Computer Methods and Programs in Biomedicine 212 (2021) 106461

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

FLIRT: A feature generation toolkit for wearable data

Simon Föll[a,1,2,*], Martin Maritsch[a,1,3], Federica Spinola[b], Varun Mishra[c,4], Filipe Barata[a,5], Tobias Kowatsch[a,d,6], Elgar Fleisch[a,d,7], Felix Wortmann[d,8]

[a] Department of Management, Technology, and Economics, ETH Zurich, Zurich, Switzerland
[b] Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland
[c] Department of Computer Science, Dartmouth College, Hanover, NH, USA
[d] Institute of Technology Management, University of St. Gallen, St. Gallen, Switzerland

ARTICLE INFO

Article history:
Received 4 March 2021
Accepted 6 October 2021

Keywords:
Physiological signal processing
Wearable sensors
Artifact detection
Signal filtering
Machine learning
Feature engineering

ABSTRACT

Background and Objective: Researchers use wearable sensing data and machine learning (ML) models to predict various health and behavioral outcomes. However, sensor data from commercial wearables are prone to noise, missing, or artifacts. Even with the recent interest in deploying commercial wearables for long-term studies, there does not exist a standardized way to process the raw sensor data and researchers often use highly specific functions to preprocess, clean, normalize, and compute features. This leads to a lack of uniformity and reproducibility across different studies, making it difficult to compare results. To overcome these issues, we present *FLIRT: A Feature Generation Toolkit for Wearable Data*; it is an open-source Python package that focuses on processing physiological data specifically from commercial wearables with all its challenges from data cleaning to feature extraction.

Methods: FLIRT leverages a variety of state-of-the-art algorithms (e.g., particle filters, ML-based artifact detection) to ensure a robust preprocessing of physiological data from wearables. In a subsequent step, FLIRT utilizes a sliding-window approach and calculates a feature vector of more than 100 dimensions – a basis for a wide variety of ML algorithms.

Results: We evaluated FLIRT on the publicly available WESAD dataset, which focuses on stress detection with an Empatica E4 wearable. Preprocessing the data with FLIRT ensures that unintended noise and artifacts are appropriately filtered. In the classification task, FLIRT outperforms the preprocessing baseline of the original WESAD paper.

Conclusion: FLIRT provides functionalities beyond existing packages that can address unmet needs in physiological data processing and feature generation: (a) integrated handling of common wearable file formats (e.g., Empatica E4 archives), (b) robust preprocessing, and (c) standardized feature generation that ensures reproducibility of results. Nevertheless, while FLIRT comes with a default configuration to accommodate most situations, it offers a highly configurable interface for all of its implemented algorithms to account for specific needs.

1. Introduction

The advances in wearable technologies and sensor quality have enabled researchers to increasingly use wearables to passively sense and record physiological and behavioral data signals in daily living conditions. Devices from a wide range of manufacturers such

* Corresponding author at Department of Management, Technology, and Economics, ETH Zurich, Weinbergstrasse 56/58, 092 Zurich, Switzerland.
E-mail addresses: sfoell@ethz.ch (S. Föll), mmaritsch@ethz.ch (M. Maritsch), fspinola@student.ethz.ch (F. Spinola), varun@cs.dartmouth.edu (V. Mishra), fbarata@ethz.ch (F. Barata), tkowatsch@ethz.ch (T. Kowatsch), efleisch@ethz.ch (E. Fleisch), felix.wortmann@unisg.ch (F. Wortmann).
[1] These authors contributed equally.
[2] [orcid: 0000-0002-4364-4282].
[3] [orcid: 0000-0001-9920-0587].
[4] [orcid: 0000-0003-3891-5460].
[5] [orcid: 0000-0002-3905-2380].
[6] [orcid: 0000-0001-5939-4145].
[7] [orcid: 0000-0002-4842-1117].
[8] [orcid: 0000-0001-5034-2023].

https://doi.org/10.1016/j.cmpb.2021.106461

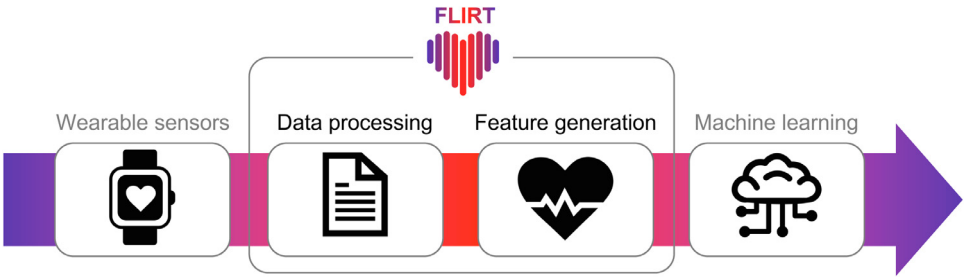

Fig. 1. Key functionalities provided by FLIRT.

Table 1
FLIRT package metadata.
Metadata description.

License	MIT
Implementation	Python 3.7+
Code repository	https://github.com/im-ethz/flirt
Documentation	https://flirt.readthedocs.io
PyPI installation	`pip install flirt`

as Empatica, Oura, Garmin, Fitbit, Lifecard, and Apple provide a variety of sensor streams like electrocardiogram (ECG), heart rate variability (HRV), and more recently, electrodermal activity (EDA). Results have shown that it is feasible to build machine learning (ML) models to detect and predict various health and behavioral outcomes, like stress, hypoglycemia, cognitive engagement, and COVID-19 onset (see for example [32,46,48,52,59]). However, data from wearables are prone to noise, missing, artifacts; and despite increasing interest in deploying such wearables for long-term studies, there is not yet a standardized way to process its raw sensor data. Researchers often implement custom functionalities to preprocess, clean, normalize, and compute features from these sensor signals, which results in a lack of uniformity across studies, thus making it difficult to compare the results from different studies. Further, the lack of a standard processing pipeline makes it challenging to reproduce or replicate the results achieved in a particular study by researchers not involved in the original study, even when the data is publicly available. Hence, one cannot quantitatively evaluate the benefit of a new ML model or approach over previous models because it is unclear if the performance improvement is due to the "better" model or a difference in the data processing steps. Thus, the lack of standardization is a significant challenge before realizing the true potential of using wearables for continuous physiological signals.

In this work, we take a concrete step towards realizing a tool that can deal with artifacts, measure gaps within the input data to construct reliable digital biomarkers. Specifically, our work outlines methods to reliably process physiological data from wearable devices to provide standardized and meaningful features for researchers to build their ML models. We bundle these functionalities into a Python package, *FLIRT: A Feature Generation Toolkit for Wearable Data* (Table 1). FLIRT is a middleware between raw sensor data and ML models that aims to provide a standardized way to generate physiological features from wearables (see Fig. 1). Researchers can then use these features to develop ML models for healthcare applications such as digital biomarkers.

Our motivation to build such a tool stemmed from our inability to effectively reproduce the results from a paper on affect and stress detection using the authors' publicly available data, even when trying to replicate the same ML model as accurately as possible as detailed by the authors. In the original paper, there are several crucial gaps in how the authors processed their data, making it infeasible to replicate results and compare if and how new ML models can lead to better performance. Our work aims to prevent such scenarios in the future. Thus, we implemented several different state-of-the-art processing methodologies and algorithms in FLIRT, which are easily configurable through various parameters. We envision that in the future, researchers can share their configurations for FLIRT, and succeeding works can leverage those parameters to emulate the same processed data and features, thus enabling evaluation of reproducibility and replicability and also the test the effectiveness of new ML models and methods compared to prior works in that domain. Furthermore, we open-sourced FLIRT to encourage further development with the power of many.

2. Related work

Prior research has shown considerable progress in applying ML algorithms to physiological data to develop, for example, novel digital biomarkers. According to their feature generation, we classify these approaches into two categories. The first category includes approaches where the ML models themselves extract the features. As an example, consider the end-to-end learning of one-dimensional convolutional neural networks [7,52,60]. The second class subsumes approaches that leverage explicit feature engineering, such as calculating a pre-defined set of features on time segments of a physiological recording [12,46,59]. Although the second class has the advantage that interpretable features can be selected, it requires an intensive effort to preprocess the physiological data and select and compute the associated features.

To point out the contribution of FLIRT, we evaluated the current state on Python packages for processing and generating physiological features. For this purpose, we first select and evaluate packages specialized in processing physiological data.

Python packages include *Neurokit2* [43], *BioSPPy* [66], *PyPhysio* [15], *PySiology* [21], *HeartPy* [23], *HRV* [11], *hrv-analysis* [17], and *pyHRV* [25]. Packages such as *Neurokit2* provide comprehensive pipelines to process any kinds of physiological signals. However, these algorithms are usually tailored to signals obtained by professional medical equipment. In order to achieve robust results with data from wearables applied in the wild, where artifacts frequently occur, such a package is not the preferable choice. Other packages such as *hrv* and *hrv-analysis* solely focus on retrieving HRV features, while other signals such as EDA cannot be processed.

Additionally, we evaluate the packages mentioned above according to their functionalities:

- **File reader:** Manufacturers of wearables (e.g., Empatica) provide non-standardized though documented file formats, which contain the desired physiological data. Moreover, recordings of medical devices (e.g., ECG) are often exported in proprietary file formats (e.g., Holter format). By directly processing proprietary file formats, such as zipped archives provided by the Empatica E4 wearable, a package facilities easy application.
- **Preprocessing:** Data recordings from commercial wearable are often prone to artifacts, measuring gaps, or deviations from the

Table 2
Overview of Python packages for physiological data processing.

Package	File reader	Sliding window	Preprocessing	ECG	IBI	EDA	ACC
BioSPPy				✓	✓	✓	
HeartPy				✓	✓		
HRV	✓		✓		✓		
hrv-analysis					✓		
Neurokit2			✓	✓	✓	✓	
pyHRV				✓	✓		
PyPhysio		✓			✓	✓	
PySiology				✓	✓	✓	
FLIRT	✓	✓	✓	✓	✓	✓	✓

Electrocardiogram (ECG), inter-beat interval (IBI), electrodermal activity (EDA), and accelerometer (ACC).

Table 3
IBI artifact detection rules.

Rule	Description
Malik [44]	Each IBI should not differ more than 20 percent compared to the preceding IBI.
Kamarth [34]	Each IBI should not increase or decrease by more than 32.5 percent nor by more than 24.5 percent from the previous interval.
Acar [1]	Removing IBIs that differ by more than the 20 percent of the mean of the last nine IBIs.
Karlsson [35]	Removing IBIs that differ by more than 20 percent of the mean of the preceding and succeeding IBI.

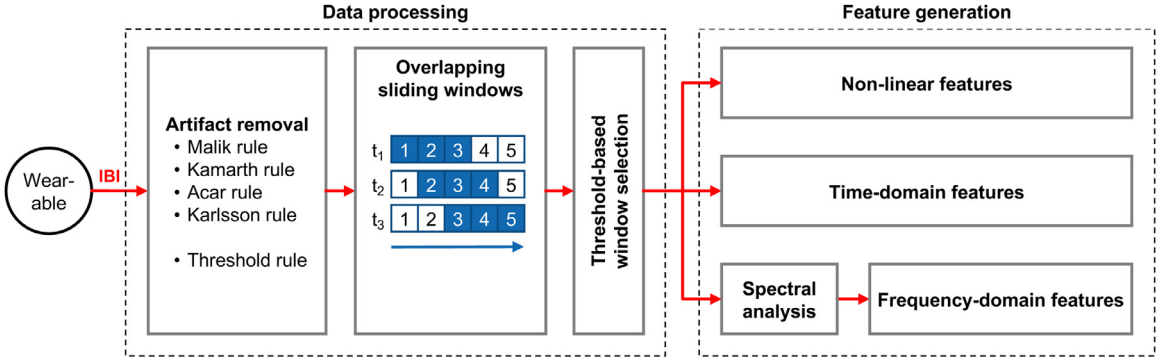

Fig. 2. IBI processing pipeline.

measurement regime [33,40,62]. Therefore, robust preprocessing to filter noise and artifacts to restore the original signal is crucial to retrieving reliable results from data obtained in the wild.

- **Signal modalities:** Recent wearable technology incorporates a range of sensor modalities with signals such as ECG, inter-beat interval (IBI), EDA, and accelerometer (ACC). Thus, packages for physiological data processing should be capable of dealing with data from those sensors.

Table 2 compares the functionalities of all evaluated packages. While the presented selection of packages cover the functionalities for specific parts of physiological data processing, there is lack of a holistic solution. We bridge this gap by providing a fully integrated package from reading proprietary files to robust preprocessing for various wearable signals specifically tailored to wearable device characteristics.

3. Computational methods

Data recordings from wearables are often prone to artifacts, measuring gaps, or deviations from the measurement regime [33,40,62]. Thus, robust preprocessing to generate a clean signal are crucial to retrieve reliable ML results from data obtained by wearables. Furthermore, although selected ML models can leverage raw signal data, the majority of available models does not do so. Thus, the user needs to manually write code that segments the time-series into windows and generate features per time window – difficult for researchers with limited background in data science.

In summary, preparing physiological data, or in general terms time series data, for ML models, comprises three steps: (1) preprocessing, (2) generate customizable (overlapping) sliding windows, and (3) feature engineering.

3.1. Cardiovascular activity

Wearable devices often use photoplethysmography (PPG) sensors to measure IBIs [4]. IBIs quantify the time distance between two consecutive heart beats. Since raw PPG recordings from commercial wearables are only rarely accessible (e.g., Empatica E4), we focus on processing the more readily available IBI (or the normal-to-normal (NN) interval) data. Using this IBI data, we calculate HRV measures, reflecting the change between consecutive heart beats. Leveraging (commercial) wearable devices to retrieve HRV measures comes with inherent challenges such as artifacts or measurement gaps [14,33,53]. Thus, we place special emphasis on increasing the reliability of the measures in our HRV processing pipeline. All steps are summarized in Fig. 2.

3.1.1. Preprocessing and window selection

As a first preprocessing step we remove obvious artifacts within the IBI series. The automatic artifact removal comprises four rules which are presented in Table 3. Additionally, IBIs outside the physiological feasible range of 250–2000ms (equals a heart rate (HR)

of 30–240) are discarded as well [55]. The cleaned IBIs are then partitioned into overlapping time windows for feature generation.

The IBI series can contain measuring gaps caused by either discarded (ectopic) beats or heart beats not recognized by the measurement device; a frequent case with wearables [14,33]. Handling missing data is an important step to avoid otherwise negatively affected HRV measures [50,53]. This error in HRV features drastically increases with the occurrence and length of measuring gaps [41,50]. However, applying interpolation methods to recover missing beats can cause a significant proportion of errors in HRV features whenever measuring gaps occur [50]. In contrast to interpolation, incomplete time windows can be discarded prior to HRV feature calculation to prevent overly inaccurate measures [44]. Thus, instead of accepting an arbitrary error in HRV measures, we increase the reliability of the HRV measure by defining a threshold criterion for the removal of incomplete windows.

We do not rely on a static threshold such as the minimum number of sinus beats because it would imply two major drawbacks. First, such approaches cannot be individualized to single subjects and, second, they are not dynamically adjusting to temporal changes in the input signal (e.g., diurnal changes in heart rates). Based on the idea of [14], we introduce an adaptive threshold analysis method. We set an adaptive threshold for a minimum number of detected beats based on the expected number of beats per individual time window. The expected number of beats is estimated via the arithmetic mean over the detected heart beats within a time window.

A time window t is not discarded if it satisfies the inequality

$$\left\lceil thresh \cdot \frac{L}{\mu_{IBI_t}} \right\rceil < N_t,$$

where L denotes the window length in seconds, N_t the number of detected valid beats in window t, μ_{IBI_t} the mean IBI of window t in seconds, and $thresh \in [0, 1]$ the threshold, which can be chosen arbitrarily based on the desired application. In summary, the higher the threshold is set the higher the required proportion of detected heart beats with respect to the amount of expected beats.

3.1.2. Feature engineering

Current research highlights three feature categories for HRV: statistical, time-domain, and frequency-domain features [2,44,47,61,63,68]. In Table 4, we summarize all features.

First, time-domain features represent the variability of the IBIs over a time specific period. Additionally, we retrieve the instantaneous heart rate from the IBIs and compute basic time domain features. In case the wearable provides raw PPG data, one can generate statistical features for the blood volume pulse. In general, time-domain features are quickly calculable but provide less discriminant power to distinguish between the two branches of the autonomic nervous system [2].

Second, frequency-domain features reflect the energy distribution over a range of frequencies. In particular, this class of HRV measures better quantifies changes in the balance between the sympathetic and parasympathetic autonomic nervous system [2]. For frequency-domain HRV features, we have to estimate the power spectral density (PSD) of the partitioned IBI sequence [9]. Estimating PSD based on the fast Fourier transform requires an interpolation method to produce an artificial equally-sampled discrete time-series [9,38,44,58]. Since interpolation can affect the power spectrum, and thus frequency-domain features [50], we rely on the Lomb-Scargle method [42,57]. This method is robust against unequally-sampled time-series and has shown promising results in the context of HRV analysis [9,38,49].

Third, non-linear HRV features quantify the uncertainty in the IBI sequence. We added them, since time- and feature-domain cannot account for the entire complexity of the mechanisms regulating HRV [58].

3.2. Electrodermal activity

In this section, we present an electrodermal activity (EDA) processing pipeline in order to retrieve reliable and standardized EDA measures for ML applications. All steps are depicted in Fig. 3.

3.2.1. Preprocessing EDA

We present two distinct categories of approaches. First, two *integrated* approaches which comprises artifact removal and noise filtering: (a) the extended Kalman filter (EKF) and (b) the particle filter (PF). Second, a *modular* approach which allows to combine low-pass filter and artifact detection algorithms.

Kalman filter The Kalman filter is an integrated, model-based approach for filtering data, which combines the data measurements with a theoretical model of the signal to estimate its true response. Our procedure implements the EKF algorithm as it has shown reasonable results in removing noise and artifacts from data gathered using wearables [67].

We estimate the state matrix as

$$\mathbf{x} = [SC_H, k_{diff}, SC_0, SCR, S]^T, \tag{1}$$

where SC_H is the hydration-dependent contribution to the skin conductance (SC), SC_0 is the baseline SC of the inert skin, SCR denotes the skin conductance response (SCR), k_{diff} is the inverse time constant of the sweat diffusion, and the variable S denotes the sudomotor nerve activation.

The model equations found in [67] are discretized and the nonlinear system is then linearized around the last predicted state in each time step. At each time step, the well-known Kalman filter steps are performed: the prediction step (prior update) and the measurement update (posterior update).

The final estimate of the SC signal is retrieved as $SC = SC_H + SC_0 + SCR$. Initialization was conducted according to [67], with the mean vector as $\mathbf{x_0} = [0, 0, 0, 0, 0]^T$, and the variance as

$$\mathbf{P_0} = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0.001 & 0.01 \\ 0 & 0 & 0 & 0.01 & 0.01 \end{bmatrix},$$

where we slightly modified the variance of SC_H, k_{diff}, and SC_0. The intuition is that an over-idealized model may hinder the convergence of the extended Kalman filter specifically when this filter is applied to real-world data.

Particle filter Similarly to the EKF, the PF is a model-based filtering algorithm. However, PF assumes in contrast to EKF no normal distribution of state and noise random variables. This allows the PF algorithm to be more widely applicable to wearable signals and scenarios with highly non-Gaussian noise.

We propose a PF algorithm based on the *pyParticleEst* Python package [54]. The algorithm is based on a linear process and measurement model:

$$\begin{aligned} x_{k+1} &= x_k + v_k \\ z_k &= x_k + w_k, \end{aligned} \tag{2}$$

where x_k is the SC signal to estimate, z_k is the EDA measurement, v_k is the process noise and w_k is the measurement noise, all at time k.

A specified number of particles are sampled from the initial state distribution to initialize the algorithm. Subsequently, the particles are updated and propagated to the next time step by first sampling from the prior distribution (the model) and then from

Table 4
HR and HRV features.

Category	Name	Description	Unit
Statistical	*min/max HR*	Minimum and maximum of the HR	bpm
	mean HR	Mean of the HR_i	bpm
	median HR	Median of the HR_i	bpm
Time domain	*SDNN*	SD of all NN intervals	ms
	RMSSD	The square root of the mean of the sum of the squares of differences between adjacent NN intervals	ms
	NN_{50}	Number of pairs of adjacent NN intervals differing by more than 50 ms in the entire recording	ms
	pNN_{50}	NN_{50} count divided by the total number of all NN intervals	%
	NN_{20}	Number of pairs of adjacent NN intervals differing by more than 20 ms in the entire recording	-
	pNN_{20}	NN_{20} count divided by the total number of all NN intervals.	%
	CVNN	Coefficient of variation equal to the ratio of SDNN divided by mean NN interval	-
	CVSD	Coefficient of variation of successive differences equal to the RMSSD divided by mean NN interval	-
	mean	Mean of the IBIs	ms
	std	Standard deviation of the IBIs	ms
	min/max	Minimum and maximum of the IBIs	ms
	ptp	Range (peak to peak) of the IBIs	ms
	sum	Sum of the IBIs	ms
	energy	Energy of the IBIs	ms^2
	skewness	Skewness of the IBIs	-
	kurtosis	Kurtosis of the IBIs	-
	peaks	Number of the IBIs	-
	rms	Root mean square of the IBIs	ms
	line_integral	Integral under the IBIs	ms
	n_above_mean	Number of IBIs above the mean	-
	n_below_mean	Number of IBIs below the mean	-
	n_sign_changes	Number of changes in the IBIs slope	-
	iqr	Interquartile range between the 25th and 75th percentile of the IBIs	ms
	iqr 5 – 95	Interquartile range between the 5th and 95th percentile of the IBIs	ms
	pct 5	5th percentile of the IBIs	-
	pct 95	95th percentile of the IBIs	-
	entropy	Entropy of the IBIs	-
	perm entropy	Permutation entropy of the IBIs	-
	svd entropy	Singular value decomposition of the IBIs entropy	-
Frequency domain	*total power*	The variance of NN intervals over the temporal segment below 0.04 Hz	ms^2
	vlf	Power in very low frequency range below or equal 0.04 Hz	ms^2
	lf	Power in low frequency range 0.04 Hz and 0.15 Hz	ms^2
	hf	Power in high frequency range 0.15 Hz and 0.4 Hz	ms^2
	lf/hf – ratio	Ratio of LF to HF	-
	lfnu	LF power in normalized units	-
	hfnu	HF power in normalized units	-

High frequency (HF), low frequency (LF), heart rate (HR), inter-beat interval (IBI), and normal-to-normal (NN) interval.

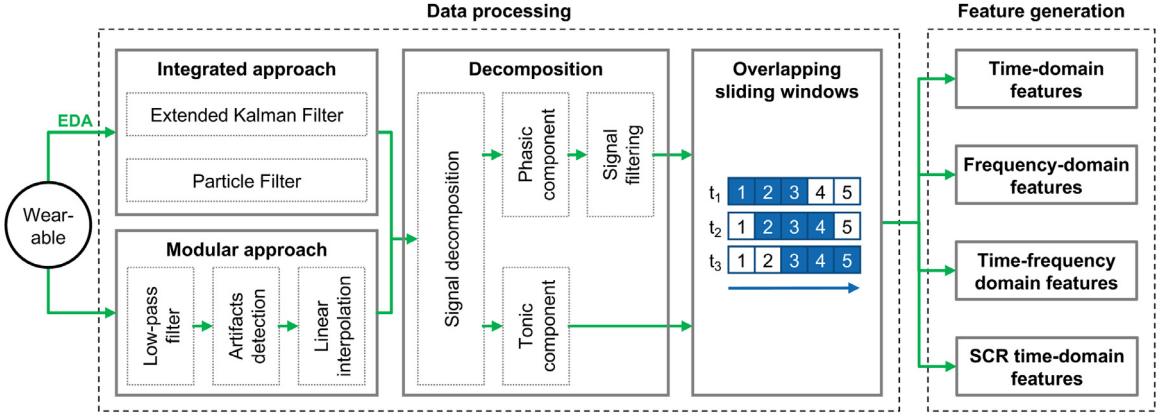

Fig. 3. EDA processing pipeline.

the posterior distribution (the measured SC signal). Additionally, a smoothing algorithm is applied to re-weight the particles, granting a larger contribution to the particles that better represent the signal.

Low-pass filter. The effect of measurement noise can be attenuated by using a low-pass filter with the desired cut-off frequency. In technical terms, a low-pass filter keeps only the frequencies below a specified threshold. Low-pass filtering is the traditional approach to automatically reduce noise in EDA signals [22,37,56]. We leverage several infinite impulse response filtering methods [69]. The cutoff frequency is set below 0.5 Hz, since the SC signal is band limited to 0.5 Hz [5].

In contrast to the integrated approach, artifacts can be detected and removed using ML algorithms as they have been proven successful in automatically and accurately detecting artifacts in EDA signals recorded with wrist-worn devices [65,72]. In FLIRT, researchers can choose from the following two integrated, pre-trained models:

EDAexplorer. Taylor et al. proposed this method, which detects motion artifacts in the EDA raw data by classifying each consecu-

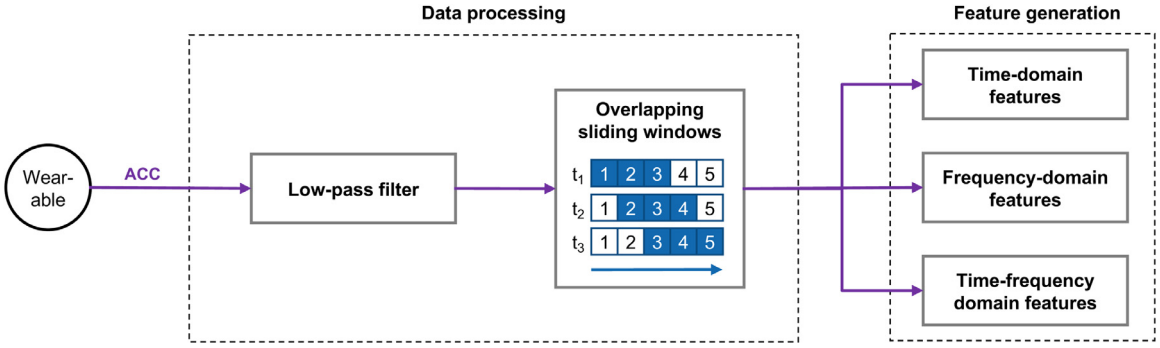

Fig. 4. ACC processing pipeline.

tive five second epoch into *artifact, questionable*, or *clean*. The classification is performed using a support vector machine on features computed on the raw and low-pass filtered EDA data (for more detailed information see [65] or [64]).

Ideas-Lab UT. Further, we include a second method [72] that detects motion artifacts in the raw EDA signal by classifying each consecutive five second epoch into *artifact* or *clean*. The classification is performed using an logistic regression (LR) model on features computed on the raw EDA data. The artifact detector was trained on available labeled data, partially recorded in a controlled environment and partially in-the-wild (for more detailed information see [72]).

These methods can be optionally enabled to optimize the outcome of EDA preprocessing. The methods' performance in detecting artifacts, however, may depend on the particular use case itself. In both cases, the artifacts are marked and the resulting gap in the raw EDA signal is linearly interpolated.

3.2.2. Decomposition

The EDA signal consists of two SC components: skin conductance response (SCR) and skin conductance level (SCL) or in other words, the phasic and tonic component, respectively. Several decomposition algorithms for both components have been implemented in the literature [6,8,13,26,29]. Below we elaborate on two algorithms for EDA signal decomposition that are included in FLIRT because of their wide adoption, cvxEDA [26] and Ledalab [13].

cvxEDA. The basic principle of cvxEDA is to model the EDA signal as the sum of a SCR term, a SCL term, and additive white Gaussian noise [26]. The algorithm then determines the SCR and SCL components that maximize the likelihood of observing a specific SC time-series. The convex optimization problem is rewritten as a standard quadratic program, which can be solved efficiently.

Ledalab. The alternative Ledalab was chosen because it requires no parameters other than the data itself and therefore can be generalized to all situations, without the need for additional parameter fine-tuning [13]. The key concept of the Ledalab algorithm is to model the SC signal as a sum of a SCR and SCL driver, convolved with an impulse response. This impulse response is modeled using the Bateman equation with parameters $\tau 1$ and $\tau 2$ (to be optimized).

In presence of physiologically incoherent (nerve firing cannot be negative) data, the SCR and SCL components can be further filtered using a low-pass Butterworth filter to remove negative SCR and SCL values.

3.2.3. Feature engineering

FLIRT computes general statistical and entropy features on both the SCR and SCL components for each window. Time domain features were chosen since they best describe the SCR and SCL signals recorded from wearable devices, as suggested by [24,28]. In contrast, frequency domain features of the EDA were proven to provide additional valuable information about the transient behaviour of the sudomotor activity [3,24]. The benefits of time-frequency domain features is that they capture the oscillatory behavior of the sudomotor response. Therefore, this class of features can provide information to detect and distinguish health conditions [24].

To calculate time-frequency domain features, the Cepstrum signal C is estimated by [24]

$$C = IFT(log|FT(X)|), \tag{3}$$

where X is the signal of interest, FT is Fourier Transform and IFT is the inverse Fourier Transform. We are interested in the real part of C, which gives us the required Mel-Frequency Cepstrum Components as a basis of our features.

Peak features are computed only on the SCR component of each window. Table 6 shows a description of the calculated peak features. We chose the best performing peak detection algorithms out of [27,36,43,51,65] and implemented them in FLIRT: the EDAexplorer algorithm [64,65] and the algorithm implemented in NeuroKit2 [43].

The EDAexplorer peak detection algorithm determines the presence of a peak based on several criteria related to a typical SCR peak morphology such as the signal's rate of change, maximum allowed rise time, and decay time. The Neurokit peak detection algorithm is based on the *SciPy* [69] Python package peak detection function. It refines its peak search by specifying constraints based on the peaks derivative and on the density of peaks within a time-window.

Table 5 outlines the main parameters and their default values that can be individualized by the user. A detailed description of all calculated features is presented in Table 6.

3.3. Accelerometer data

In addition to sensors that passively record physiological data, commercial wearables also have an embedded acceleration sensor. In most cases a three-axis ACC is utilized, measuring accelerations in the x-, y-, and z-axis. ACC data has been utilized for ML-based tasks such as human activity recognition or inferring physical activity intensity [16,71]. Even physiological features calculated from wearable sensors such as HRV can be improved by taking into account ACC information [18,45].

3.3.1. Preprocessing ACC

As outlined in prior work, filtering ACC data is an important step to reduce noise in the recordings. Although the number of ways to filter the signal is many, there is a consensus in applying

Table 5
Overview of parameters for EDA preprocessing. Corresponding default parameter values are indicated in parentheses.

Algorithm	Available main parameter
Low-pass filter [69]	`cutoff` (1e-1), `filter` ('butter')
EKF [67]	`d_min` (5e-3) `d_max` (5e-1), `min_diff` (1e-3), `max_diff` (1e-1), `min_Savi_Go` (-3e-2), `max_Savi_Go` (3e-2)
PF [54]	`num_particles` (80), `P0_variance` (2-e2), `Q_variance` (2e-2), `R_variance` (6e-2)
EDAexplorer[65]	–
Ideas-Lab UT[72]	–
cvxEDA [26]	`delta_knot` (15), `cvx_alpha` (8e-3), `gamma` (1e-3)
Ledalab [13]	`optimization` (0)
EDAexplorer peaks [65]	`threshold` (1e-2)
Neurokit peaks [43]	`amplitude_min` (3e-2)

Table 6
EDA features.

Category	Name	Description	Unit
Time domain	*mean*	Mean of the SCR and SCL	μS
	std	Standard deviation of the SCR and SCL	μS
	min/max	Minimum and maximum of the SCR and SCL	μS
	ptp	Range (peak to peak) of SCR and SCL within a time interval	μS
	sum	Sum of the SCR and SCL values with a time interval	μS
	energy	Energy of the SCR and SCL	μS^2
	skewness	Skewness of the SCR and SCL	-
	kurtosis	Kurtosis of the SCR and SCL	-
	peaks	Number of SCR and SCL peaks with a time interval	-
	rms	Root mean square of the SCR and SCL	μS
	line_integral	Integral under the SCR and SCL curve	$\mu S.s$
	n_above_mean	Number of SCR and SCL data-points above the mean	-
	n_below_mean	Number of SCR and SCL data-points below the mean	-
	n_sign_changes	Number of changes in the SCR and SCL slope	-
	iqr	Interquartile range between the 25th and 75th percentile of the SCR and SCL	μS
	iqr_5 – 95	Interquartile range between the 5th and 95th percentile of the SCR and SCL	μS
	pct_5	5th percentile of the SCR and SCL	-
	pct_95	95th percentile of the SCR and SCL	-
	entropy	Entropy of the SCR and SCL	-
	perm_entropy	Permutation entropy of the SCR and SCL	-
	svd_entropy	Singular value decomposition of the SCR and SCL entropy	-
Frequency domain	*sma*	Signal magnitude area of the frequency domain SCR and SCL	μS
	energy	Energy of the frequency domain SCR and SCL	μS^2
	kurtosis	Kurtosis of the frequency domain SCR and SCL	-
	iqr	Interquartile range of the frequency domain SCR and SCL	$\mu S/Hz$
	spectral_power	5 spectral power magnitudes in the [0.05-0.55] Hz bands for the power density of the SCR and SCL	$\mu S^2.Hz$
	var_power	Variance of the SCR and SCL spectral power	μS^2
Time-frequency domain	*mean*	Mean of the SCR's and SCL's MFC signal	μS
	std	Mean of the SCR's and SCL's MFC signal	μS
	median	Median of the SCR's and SCL's MFC signal	μS
	iqr	Interquartile range of the SCR's and SCL's MFC signal	μS
	skewness	Skewness of the SCR's and SCL's MFC signal	-
	kurtosis	Kurtosis of the SCR's and SCL's MFC signal	-
SCR time-domain features	*peaks*	Number of SCR peaks	-
	rise_time	Mean of the SCR peaks rise time	s
	max_deriv	Mean value of the maximum derivative of the SCR peaks	$\mu S/s$
	amp	Mean amplitude of the SCR peaks	μS
	decay_time	Mean of the SCR peaks decay time	s
	scr_width	Mean width of the SCR peaks	s
	auc_mean	Mean area-under-curves of the SCR peaks	$\mu S.s$
	auc_sum	Sum of the area-under-curves of the SCR peak	$\mu S.s$

Mel-Frequency Cepstrum (MFC), skin conductance level (SCL), and skin conductance response (SCR).

low-pass filters [20,30,70]. As suggested by Fridolfsson et al. [20], we set the default cut-off frequency to 10 Hz.

3.3.2. Feature engineering

For ACC, FLIRT provides an additional generic set of features consisting of time domain, frequency domain, and time-frequency domain features [10,31,39,59]. We calculate this set of features on the filtered ACC data. Following the idea of [59], we computed the set of features for each ACC axis and additionally on the l2-norm of these three axes.Table 7 summarizes those features and Figure 4 depicts our ACC pipeline. Note that for the spectral power features we chose the ranges according to Huynh et al. [31] and estimated the Cepstrum signal with Eq. 3. In addition, the generic set of statistical features applied to ACC could as well be leveraged for other sensor data such as temperature readings.

4. Software description

FLIRT is implemented in Python and supports Python 3 with versions 3.7 and above. The package structure is shown in Table 8.

We provide an extensive documentation of the application programmer interface (API) via the popular *Sphinx* Python documentation tool. It translates source-code documentation markup into pretty, human-readable documentation. Furthermore, the API doc-

Table 7
ACC features.

Category	Name	Description	Unit
Time domain	*mean*	Mean of the ACC signal	g
	std	Standard deviation of the ACC signal	g
	min/max	Minimum and maximum of the ACC signal	g
	ptp	Range (peak to peak) of the ACC signal	g
	sum	Sum of the ACC signal	g
	energy	Energy of the ACC signal	g^2
	skewness	Skewness of the ACC signal	-
	kurtosis	Kurtosis of the ACC signal	-
	peaks	Number of the ACC signal	-
	rms	Root mean square of the ACC signal	g
	line_integral	Integral under the ACC signal	g
	n_above_mean	Number of ACC signal above the mean	-
	n_below_mean	Number of ACC signal below the mean	-
	n_sign_changes	Number of changes in the ACC signal slope	-
	iqr	Interquartile range between the 25th and 75th percentile of the ACC signal	g
	iqr_5 – 95	Interquantile range between the 5th and 95th percentile of the ACC signal	g
	pct_5	5th percentile of the ACC signal	-
	pct_95	95th percentile of the ACC signal	-
	entropy	Entropy of the ACC signal	-
	perm_entropy	Permutation entropy of the ACC signal	-
	svd_entropy	Singular value decomposition of the ACC signal entropy	-
Frequency domain	*sma*	Signal magnitude area of the frequency domain ACC signal	g
	energy	Energy of the frequency domain ACC signal	g^2
	kurtosis	Kurtosis of the frequency domain ACC signal	-
	iqr	Interquartile range of the frequency domain ACC signal	g/Hz
	spectral_power	3 spectral power magnitudes in the [1–5.5] Hz bands for the power density of the ACC signal	$g^2.Hz$
	var_power	Variance of the ACC signal spectral power	g^2
Time-frequency domain	*mean*	Mean of the ACC signal's MFC signal	g
	std	Mean of the ACC signal's MFC signal	g
	median	Median of the ACC signal's MFC signal	g
	iqr	Interquartile range of the ACC signal's MFC signal	g
	skewness	Skewness of the ACC signal's MFC signal	-
	kurtosis	Kurtosis of the ACC signal's MFC signal	-

Accelerometer (ACC) and gravitational force equivalent (g).

Table 8
FLIRT software sub-packages.

Sub-package	Description
`flirt`	Provides access to most common functions (`get_eda_features, get_hrv_features, get_acc_features`).
`flirt.reader`	Provides implementations to read common file formats such as Empatica E4 (`flirt.reader.empatica`) or Holter devices (`flirt.reader.holter`).
`flirt.with_`	Convenience sub-package to provide functions for commonly used methods. For example, for a ready Empatica E4 zip archive a function call to `flirt.with_.empatica('E4.zip')` will automatically parse all available data and generate features.
`flirt.eda`	Provides access to low-level feature functions for processing EDA data.
`flirt.hrv`	Provides access to low-level feature functions for processing HRV data.
`flirt.acc`	Provides access to low-level feature functions for processing ACC data.
`flirt.stats`	Provides access to a standard set of statistical aggregation functions which will generate features in a window-based approach for arbitrary time-series data.

umentation can easily be extended by custom pages like articles. The documentation artifact is provided on the public *readthedocs*[9] service.

To facilitate fast and easy installation, we provide FLIRT via PyPI and it can easily be installed to Python environments via `pip install flirt`.

In its core, FLIRT uses parallelization wherever possible and allows for multi-threaded execution. Specifically, FLIRT by default uses parallel multi-core processing for the feature generation loops. Since the feature calculation process for each time window does not depend on information from other time windows, the sequence of execution is irrelevant and thus can be parallelized and executed in an arbitrary order. Furthermore, when working with complex signal decomposition methods, we rely on already existing, vectorized implementations out of NumPy and SciPy [69].

5. Results and discussion

The objective of this section is to validate reproducible sets of features provided by *FLIRT*. For this purpose, we rely on the publicly available Wearable Stress and Affect Detection (WESAD) dataset and its classification performance as baseline [59]. WESAD contains wearable sensor data from 15 participants going through task-induced emotions. During the study, an Empatica E4 and a RespiBan sensor were used to record the physiological response to the emotions relaxation, amusement, and stress. The Empatica E4 in particular is an established device whose data has been used in a variety of published studies and articles [19]. Since the WESAD dataset defines a clear ML-task and baseline as well as uses a widely established wearable, we will rely on this publicly available dataset for validation.

Our validation follows the original WESAD paper. Nevertheless, we found that reproducibility of their features was limited due to incomplete information about the algorithms as well as a lack of publicly available code. Therefore, we use the results from the

[9] https://flirt.readthedocs.io

Table 9
Classification performance on the WESAD dataset. The mean (standard deviation) of the respective macro F1-score over 5 randomly initialized runs are reported. Note that LDA is a deterministic model and thus no standard deviation is reported. Best mean F1-scores are highlighted in bold.

		Parameters		ML model			
		Artifact detection	**Window selection**	**LDA**	**RF**	**AB**	**DT**
HRV	**WESAD**	–	–	54.72	53.83 (0.11)	53.29 (0.16)	51.15 (0.31)
	FLIRT default	Malik rule	threshold=0.0	**57.59**	57.13 (0.59)	54.85 (0.77)	49.23 (1.12)
	FLIRT threshold	Malik rule	threshold=0.2	53.21	52.28 (0.43)	49.14 (1.36)	47.78 (0.64)
		Data preprocessing	**Decomposition method**				
EDA	**WESAD**	–	–	42.72	45.74 (0.06)	49.06 (0.59)	45.48 (0.17)
	FLIRT integrated	EKF	cvxEDA	51.54	43.46 (0.34)	**51.96 (0.32)**	44.70 (0.45)
	FLIRT modular	B.worth, LR	cvxEDA	50.43	42.17 (0.40)	51.02(0.43)	41.98 (0.32)
		Data preprocessing					
ACC	**WESAD**	-		36.27	46.50 (0.26)	46.38 (1.02)	43.91 (1.16)
	FLIRT	Unfiltered signal		**56.55**	55.58 (0.31)	55.92 (1.02)	52.99 (0.75)
	FLIRT	Butterworth filter		54.93	48.73 (0.24)	48.70 (0.47)	49.32 (0.35)

AdaBoost (AB), accelerometer (ACC), decision tree (DT), electrodermal activity (EDA),extended Kalman filter (EKF), inter-beat interval (IBI), linear discriminant analysis (LDA), logistic regression (LR), machine learning (ML), and random forest (RF).

original paper as baseline. Second, we calculated all features uniformly over a 60 seconds time window,[10] with a sliding window shift of 1/4 seconds, and assigned the corresponding emotion label to each window. For the classification task, we compare four ML models from the original WESAD paper (Random forest (RF), AdaBoost (AB), decision tree (DT), and linear discriminant analysis (LDA)). Our objective is to compare the feature generation between FLIRT and WESAD in particular. Although emotion recognition is a challenging task and might require more complex ML models, we aim to be comparable to the ML pipeline presented in the original WESAD paper, and thus we used the same models, parameters, and implementations. We report classification with the macro F1-score based on a leave-one-subject-out cross-validation to assess model generalization capabilities to unseen subjects. We summarize the results in Table 9.

In general, FLIRT offers many ways to combine algorithms for feature calculation. In this section, we evaluate the basic data processing options from which the user can choose. In the case of HRV, the F1-score increases by 2.87 pp for the overall best performing RF model based on FLIRT's default HRV features without window selection (i.e., threshold = 0). For the EDA modality, the integrated EDA pipeline yields an overall improvement of 2.9 pp based on the LDA model. Finally, the ACC features achieve an improvement of 10.05 pp for the LDA model. Although we used the same ML model hyperparameters as the original WESAD paper, where the parameters were tuned to their feature sets, we achieve overall better results than the WESAD baseline.

For the ACC features, we included FLIRT without filtering ACC data as presented in [59]. It is noticeable that ACC features achieve strong improvements compared to the baseline. We see two reasons for this improved performance. First, FLIRT calculates significantly more features than WESAD (92 vs. 19). Second, emotions in WESAD are task-induced, e.g., stress from public speaking and amusement from watching a video. Therefore, it seems reasonable that our large set of ACC features simply captures the emotion-specific task (sitting vs. gesticulating) rather than the induced emotion. However, there is a high likelihood that even the original WESAD-based ACC classifications makes use of this Clever Hans strategy. In essence, the WESAD setting suggests to avoid the use of ACC features and we only include this analysis to maintain full comparability to WESAD.

[10] In general, WESAD uses 60 seconds as well, however, the authors change the window length for ACC to 5 seconds.

6. Conclusion

In this paper, we presented *FLIRT: A Feature Generation Toolkit for Wearable Data*, a Python package that focuses explicitly on standardized processing of physiological data from commercial wearables, from data cleaning to feature extraction. FLIRT supports researchers to leverage rich sensing data (HRV, EDA, ACC) obtained by wearables to build ML models for various health and behavioral outcome detections. Instead of writing custom code to preprocess, clean, and compute features, which results in a lack of uniformity across studies, FLIRT helps researchers to create fast and reproducible results across disciplines and applications with state-of-the-art methods.

However, we tested FLIRT on a publicly available dataset based on which we demonstrated an increase in classification performance. Whether this increase can be achieved in other applications remains to be seen. However, note that FLIRT enables the reproducibility of the results and contributes to research in this area. Last, applying FLIRT can inherit a challenge that is choosing the parameters of FLIRT. Although we included default parameters tailored to supported wearables (e.g., Empatica E4), the user must manually tune these parameters for yet unsupported wearables.

Soon, we look forward to extending the capabilities of FLIRT, and welcome the community to this open-source project. Future directions include a detailed performance comparison to related Python packages, incorporating different commercial wearable devices (e.g., Apple Watch), and increasing the functionalities with improved algorithms and feature selection capabilities.

Declaration of Competing Interest

T.K. and E.F. are affiliated with the Center for Digital Health Interventions (www.c4dhi.org), a joint initiative of the Department of Management, Technology and Economics at ETH Zurich and the Institute of Technology Management at the University of St. Gallen, which is funded in part by the Swiss health insurer CSS. T.K. and E.F. are also co-founders of Pathmate Technologies, a university spin-off company that creates and delivers digital clinical pathways. However, Pathmate Technologies is not involved in any way. The remaining authors declare that they have no competing interests.

CRediT authorship contribution statement

Simon Föll: Conceptualization, Methodology, Software, Writing – original draft, Project administration. **Martin Maritsch:** Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Federica Spinola:** Software, Visualization, Writing – original draft. **Varun Mishra:** Validation, Writing – review & editing. **Filipe Barata:** Methodology, Writing – review & editing. **Tobias Kowatsch:** Writing – review & editing. **Elgar Fleisch:** Writing – review & editing. **Felix Wortmann:** Methodology, Writing – review & editing, Supervision.

Acknowledgment

This work was part-funded by the Hasler Stiftung, Project 20039.

References

[1] B. Acar, I. Savelieva, H. Hemingway, M. Malik, Automatic ectopic beat elimination in short-term heart rate variability measurement, Comput Methods Programs Biomed 63 (2000) 123–131.
[2] R.U. Acharya, P.K. Joseph, N. Kannathal, C.M. Lim, J.S. Suri, Heart rate variability: a review, Med. Biol. Eng. Comput. 44 (12) (2006) 1031–1051, doi:10.1007/s11517-006-0119-0.
[3] A. Alberdi, A. Aztiria, A. Basarab, Towards an automatic early stress recognition system for office environments based on multimodal measurements: a review, J Biomed Inform 59 (2016) 49–75, doi:10.1016/j.jbi.2015.11.007.
[4] J. Allen, Photoplethysmography and its application in clinical physiological measurement, Physiol Meas 28 (3) (2007) 1–39, doi:10.1088/0967-3334/28/3/R01.
[5] M.R. Amin, R.T. Faghih, Inferring autonomic nervous system stimulation from hand and foot skin conductance measurements, 2018, pp. 655–660.
[6] M.R. Amin, R.T. Faghih, Tonic and phasic decomposition of skin conductance data: A generalized-cross-validation-based block coordinate descent approach, in: 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019, pp. 745–749, doi:10.1109/EMBC.2019.8857074.
[7] R. Avram, J.E. Olgin, P. Kuhar, J.W. Hughes, G.M. Marcus, M.J. Pletcher, K. Aschbacher, G.H. Tison, A digital biomarker of diabetes from smartphone-based vascular signals, Nat. Med. (2020).
[8] D.R. Bach, K.J. Friston, R.J. Dolan, An improved algorithm for model-based analysis of evoked skin conductance responses, Biol Psychol 94 (2013) 490–497, doi:10.1016/j.biopsycho.2013.09.010.
[9] M. Bachler, Spectral analysis of unevenly spaced data: models and application in heart rate variability, SNE Simulation Notes Europe 27 (4) (2017) 183–190, doi:10.11128/sne.27.tn.10393. URL: https://www.sne-journal.org/10383
[10] L. Bao, S.S. Intille, Activity recognition from user-annotated acceleration data, Pervasive Computing, Springer Berlin Heidelberg, 2004, doi:10.1007/978-3-540-24646-6_1.
[11] R. Bartels, hrv, 2020, [Online; accessed October 26, 2020], URL: https://hrv.readthedocs.io/en/latest/.
[12] M.O. Bekkink, M. Koeneman, B.E. de Galan, S.J. Bredie, Early detection of hypoglycemia in type 1 diabetes using heart rate variability measured by a wearable device, Diabetes Care 42 (4) (2019) 689–692, doi:10.2337/dc18-1843.
[13] M. Benedek, C. Kaernbach, Decomposition of skin conductance data by means of nonnegative deconvolution, Psychophysiology 47 (2010) 647–658, doi:10.1111/j.1469-8986.2009.00972.x.
[14] B. Bent, B.A. Goldstein, W.A. Kibbe, J.P. Dunn, Investigating sources of inaccuracy in wearable optical heart rate sensors, npj Digital Medicine 3 (1) (2020) 18, doi:10.1038/s41746-020-0226-6.
[15] A. Bizzego, A. Battisti, G. Gabrieli, G. Esposito, C. Furlanello, Pyphysio: a physiological signal processing library for data science approaches in physiology, SoftwareX 10 (2019) 100287, doi:10.1016/j.softx.2019.100287. URL: http://www.sciencedirect.com/science/article/pii/S2352711019301839
[16] P. Casale, O. Pujol, P. Radeva, Human Activity Recognition from Accelerometer Data Using a Wearable Device BT - Pattern Recognition and Image Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 289–296.
[17] R. Champseix, hrv-analysis, 2020, [Online; accessed October 26, 2020], URL: https://aura-healthcare.github.io/hrvanalysis/.
[18] E.P. Doheny, M.M. Lowery, A. Russell, S. Ryan, Estimation of respiration rate and sleeping position using a wearable accelerometer, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 4668–4671, doi:10.1109/EMBC44109.2020.9176573.
[19] Empatica, Empatica - scientific publications, 2020, [Online; accessed October 26, 2020], URL:https://www.empatica.com/research/publications/.
[20] J. Fridolfsson, M. Börjesson, C. Buck, Ö. Ekblom, E. Ekblom-Bak, M. Hunsberger, L. Lissner, D. Arvidsson, Effects of frequency filtering on intensity and noise in accelerometer-based physical activity measurements, Sensors 19 (9) (2019), doi:10.3390/s19092186.
[21] G. Gabrieli, Pysiology - physiological analysis made easy, 2020, [Online; accessed October 26, 2020], URL: https://pysiology.readthedocs.io/en/latest/.
[22] S. Gashi, E.D. Lascio, B. Stancu, V.D. Swain, V. Mishra, M. Gjoreski, S. Santini, Detection of artifacts in ambulatory electrodermal activity data, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (2020).
[23] P. van Gent, Python heart rate analysis toolkit, 2020, [Online; accessed October 26, 2020], URL: https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/heartrateanalysis.html.
[24] P. Ghaderyan, A. Abbasi, An efficient automatic workload estimation method based on electrodermal activity using pattern classifier combinations, International Journal of Psychophysiology 110 (2016) 91–101, doi:10.1016/j.ijpsycho.2016.10.013.
[25] P. Gomes, Pyhrv - python toolbox for heart rate variability, 2020, [Online; accessed October 26, 2020], URL: https://pyhrv.readthedocs.io/en/latest/index.html.
[26] A. Greco, G. Valenza, A. Lanata, E.P. Scilingo, L. Citi, Cvxeda: a convex optimization approach to electrodermal activity processing, IEEE Trans. Biomed. Eng. 63 (2016) 797–804, doi:10.1109/TBME.2015.2474131.
[27] S. Halem, E. Roekel, L. Kroencke, N. Kuper, J. Denissen, Moments that matter? on the complexity of using triggers based on skin conductance to sample arousing events within an experience sampling framework, Eur J Pers 34 (2020), doi:10.1002/per.2252.
[28] J.A. Healey, R.W. Picard, Detecting stress during real-world driving tasks using physiological sensors, IEEE Trans. Intell. Transp. Syst. 6 (2005) 156–166, doi:10.1109/TITS.2005.848368.
[29] F. Hernando-Gallego, D. Luengo, A. Artes-Rodriguez, Feature extraction of galvanic skin responses by nonnegative sparse deconvolution, IEEE J Biomed Health Inform 22 (2018) 1385–1394, doi:10.1109/JBHI.2017.2780252.
[30] M. Huang, G. Zhao, L. Wang, F. Yang, A pervasive simplified method for human movement pattern assessing, in: 2010 IEEE 16th International Conference on Parallel and Distributed Systems, 2010, pp. 625–628, doi:10.1109/ICPADS.2010.65.
[31] Q.T. Huynh, U.D. Nguyen, B.Q. Tran, Power spectral analyses to detect falls using 3-d accelerometers, in: T. Vo Van, T.A. Nguyen Le, T. Nguyen Duc (Eds.), 6th International Conference on the Development of Biomedical Engineering in Vietnam (BME6), Springer Singapore, Singapore, 2018, pp. 191–195.
[32] N.C. Jacobson, H. Weingarden, S. Wilhelm, Digital biomarkers of mood disorders and symptom change, npj Digital Medicine 2 (1) (2019) 3, doi:10.1038/s41746-019-0078-0.
[33] E. Jovanov, Preliminary analysis of the use of smartwatches for longitudinal health monitoring, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 865–868, doi:10.1109/EMBC.2015.7318499.
[34] M. Kamath, E. Fallen, Correction Ofthe Heart Rate Variability Signal for Ectopics and Missing Beats, in: M. Malik, A. Camm (Eds.), Heart Rate Variability, Futura Pub. Co. Inc., Armonk, N.Y., 1995, pp. 75–85.
[35] M. Karlsson, R. Hörnsten, A. Rydberg, U. Wiklund, Automatic filtering of outliers in RR intervals before analysis of heart rate variability in holter recordings: a comparison with carefully edited data, Biomed Eng Online 11 (1) (2012) 2, doi:10.1186/1475-925X-11-2.
[36] K.H. Kim, S.W. Bang, S.R. Kim, Emotion recognition system using short-term monitoring of physiological signals, Medical & Biological Engineering & Computing 42 (2004), doi:10.1007/BF02344719.
[37] I.R. Kleckner, R.M. Jones, O. Wilder-Smith, J.B. Wormwood, M. Akcakaya, K.S. Quigley, C. Lord, M.S. Goodwin, Simple, transparent, and flexible automated quality assessment procedures for ambulatory electrodermal activity data, IEEE Trans. Biomed. Eng. 65 (2018) 1460–1467.
[38] P. Laguna, G. Moody, R. Mark, Power spectral density of unevenly sampled data by least-square analysis: performance and application to heart rate signals, IEEE Trans. Biomed. Eng. 45 (6) (1998) 698–715, doi:10.1109/10.678605.
[39] O.D. Lara, M.A. Labrador, A survey on human activity recognition using wearable sensors, IEEE Communications Surveys Tutorials 15 (3) (2013) 1192–1209, doi:10.1109/SURV.2012.110112.00192.
[40] F. Larradet, R. Niewiadomski, G. Barresi, D.G. Caldwell, L.S. Mattos, Toward emotion recognition from physiological signals in the wild: approaching the methodological issues in real-Life data collection, Front Psychol 11 (2020) 1111, doi:10.3389/fpsyg.2020.01111. URL: https://www.frontiersin.org/article/10.3389/fpsyg.2020.01111
[41] N. Lippman, K.M. Stein, B.B. Lerman, Comparison of methods for removal of ectopy in measurement of heart rate variability, American Journal of Physiology-Heart and Circulatory Physiology 267 (1) (1994) H411–H418, doi:10.1152/ajpheart.1994.267.1.H411. PMID: 7519408
[42] N.R. Lomb, Least-squares frequency analysis of unequally spaced data, Astrophys Space Sci 39 (2) (1976) 447–462, doi:10.1007/BF00648343.
[43] D. Makowski, T. Pham, Z.J. Lau, J.C. Brammer, F. Lespinasse, H. Pham, C. Schölzel, S.H. Annabel Chen, Neurokit2: A python toolbox for neurophysiological signal processing, Zenodo, 2020, doi:10.5281/ZENODO.3597887. URL: https://github.com/neuropsychology/NeuroKit
[44] M. Malik, J.T. Bigger, A.J. Camm, R.E. Kleiger, A. Malliani, A.J. Moss, P.J. Schwartz, Heart rate variability: standards of measurement, physiological interpretation, and clinical use, Eur. Heart J. 17 (3) (1996) 354–381, doi:10.1093/oxfordjournals.eurheartj.a014868.
[45] M. Maritsch, C. Bérubé, M. Kraus, V. Lehmann, T. Züger, S. Feuerriegel, T. Kowatsch, F. Wortmann, Improving heart rate variability measurements from consumer smartwatches with machine learning, in: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Comput-

ing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, in: UbiComp/ISWC '19 Adjunct, Association for Computing Machinery, New York, NY, USA, 2019, pp. 934–938, doi:10.1145/3341162.3346276.
[46] M. Maritsch, S. Föll, V. Lehmann, C. Bérubé, M. Kraus, S. Feuerriegel, T. Kowatsch, T. Züger, C. Stettler, E. Fleisch, et al., Towards wearable-based hypoglycemia detection and warning in diabetes, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–8.
[47] J.E. Mietus, C.-K. Peng, I. Henry, R.L. Goldsmith, A.L. Goldberger, The pnnx files: re-examining a widely used heart rate variability measure, Heart 88 (4) (2002) 378–380, doi:10.1136/heart.88.4.378.
[48] V. Mishra, S. Sen, G. Chen, T. Hao, J. Rogers, C.-H. Chen, D. Kotz, Evaluating the reproducibility of physiological stress detection models, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (4) (2020), doi:10.1145/3432220.
[49] G. Moody, Spectral analysis of heart rate without resampling, in: Proceedings of Computers in Cardiology Conference, IEEE Comput. Soc. Press, 1993, pp. 715–718, doi:10.1109/CIC.1993.378302.
[50] D. Morelli, A. Rossi, M. Cairo, D.A. Clifton, Analysis of the impact of interpolation methods of missing RR-intervals caused by motion artifacts on HRV features estimations, Sensors 19 (14) (2019) 1–14, doi:10.3390/s19143163. URL: https://www.ncbi.nlm.nih.gov/pubmed/31323850 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6679245/ https://www.mdpi.com/1424-8220/19/14/3163
[51] M. Nabian, Y. Yin, J. Wormwood, K.S. Quigley, L.F. Barrett, S. Ostadabbas, An open-source feature extraction tool for the analysis of peripheral physiological data, IEEE J Transl Eng Health Med 6 (2018), doi:10.1109/JTEHM.2018.2878000.
[52] A. Natarajan, H.-W. Su, C. Heneghan, Assessment of physiological signs associated with COVID-19 measured using wearable devices, npj Digital Medicine 3 (1) (2020) 156, doi:10.1038/s41746-020-00363-7.
[53] B.W. Nelson, C.A. Low, N. Jacobson, P. Areán, J. Torous, N.B. Allen, Guidelines for wrist-worn consumer wearable assessment of heart rate in biobehavioral research, npj Digital Medicine 3 (1) (2020) 90, doi:10.1038/s41746-020-0297-4.
[54] J. Nordh, K. Berntorp, B. Nizette, pyparticleest, 2018, URL:https://github.com/jerkern/pyParticleEst.
[55] C. Orphanidou, T. Bonnici, P. Charlton, D. Clifton, D. Vallance, L. Tarassenko, Signal quality indices for the electrocardiogram and photoplethysmogram: derivation and applications to wireless monitoring, IEEE J Biomed Health Inform 19 (3) (2014), doi:10.1109/JBHI.2014.2338351. 1–1, URL: http://ieeexplore.ieee.org/document/6862843/
[56] H.F. Posada-Quintero, K.H. Chon, Innovations in electrodermal activity data collection and signal processing: A systematic review, volume 20, MDPI AG, 2020.
[57] J.D. Scargle, Studies in astronomical time series analysis. II - statistical aspects of spectral analysis of unevenly spaced data, Astrophys. J. 263 (1982) 835, doi:10.1086/160554.
[58] T. Schaffer, B. Hensel, C. Weigand, J. Schüttler, C. Jeleazcov, Evaluation of techniques for estimating the power spectral density of RR-intervals under paced respiration conditions, J Clin Monit Comput 28 (5) (2014) 481–486, doi:10.1007/s10877-013-9447-4.
[59] P. Schmidt, A. Reiss, R. Duerichen, K.V. Laerhoven, Introducing wesad, a multimodal dataset for wearable stress and affect detection, ICMI 2018 - Proceedings of the 2018 International Conference on Multimodal Interaction (2018) 400–408, doi:10.1145/3242969.3242985.
[60] P. Schwab, W. Karlen, A deep learning approach to diagnosing multiple sclerosis from smartphone data, IEEE J Biomed Health Inform 25 (4) (2021) 1284–1291, doi:10.1109/JBHI.2020.3021143.
[61] F. Shaffer, J.P. Ginsberg, An overview of heart rate variability metrics and norms, Front Public Health 5 (2017) 1–17, doi:10.3389/fpubh.2017.00258. URL: http://journal.frontiersin.org/article/10.3389/fpubh.2017.00258/full
[62] L. Smital, C.R. Haider, M. Vitek, P. Leinveber, P. Jurak, A. Nemcova, R. Smisek, L. Marsanova, I. Provaznik, C.L. Felton, B.K. Gilbert, D.R.H. III, Real-Time quality assessment of long-term ECG signals recorded by wearables in free-living conditions, IEEE Trans. Biomed. Eng. 67 (10) (2020) 2721–2734, doi:10.1109/TBME.2020.2969719.
[63] J.P. Spiers, B. Silke, U. McDermott, R.G. Shanks, D.W.G. Harron, Time and frequency domain assessment of heart rate variability: a theoretical and clinical appreciation, Clinical Autonomic Research 3 (2) (1993) 145–158, doi:10.1007/BF01819000.
[64] S. Taylor, N. Jaques, W. Chen, S. Fedor, A. Sano, R. Picard, Automatic identification of artifacts in electrodermal activity data, 2015a, (https://github.com/MITMediaLabAffectiveComputing/eda-explorera).
[65] S. Taylor, N. Jaques, W. Chen, S. Fedor, A. Sano, R. Picard, Automatic identification of artifacts in electrodermal activity data, volume 2015-November, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 1934–1937, doi:10.1109/EMBC.2015.7318762.
[66] Instituto de Telecomunicacoes, Biosppy - biosignal processing written in python, 2020, [Online; accessed October 26, 2020], URL: https://biosppy.readthedocs.io/en/stable/.
[67] C. Tronstad, O.M. Staal, S. Saelid, O.G. Martinsen, Model- based filtering for artifact and noise suppression with state estimation for electrodermal activity measurements in real time, volume 2015-November, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 2750–2753, doi:10.1109/EMBC.2015.7318961.
[68] H.J. Van Dellen, J. Aasman, L.J.M. Mulder, G. Mulder, Time domain versus frequency domain measures of heart-rate variability, in: J.F. Orlebeke, G. Mulder, L.J.P. Van Doornen (Eds.), Psychophysiology of Cardiovascular Control. Models, Methods, and Data, Plenum Press, New York, 1985, pp. 353–374.
[69] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, Scipy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (2020) 261–272, doi:10.1038/s41592-019-0686-2.
[70] W.-Z. Wang, B.-Y. Huang, L. Wang, Analysis of filtering methods for 3d acceleration signals in body sensor network (2011). 10.1109/ISBB.2011.6107697
[71] C.-C. Yang, Y.-L. Hsu, A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring, Sensors, 2010, doi:10.3390/s100807772.
[72] Y. Zhang, M. Haghdan, K.S. Xu, Unsupervised motion artifact detection in wrist-measured electrodermal activity data. arXiv:1707.08287.