:BF3,24,29,30	8
Total	22	1	42	51	116

Table 7: B-Fit (BF) feature results (mean $\pm$ standard deviation) with $t_{mins} = 1$ minute. Pre and post intervention values are separated by a comma.

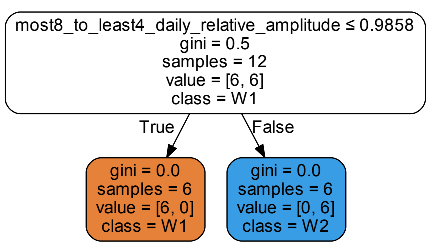
ID	Number of bouts	Bout minutes	Daily steps	Sedentary %
BF1	73.50, 89.67	2.35 $\pm$ 1.93, 2.63 $\pm$ 2.61	3479.00, 4658.33	88.37%, 84.43%
BF3	81.00, 15.83	2.57 $\pm$ 2.54, 2.75 $\pm$ 1.85	4279.50, 1161.44	86.30%, 97.44%
BF18	88.50, 27.50	2.72 $\pm$ 2.86, 2.36 $\pm$ 2.08	5886.67, 4558.50	84.06%, 86.32%
BF20	81.17, 60.33	3.90 $\pm$ 5.27, 3.71 $\pm$ 4.85	11177.00, 7399.67	79.11%, 85.71%
BF23	73.67, 76.50	2.96 $\pm$ 4.06, 2.60 $\pm$ 3.22	6994.17, 5470.50	85.71%, 86.22%
BF24	105.33, 88.00	2.63 $\pm$ 2.46, 2.66 $\pm$ 2.39	7127.00, 6207.67	81.82%, 84.85%
BF26	64.33, 63.50	3.30 $\pm$ 4.20, 3.45 $\pm$ 3.61	7354.67, 6181.17	85.78%, 85.90%
BF27	104.17, 102.50	5.52 $\pm$ 7.51, 3.35 $\pm$ 3.79	17680.78, 11440.00	66.66%, 77.18%
BF28	99.50, 116.67	2.40 $\pm$ 2.49, 2.40 $\pm$ 2.57	5844.50, 6731.83	84.11%, 81.46%
BF29	85.00, 80.50	2.99 $\pm$ 3.24, 3.58 $\pm$ 4.29	1136.51, 1210.85	82.94%, 81.62%
BF30	83.00, 89.50	2.51 $\pm$ 2.73, 2.31 $\pm$ 2.23	5753.50, 4868.83	86.16%, 86.44%



(a) BF3



(b) BF26



(c) BF29

Figure 4: Decision trees for B-Fit (BF) participants with significant virtual classifier change scores for t_{mins}= 5 minutes.

abilities of the presented methods to detect change are evaluated on two original datasets: 1) 5 synthetic profiles and 2) 11 participant's Fitbit data from an intervention study.

5.1. Hybrid Synthetic Dataset

The HS dataset reveals several insights into the change detection algorithms. First, the time interval length yielding the highest number of significant changes is t_{mins} = 5 minutes with 12 changes, closely followed by t_{mins} = 20 minutes with 11 changes (see Table 3). Since HS profiles are sampled from a volunteers real user Fitbit data, these intervals suggest movement patterns occur in 5 and 20 minute chunks for this individual. For all time interval lengths, the algorithms do not detect a significant change between window one and window two data for the HS0 profile. HS0 is generated to exhibit small day-to-day variation in step intensity and is not characterized by large changes between windows; however, PCAR detects an average of 6.75% change for HS0. We can use PCAR HS0 values as baseline change scores for relative interpretations of PCAR HS1-4 values.

HS2 and HS4 were generated to exhibit abrupt changes between the first and second window, whereas HS1 and HS3 exhibit gradual day-to-day change from day one of window one to the last day of window 2. For HS1-4 profiles, significant changes between windows are detected. For all time interval lengths, the virtual classifier approach picks up the most changes (43 changes), followed by RuLSIF (33), sw-PCAR (17), and texture-based (17). Changes for HS2 (35) and HS1 (29) are the most frequently detected, followed by HS4 (26) and HS3 (20). As a group, the algorithms' are able to sense change in value (HS1, HS2) and changes in variability (HS3, HS4), with 64 and 46 changes respectively. RuLSIF struggles to detect gradual variability change (HS3: 8), but perfectly detects window-based value change (HS2: 13) for all time intervals. In fact, perfect detections are made by virtual classifier (HS2, HS4: 13) and near perfect detections are made for sw-PCAR (HS1: 12). Investigating the t_{mins} = 5 minutes results reveal all four algorithms determine significant changes for HS2 and HS4 (see Table 3); however, for HS2 and HS4, PCAR identifies lower change (38.07% and 13.77%) than the gradual change profiles HS1 and HS3 (53.33% and 44.67%). This HS4 PCAR score is less than the $t_{5minutes}$ HS0 PCAR score, 32.95%, implying PCAR did not detect noteworthy change for HS4.

Upon inspection of the associated decision trees for HS2-4 (see Figure 3, the features of texture density, average daily rest minutes, number of bouts, and window-based relative amplitude are discriminatory features. The explanatory power of the features is potentially useful for reporting to the wearable sensor user the dimensions of change in their physical activity. Features useful for such purposes are simple, common features that do not require interpretation. For example, texture density or relative amplitude are useful features for detecting changes in PA patterns, but are relatively unimportant to a user. More meaningful features to a user include number of bouts, number of steps per bout, rest period minutes, and sedentary percent. Table 5 shows these features for the HS profiles. HS0 exhibits quite similar window one and window two values for all features. HS2 and HS4 both have small standard deviations due window-based change in lieu of day-to-day change (HS1 and HS3).

5.2. B-Fit Dataset

Analyzing the BF participants' data poses additional challenges that are not present with the HS profiles. Real-world human subject data is inherently noisy, characterized by seemingly random bouts of PA and rest periods. Furthermore, self-report and direct measurement of physical activity are often not congruent, with previous studies reporting correlations in as wide of a range of -0.71 to 0.96 [26]. For the BF group, Table 2 shows a wide spread of self-reported goal achievement ratings for the exercise and cardiovascular risk factor categories. For example, BF24, BF28, and BF29 rated their exercise goal achievements low (exercise: 0.6, 0, 1 respectively). Due to heart problems, BF28's doctor instructed him not participate in exercise-related activities. On the other hand, BF3 rated their goal achievements the highest (exercise: 3; cardio 2.86). Upon inspection of BF3's data, it is evident there is a discrepancy between the participant's perception of her PA and the steps recorded by the Fitbit (specifically 9/14-9/19 compared to 11/19-11/21). It is not uncommon for self-reported measures of physical activity to be inconsistent with direct measures [26]; therefore, the self-ratings presented in Table 2 are used for insights into individual goal achievements, not as ground truth information for changes exhibited. The issues with self-reported PA measures exacerbate the need for unsupervised change detection and analysis methods.

Depending on the algorithm, significant changes are commonly detected for 5 out of the 11 BF participants: BF3: 35; BF27: 14; BF29:14; BF20: 13, BF26: 12; (see Table 3). Virtual classifier and sw-PCAR detect the highest number of changes (51 and 42 changes each), but the distribution of detected changes is highly influenced by time interval length (3.92 ± 1.44, 3.23 ± 2.42 number of changes detected respectively). sw-PCAR is not sensitive for small time intervals (t_{mins}=1 minutes) or large time intervals (t_{mins}={45, 50, 55, 60 minutes}), and the number of changes detected decreases as time interval length increases. Virtual classifier does not appear to be as heavily influenced by the time interval length. The texture-based approach is the least sensitive algorithm, only detecting change for BF27 with t_{mins} = 60 minutes. Finally, PCAR detects the most changes with time window t_{mins} = 35 minutes.

Performing change analysis and investigating the changes detected yields insights for several of the participants. BF3 rated herself as completely meeting her exercise goal of walking more; however, the Fitbit data tells a different story. Several features in Table 5 show decreased PA for BF3: average number of bouts (W1: 81.00, W2: 15.83), daily steps (W1: 4279.50, W2: 1161.44 steps), and percentage of time sedentary (W1: 86.30%, W2: 97.44%). Additionally, BF3's decision tree (see Figure 4a) provides evidence that she rested more during post-intervention testing. In summary, the features suggest the changes detected by the algorithms are actually changes in the opposite direction of her goal. Contrary to BF3, BF29 exhibited a significant change (as detected consistently by virtual classifier) in the direction towards her goal of walking more. Inspection of BF29's features shows an increase bout minutes, bout steps, and average steps per day. Average daily steps increased from 1136.51 steps pre-intervention to 1210.85 steps post-intervention testing, a 6.54% increase. The remaining participants with significant changes (BF3, BF20, BF24, BF26, BF27, and BF30) demonstrated a decrease in average daily steps taken from pre to post intervention. While this suggests the exercise component of the intervention was not successful for these participants, the participants' physical activity levels may have been influenced by seasonal effects [27]. Pre-intervention Fitbit data collection occurred in September, which is considerably warmer than November/December in Washington State, which was the period of post-intervention data collection. It is also worth noting the participants exhibited improvements in other physical activity features. For example, relative amplitude has been reported to decrease with worsening health [28], thus BF26 and BF29's increased relative amplitude post-intervention is healthy (see Figures 4b and 4c). Also, BF28 was not planning on increasing exercise; however, BF28 increased their daily steps post-intervention by 15.18%.

One of the limitations of this study includes only having one week of pre-intervention Fitbit data for BF participants. With at least two weeks of pre-intervention data, change scores can be computed between week one and two of pre-intervention data to provide an estimate of inter-week variability. With a quantification of inter-week variability, we can determine if the change measured between pre and post-intervention weeks is due to the intervention or natural variability. An additional limitation includes not having full 7 days of BF data during pre and post intervention weeks. Finally, more sophisticated methods to fill missing data could be utilized with fitness trackers that include heart rate monitors, due to more reliable detection of sensor donned/doffed. Consequently, future work includes performing change analysis on real-world datasets from different fitness trackers, multivariate data (i.e. heart rate, elevation, etc.), labeled activity data, and longer windows of time. With time series data longer than two years, several additional analyses could be performed: daily/weekly/monthly/yearly period analysis, slicing along different dimensions (i.e. Mondays, weekends, holidays, or activities if labeled information is available), etc.

6. Conclusion

We address the problem of unsupervised physical activity change detection and analysis. We compare five change detection approaches, four from the literature and one proposed algorithm. We objectively compare the algorithms' abilities to capture different types of changes in five distinct synthetic datasets representing realistic changes in physical activity patterns. We also evaluate the algorithms on real-world physical activity data collected from an intervention study where 11 Fitbit users set goals to improve various facets of their health. The results indicate the algorithms detected the most significant changes in both datasets for time interval lengths of 1, 5, and 15 minutes. For the synthetic dataset, the virtual classifier and sw-PCAR approaches picked up on the highest number of changes. Changes for profiles exhibiting large changes between windows are more likely to be detected than those exhibiting incremental day-to-day changes. For the real world dataset, the algorithms frequently detect changes for 5 of the 11 participants. Change analysis for these 5 participants' physical activity features reveal only 1 exhibits an increase in daily steps taken post-intervention. Contextual features such as average number of daily steps, minutes spent resting, number of steps per bout, and sedentary percent provide an explanation of the changes detected. The algorithms and analysis methods are useful data mining techniques for unsupervised, window-based change detection. Future work involves implementing the algorithms in an online, smartphone application to track users' physical activity and motivate their progress towards their health goals.

Acknowledgements

We wish to thank Catherine Sumida and Thao Vo for their help in data collection. This work is supported in part by [Grant?].

References

[1] P. Lally, C. van Jaarsveld, H. Potts, J. Wardle, How are habits formed: Modelling habit formation in the real world, Eur. J. Soc. Psychol. 40 (6) (2010) 998–1009.

[2] B. H. Dobkin, Wearable motion sensors to continuously measure real-world physical activities, Curr Opin Neurol 26 (6) (2013) 602–608.

[3] J. Merilahti, P. Viramo, I. Korhonen, Wearable monitoring of physical functioning and disability changes, circadian rhythms and sleep patterns in nursing home residents, IEEE Journal of Biomedical and Health Informatics PP (99) (2015) 1–1.

[4] P. Paavilainen, I. Korhonen, J. Ltjnen, L. Cluitmans, M. Jylh, A. Srel, M. Partinen, Circadian activity rhythm in demented and non-demented nursing-home residents measured by telemetric actigraphy, J Sleep Res 14 (1) (2005) 61–68.

[5] S. Wang, M. Skubic, Y. Zhu, Activity Density Map Visualization and Dissimilarity Comparison for Eldercare Monitoring, IEEE Transactions on Information Technology in Biomedicine 16 (4) (2012) 607–614.

[6] T.-H. Tan, M. Gochoo, K.-H. Chen, F.-R. Jean, Y.-F. Chen, F.-J. Shih, C. F. Ho, Indoor activity monitoring system for elderly using RFID and Fitbit Flex wristband, in: 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), 2014, pp. 41–44.

[7] P. N. Dawadi, D. J. Cook, M. Schmitter-Edgecombe, Modeling patterns of activities using activity curves, Pervasive and Mobile Computing

[8] S. Liu, M. Yamada, N. Collier, M. Sugiyama, Change-point detection in time-series data by relative density-ratio estimation, Neural Networks 43 (2013) 72–83.

[9] S. Hido, T. Id, H. Kashima, H. Kubo, H. Matsuzawa, Unsupervised Change Analysis Using Supervised Learning, in: T. Washio, E. Suzuki, K. M. Ting, A. Inokuchi (Eds.), Advances in Knowledge Discovery and Data Mining, no. 5012 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 148–159, dOI: 10.1007/978-3-540-68125-0_15.

[10] M. Yamada, A. Kimura, F. Naya, H. Sawada, Change-point Detection with Feature Selection in High-dimensional Time-series Data, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13, AAAI Press, Beijing, China, 2013, pp. 1827–1833.

[11] D.-H. Tran, M. M. Gaber, K.-U. Sattler, Change Detection in Streaming Data in the Era of Big Data: Models and Issues, SIGKDD Explor. Newsl. 16 (1) (2014) 30–38.

[12] K. Feuz, D. Cook, C. Rosasco, K. Robertson, M. Schmitter-Edgecombe, Automated Detection of Activity Transitions for Prompting, IEEE Transactions on Human-Machine Systems 45 (5) (2015) 575–585.

[13] F. Javed, S. Farrugia, M. Colefax, K. Schindhelm, Early Warning of Acute Decompensation in Heart Failure Patients Using a Noncontact Measure of Stability Index, IEEE Transactions on Biomedical Engineering 63 (2) (2016) 438–448.

[14] M. Hu, S. Zhou, J. Wei, Y. Deng, W. Qu, Change-point Detection in Multivariate Time-series Data by Recurrence Plot, WSEAS Transactions on Computers 13 (2014) 592–599.

[15] Y. Xu, Z. Zhang, P. Yu, B. Long, Pattern Change Discovery Between High Dimensional Data Sets, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, ACM, New York, NY, USA, 2011, pp. 1097–1106.

[16] W. Ng, M. Dash, A change detector for mining frequent patterns over evolving data streams, in: IEEE International Conference on Systems, Man and Cybernetics, 2008. SMC 2008, 2008, pp. 2407–2412.

[17] C. J. Caspersen, K. E. Powell, G. M. Christenson, Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research., Public Health Rep 100 (2) (1985) 126–131.

[18] F. Albregtsen, others, Statistical texture measures computed from gray level coocurrence matrices, Image processing laboratory, Department of informatics, university of oslo (2008) 1–14.

[19] L. Chen, J. Hoey, C. Nugent, D. Cook, Z. Yu, Sensor-Based Activity Recognition, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (6) (2012) 790–808.

[20] Y. Benjamini, Y. Hochberg, Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, Journal of the Royal Statistical Society. Series B (Methodological) 57 (1) (1995) 289–300.

[21] O. Maimon, L. Rokach, Data mining and knowledge discovery handbook, Springer, New York, 2005.

[22] J. W. Tukey, Exploratory data analysis.

[23] F. E. Grubbs, Procedures for Detecting Outlying Observations in Samples, Technometrics 11 (1) (1969) 1.

[24] V. J. Hodge, J. Austin, A Survey of Outlier Detection Methodologies, Artif Intell Rev 22 (2) (2004) 85–126.

[25] R. Refinetti, G. C. Lissen, F. Halberg, Procedures for numerical analysis of circadian rhythms, Biol Rhythm Res 38 (4) (2007) 275–325.

[26] S. A. Prince, K. B. Adamo, M. E. Hamel, J. Hardt, S. C. Gorber, M. Tremblay, A comparison of direct versus self-report measures for assessing physical activity in adults: a systematic review, International Journal of Behavioral Nutrition and Physical Activity 5 (2008) 56.

[27] P. Tucker, J. Gilliland, The effect of season and weather on physical activity: A systematic review, Public Health 121 (12) (2007) 909–922.

[28] J. Merilahti, T. Petkoski-Hult, M. Ermes, M. v. Gils, H. Lahti, A. Ylinen, L. Autio, E. Hyvrinen, J. Hyttinen, Evaluation of new concept for balance and gait analysis: patients with neurological disease, elderly people and young people, Gerontechnology 7 (2) (2008) 164.