ALPS: A Web Platform for Analysing Multimodal Sensor Data in the Context of Digital Health

Felix Musmann
Digital Health Center
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
felix.musmann@student.hpi.de

Ariane Morassi Sasso
Digital Health Center
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
ariane.morassi-sasso@hpi.de

Bert Arnrich
Digital Health Center
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
bert.arnrich@hpi.de

***Abstract*—The Internet of Things (IoT) enables us to record a vast amount of information about activities, the environment and the physiological state of a person. In particular, wearables promise the development of new methods for prevention and treatment of diseases. Clinical studies often involve multiple devices from different manufacturers, which make use of different data formats and usually offer no way to synchronize them. Additionally, existing analysis tools are often tailored to a very specific use case. Thus, professionals working with data collection and analysis execute a lot of manual work to gather and combine the recorded data. This paper presents ALPS, an extensible web platform with an integrated event-based synchronization method that enables researchers with clinical and engineering background to analyze multimodal sensor data. Plug-ins for new devices, filtering and analysis methods allow the customization for different research scenarios. A case study on Heart Rate Variability (HRV) shows that the platform simplifies the comparative analysis of multiple signals and supports the exploration of data from different wearables.**

***Index Terms*—wearable computers, electronic healthcare, web services, information management, time series analysis, synchronization**

I. Introduction

The rising numbers in sales [1] show that wearables are starting to become ubiquitous in everyday life and are able to record a variety of motion, environmental and physiological signals. These signals can be recorded continuously without effort of the user. Additionally, companies constantly develop new devices for consumers, trying to increase data quality, comfort and diversity of sensors. By the combination of these, a more holistic image of a person and the environment can be recorded. Therefore, facilitating the monitoring of health conditions such as chronic diseases, which are widely recognized as the dominant burden of disease worldwide [2].

Among them, the group of Cardiovascular Diseases (CVDs) is the leading cause of death. It includes all heart and circulatory diseases, e.g., strokes, heart attacks, heart failure and hypertension [3]. Risk factors such as diabetes, high blood pressure, and obesity are related by their metabolic origin and often linked to an unhealthy lifestyle, which is characterized by an unbalanced diet and low physical activity. These risk factors are also part of the criteria for the Metabolic Syndrome (MetS), which has become one of the main reasons behind an epidemic of CVDs in developed and developing countries [4]. To manage the MetS, the International Diabetes Federation [4] recommends lifestyle interventions, which consist of calorie restriction, an increase in physical activity and a change in diet. However, they often report high dropout rates [5] and predictors for good adherence vary from seeing success early on to starting the intervention with a lower Body Mass Index (BMI) [6]. Furthermore, participants need to maintain the changes after the intervention ends.

To monitor the success or relapse of a participant with MetS during an intervention, a variety of parameters can be collected. All of these parameters must be collected actively and not all of them can be collected by the participant. These usually are weight, waist circumference, glucose level and blood pressure [4]. Heart rate and Heart Rate Variability (HRV) can extend this set of parameters. MetS has been linked to the dysfunction of the Autonomic Nervous System (ANS), which can result in elevated heart rate and diminished HRV indices [7]. HRV indices are calculated from the variation in the time interval between heartbeats, known as Interbeat Intervals (IBIs). For this, the time between two adjacent R-peaks of an Electrocardiography (ECG) recording is used, but several studies have shown a high correlation between HRV and Pulse Rate Variability (PRV) indices [8]. Pulse rate and Pulse Rate Variability (PRV) of a person, among other indices, can be obtained from a Photoplethysmography (PPG) sensor, which is present in many wearables and can measure changes in the blood volume. Thus, wearables with PPG might be a viable tool to monitor lifestyle interventions, but require a comparative analysis between devices, among other things.

Clinical studies are already using multiple devices on their subjects. Due to technical requirements and positioning of individual sensors, sometimes, it is not possible to combine them in one physical device. These independent devices have to be synchronized with high precision to allow a mean-

This research has been partly funded by the Federal Ministry of Education and Research of Germany in the framework of KI-LAB-ITSE (project number 01IS19066).

ingful combined analysis of their recorded signals. Since interoperability of devices from different manufacturers is still lacking, researchers fallback to develop their own solutions for synchronization or do it manually. The handling of different data formats, system times, sampling frequencies and clock drifts is a time demanding task that is done before any analysis.

Data quality is another challenge for wearables, since only accurate recordings allow a sound analysis [9]. Often, the fastest way to make a first estimation on data quality is to leverage the human capabilities in pattern recognition by visualizing and looking at the data. Thereby, artifacts such as data points that are missing or that are out of the expected range can be spotted and extracted for further inspection. During data exploration, the combination of the signals allows for a deeper understanding of the person and the context.

This paper presents a web platform for signal analysis, that offers solutions for the aforementioned problems of data management, synchronization, visualization and evaluation to support researchers in clinical studies with different devices. It also demonstrates its importance on an use case related to the calculation of HRV indices from ECG and PPG sensors. ALPS is open source[1] and built to be extended in terms of supported data formats, filters and analysis methods. In the repository, a link to a demo and short videos demonstrating the user interface are available. The paper is structured as follows: section II gives an overview of wearables used in digital health and how they can be synchronized as well as currently available tools to analyze HRV. Section III shows the features and architecture of ALPS, while section IV briefly describes a case study with HRV analysis. Lastly, sections V and VI present the benefits, limitations and next steps for the platform.

II. Background

A. Wearables in Digital Health

The broad term wearables entails a variety of items that can be attached to the body and can contain electronics, mechanical technologies or functional materials [10]. Modern wrist-worn fitness trackers are probably the first items that come to mind when thinking about wearables. This section will focus on these non-invasive devices, that are used to monitor specific vital signs, biomarkers, activity and environmental context information of the user, frequently or continuously, in everyday life. They can not only sense, but also process, store, transmit and use the obtained data for feedback or actions [11]. Miniaturization and declining costs in hardware production have made wearables more affordable and capable, making them more attractive to consumers as well as researchers. The sales figures of wearables worldwide have increased more than sixfold from 28.8 millions in 2014 to 172.2 millions in 2018 alone [12]. The promise of wearables for researchers and clinicians is to improve clinical trials, preventive measures and treatments. Wearable devices can contain multiple sensors for activity, environmental, and physiological sensing [13] that supplement each other and thus create a more holistic recording of a person.

1) Activity and Environmental Sensors: To track motion or gestures, the accelerometer counts as one of the most used sensors, which is able to measure acceleration in one or more axes. Gyroscopes are used to keep track of the orientation of a device or measuring its angular velocity, which specifies how fast an object is rotating. A magnetometer measures a magnetic field and can be used to determine the earth's magnetic north and the devices orientation to it. The combination of accelerometers and gyroscopes is also known as an Inertial Measurement Unit (IMU) and often complemented by a magnetometer [14]. Devices incorporating these sensors can be placed on different body parts to classify a multitude of body movements and activities, such as walking, lying and standing [15]. Moreover, a Surface Electromyography (sEMG) allows to evaluate muscle activity. Therefore, electrodes are placed on the skin above the muscle, which record the electrical activity of the skeletal muscle. Flexible electronics have been successfully applied to monitor mechanical signals, e.g., for strain and pressure sensing on the sole of the foot or for diagnosing Parkinson's disease [16].

Activities can also be detected by using the Global Positioning System (GPS). The continuous recording of the position makes it possible to deduce speed and orientation. Together with information resources, like maps or data from air quality measuring stations, it can be used to gather additional information about the environment. GPS is limited to outdoor usage, but air quality can - to some degree - also be assessed directly by a wearable device. Several types of sensors exist to measure the humidity and temperature of the air [13]. Gas sensors can measure the concentration of volatile organic compounds (VOC) to rate indoor air quality [17]. Barometric pressure sensors can be used to determine the altitude and change in vertical position, which is also valuable for activity detection.

2) Physiological Sensors: Heart rate is one of the most important vital signs that a majority of wearables obtain by using a Photoplethysmography (PPG) sensor. The PPG signal reflects the change of blood volume in living skin tissue. With each cardiac cycle, blood is pumped through the body and causes arteries and arterioles to distend and create a peak in the signal. A light-emitting diode (LED) shines light at the skin and a photodiode measures the reflected or transmitted light. Reflective PPG sensors usually use green or yellow LEDs, whereas transmissive ones use red or near-infrared LEDs because of their higher penetration [18]. Strictly speaking, only the pulse rate can be deduced from PPG. Under normal conditions it is synchronized with the heart rate, but individuals with a heart condition, e.g., atrial flutter, can have a heart rate differing from the pulse rate [19]. PPG containing red and near-infrared LEDs are also used to measure the oxygen saturation of the blood (SpO_2) [20].

Nevertheless, ECG is the gold standard to measure heart rate and therefore, the contractions of the heart. Each contraction of the heart generates a change in electrical voltage, which can be measured on the body surface. Electrodes are placed on the

[1] https://github.com/hpi-dhc/alps

torso that are configured as bipolar leads, one electrode being the positive input and the other, the negative. The difference in electric potential between two electrodes is recorded. For a clinical ECG, usually 10 electrodes are used to form a 12-lead configuration [21]. Since this configuration is not viable for wearables that are targeted to consumers, a 1-lead configuration with two electrodes is commonly used for basic heart rate readings and even some arrhythmia. A possible form for continuous 1-lead ECG reading can be a chest strap, but short-term readings can also be done with wrist-worn devices. The user therefore has to touch the device with the opposite hand to create the lead from arm to arm.

The flow of electricity through the skin can also be measured and it is known as Electrodermal Activity (EDA). It can be measured on the skin with a weak electrical charge applied between two electrodes close to each other. It has been used as an indicator for stress and emotional activity. Body temperature sensors can add additional context and improve this analysis [22]. Moreover, the ability to analyze sweat continuously promises a multitude of new biomarkers [23]. Companies like Eccrine Systems[2] are developing wearables that stimulate the skin locally to produce sweat and, that contain a management system for the fluids to avoid contamination. While new sensors are being developed, non-invasive wearables are able to sense a variety of physiological signs with available technology.

B. Synchronization for Multimodal Analysis

Before any meaningful analysis of multiple devices, they have to be synchronized with high precision. To achieve this, a synchronization method should be used and it needs to solve multiple problems. Each device contains a computer clock, possibly running at a different frequency and local time. Additionally, the frequency of a clock might not be stable over the time of a recording. This is known as jitter for short-term or drift for long-term variations [24]. Since resources in computation, memory and power are limited on wearables, real-time solutions must be designed in an appropriate way. Many researchers have come up with solutions that are able to synchronize individual sensors, such as Wireless Body Area Networks (WBAN) or Body Sensor Networks (BSN) [25].

Although a WBAN should be the goal - especially for consumer grade products - no standard has been defined that is widely adapted by manufacturers. This poses an obstacle to research, since devices used in experiments and clinical trials might record data locally or stream it via Bluetooth to a proprietary software. When devices are not able to communicate with each other, an offline synchronization needs to be applied afterwards. A trivial approach for devices recording the same physiological signal would be to simply shift the signals to achieve the highest cross-correlation, but this is rarely the case. Another approach is an event-based synchronization, in which the same event is recorded by each device [26]. This can be a sound, force or motion event, dependent on the available sensors. An algorithm that spots these events can use them to eliminate not only the shift in local times, but also clock drift.

[2] https://www.eccrinesystems.com/

C. HRV Tools

Despite the fact that there are numerous software solutions for HRV analysis available, only few come free of charge or are open source and include a Graphical User Interface (GUI). Without a GUI it is necessary for the user to have at least a minimal set of programming skills. This leads to studies using a variety of proprietary HRV analysis software with different algorithms for preprocessing the data and computing parameters. Because there are a multitude of possible configurations for computing HRV indices, studies do not always report them [34]. As a result, the reproducibility of a study is impaired. One solution for this issue is the use of open source software, which give not only transparency to the utilized algorithms, but also the ability to reconstruct study results more easily.

Libraries with a public source code already exist for multiple programming languages, e.g. pyHRV [35] for Python, RHRV [34] for R or PhysioNet Cardiovascular Signal Toolbox [36] for Matlab. There are also available tools that are open source and have a GUI such as gHRV [29] and HRVAS [30], but both only accept IBI series as input, which makes an external preprocessing necessary. Therefore, the most popular software with a GUI is Kubios HRV [32], which is available in a limited free version and is considered to be the *ad hoc* gold standard of the field. Although its developers give some insight into the used algorithms, it is not open source nor extensive. Another software, LabVIEW [33] offers a comprehensive visual programming environment for system design, although not focused on HRV analysis. It can directly interface with ECG devices, but is also the most expensive of the analyzed software solutions and might therefore be improper. Lastly, there is ARTiiFact [28], which has a simple GUI and focuses on the removal of artifacts from both ECG and IBI time series.

For the analysis of HRV, time segments of the recorded signal have to be selected. This can be the IBI signal or even the original ECG or PPG, if it is supported by the software. Kubios and gHRV support the selection of multiple segments, that can be analyzed individually. Leveraging this feature, gHRV is the only software that implements significance tests between different time segments. With ARTiiFact the user can select only one segment at a time for analysis. HRVAS always analyzes the complete input, which requires an external preselection. While none of the open source solutions provide a beat detection from ECG or PPG data, Kubios is one of few software that supports both, but in a paid premium version. Batch processing is available in HRVAS and ARTiiFact only, in which multiple preprocessed IBI files can be selected at once. All the other solutions can analyze only one input file at a time and export the results. Moreover, all analyzed software lack the ability to easily examine HRV data of groups or multiple signals at once.

In conclusion, existing tools already offer a variety of features to analyze HRV with a GUI, but closed source solutions are lacking in full transparency on the used algorithms. Open

Name	Platform	Frequency Domain	Non-linear Domain	Beat Detection	Open Source
aHRV [27]	Windows	CZF, Fourier, AR	Poincaré	ECG	No
ARTiiFact [28]	Windows (Matlab)	Fourier	No	ECG	No
gHRV [29]	Linux, Windows, macOS (Matlab)	Fourier	Poincaré, ApEn, Fractal dimension	No	Yes
HRVAS [30]	Linux, Windows, macOS (Matlab)	Fourier, AR, Lomb	Poincaré, SampEn, DFA	No	Yes
Kardia [31]	Linux, Windows, macOS (Matlab)	Fourier, AR	DFA	No	Yes
Kubios [32]	Linux, Windows, macOS (Matlab)	Fourier, AR, Lomb	Poincaré, DFA, ApEn, SampEn, RPA, MSE	ECG, PPG	No
LabVIEW [33]	Linux, Windows, macOS	Fourier, AR, Wavelet	Poincaré, DFA, RPA	ECG	No

TABLE I: Selection of available HRV analysis tools with graphical user interface. Tools developed with Matlab might either be compiled from source, if available, or downloaded for the appropiate system.

source solutions require at least some preprocessing of the data for either compatibility or even selection of time segments. Additionally, none of the presented software are able to display data from other devices or sensors, such as accelerometer or temperature, to gain more insight into the data or offer an easy extension of functionality.

III. System Design

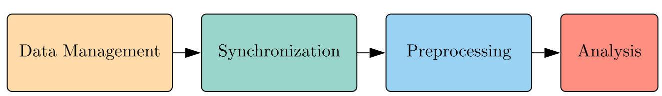


Fig. 1: Workflow of the web platform.

The general workflow of ALPS is divided into four main steps (depicted in Fig. 1), which guide the user from data upload to the analysis of results. This section will describe each of them.

A. Data Management

Data management refers to the way the collected data is organized and stored. It is important that studies have a consistent data structure, especially for studies with large amounts of data, i.e., that are recorded over a long period of time with multiple stakeholders, e.g., investigators, student assistants, clinicians, etc. In that sense and to make it easier to navigate large amounts of data, the platform provides a simple predefined structure as depicted in Fig. 2. The first level of the structure is the *subject* level, which can have multiple *sessions*, e.g., a subject can have recordings on multiple days. Each *session* can contain many *datasets* that are derived from, for example, different wearables and each *dataset* can possess data from multiple sensors in a wearable such as data from ECG, PPG, accelerometer, temperature, etc.

B. Synchronization

To be able to explore and analyze signals from multiple devices, they need to be synchronized. The synchronization process aims at eliminating the differences in system time and clock drift of the devices, which occur if the processors run independently. A lot of research and development has gone into messaging protocols that try to synchronize devices wirelessly [25]. Nevertheless, it is common for a study to use devices that are not able to communicate with each other.

Sessions with more than one dataset can be synchronized manually or by using an event-based algorithm to automatically calculate the parameter for time shift and the stretch factor. The algorithm implemented is specialized in detecting shakes in the accelerometer signal at the beginning and end of a recording. First it needs information about the characteristic of the shake. To compensate the different value ranges of the accelerometers, the magnitude is normalized. Since only shakes at the beginning and end are considered, a time window has to be provided in seconds. Furthermore, a threshold for the minimum peak height, the maximal distance between two adjacent peaks, and the minimum number of peaks in a shake is needed and can be adjusted. The algorithm checks every peak, that is above the threshold, in terms of its distance to the previous peak and if it falls into the start or end window of the signal, to merge them into peak sequences. Sequences with a length below a minimum number of peaks for a shake are filtered out. Finally, the sequence with the highest weight, which is defined by the sum of median and mean of each sequence, is selected as the shake for each time window respectively. Fig. 3 shows the peak sequences of two normalized and unsynchronized shake events. The time

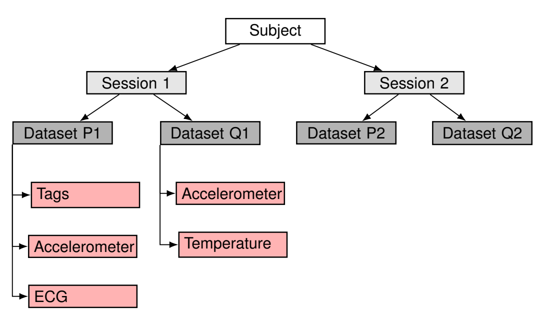


Fig. 2: The hierarchical data structure of the web platform is composed of subjects (white), sessions (light grey, e.g., recordings before and after intervention), datasets (dark grey), and signals (red).

shift between the shake events is calculated by maximizing the cross-correlation between two corresponding shake events. Additional stretching of the signals can compensate possible clock drift. A pseudocode of the algorithm can be found in the appendix (subsection A and subsection B).

Fig. 4 shows the synchronization screen. The graph shows the selected signals and can be zoomed. On the left side of the screen, the signals for each dataset of the session and the reference signal can be selected. The reference signal is used for the pairwise synchronization and its beginning is the zero point for the stretching of the signals. The stretch factor and time shift can be adjusted for each dataset.

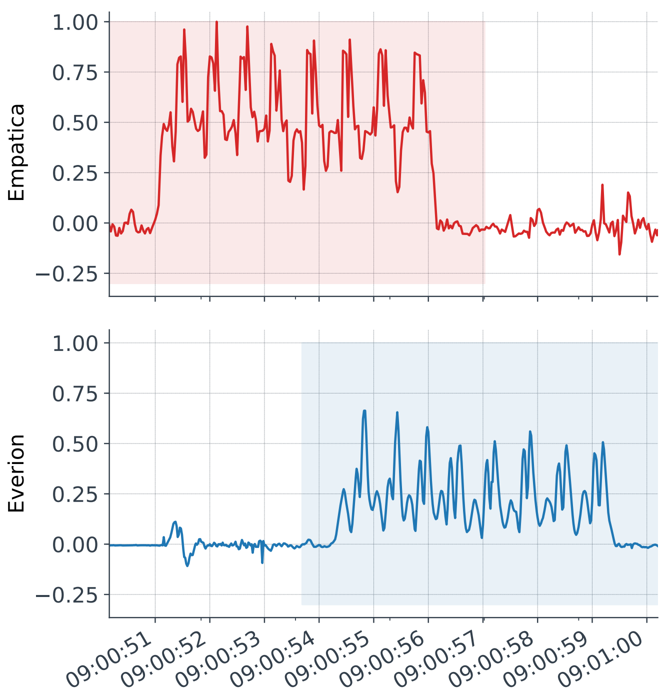


Fig. 3: Detected shake events of two unsynchronized accelerometer signals.

C. Signal Exploration

Since ALPS is capable of handling datasets from multiple devices, a greater variety of signals can be visualized at once. This gives the user the possibility to understand the context of an otherwise isolated time series. In the case of HRV analysis, the PPG sensor is very susceptible to motion. Therefore, high amplitudes in the accelerometer signal are a good indicator for possible inaccuracies in the calculated IBI values and might explain problems with the recording. Fig. 5 shows how the IBI values from the Faros™180[3] ECG device, the EDA of the wristband Empatica E4[4] and the accelerometer of the armband Biovotion Everion[5] can be displayed in sync on the *Preprocessing* screen.

[3]https://www.bittium.com/medical/bittium-faros
[4]https://www.empatica.com/en-eu/research/e4/
[5]https://www.biovotion.com/everion/

The GUI offers a variety of features to interactively explore the signals. Signals from any dataset in the session can be added for visualization. If a signal was downsampled for visualization, it will be marked as such. Downsampling occurs, if the currently visualized excerpt has more than 2000 samples. Mean as well as maximum and minimum values are computed for every resulting bin. This is done to preserve more information about the characteristic of the signal, since the mean alone suppresses outliers in the time series, which are valuable for visual inspection. Each graph can be configured to display dots for each data point ● and to fix the y axis to the minimum and maximum values of the signal ⇕. Three interaction modes allow to zoom into the signal 🔍 by marking the respective section inside any of the displayed graphs, drag a signal ✋ along the x-axis, and label samples for analysis 🏷. As all graphs share the domain of the x axis, each signal will be zoomed or moved to the same position in time. Labels are created per user, can be named as desired and reused across sessions and subjects. Each label can have multiple analysis samples per session, e.g., a label called *Sleeping Phase*. Thereby, artifacts or longer sections with low data quality can be excluded from the analysis. Analysis samples are displayed in all graphs with a color associated to the corresponding label. Additionally, one of the tag information from the datasets can be selected and the individual tags can be visualized as red vertical lines with their text value.

D. Analysis and Result Comparison

An analysis can be performed on one or multiple signals (see Fig. 6). A signal graph is placed at the top of the screen with the results below. Each analysis method is displayed with the results of all selected signals, which can consist of a table of parameters and an interactive plot for each signal. The used configuration of the analysis method is saved and can be viewed in retrospect. Analysis results can be downloaded as a Comma Separated Values (CSV) file. The export of analysis results for more than one session or subject is available from the subject list screen after log in. All selected analysis methods are executed on each selected signal and one of the defined labels with the corresponding signal segments. Parameters of the analysis methods can be configured. When executing the analysis, previous results will be deleted. If multiple signals have been selected, the table with the result parameters will pivot and display a row for each signal. This allows the user to easily compare individual HRV indices like the Low Frequency/High Frequency (LF/HF) ratio from the IBI series of an ECG and PPG signal.

E. Architecture

The back end is responsible for the business logic of data management and processing the time series data. Fig. 7 gives an overview of all the services running on the back end, which are explained in the following. The main component of the back end is a Django [37] instance with PostgreSQL [38], that provides the Web Application Programming Interface (API) and thus makes its functionalities available to any

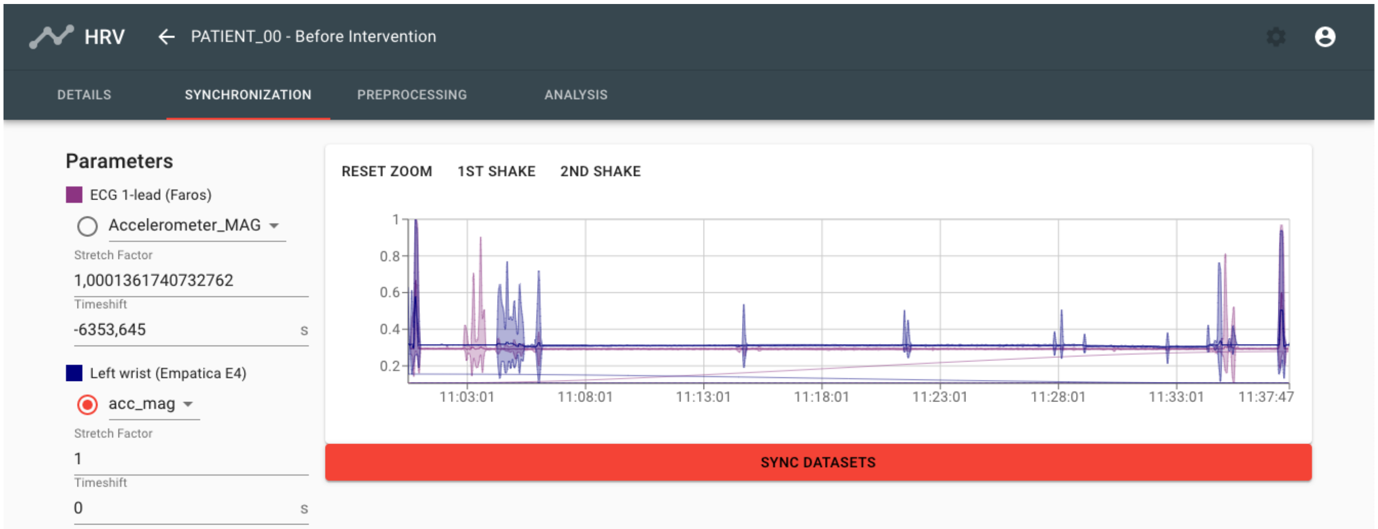


Fig. 4: Screen to synchronize datasets with automatically calculated parameters for time shift and stretching of the datasets on the left and a preview of the two synchronized accelerometer signals on the right.

compatible client. Since one design goal of the back end was easy extensibility, everything is open source and the main components are written in Python, which has become the most popular language among data scientist [39]. For the in-memory handling and analysis of time series data the application makes use of the Pandas library [40]. To cope with procedures executed by the back end, that can be computationally heavy, e.g., the parsing of large datasets and complex analysis, a task queue is necessary, that processes incoming requests. This is accomplished by Celery [41] in conjunction with Redis [42] as the message broker. Celery manages the workers, that process the actual tasks. Workers can have multiple processes and be spread across multiple servers for good scalability depending on the workload. Uploaded data is parsed by the corresponding plug-in (see subsection III-F) and can be saved in a database or in a binary Apache Parquet[6] file, which proved to be a good compromise between reliability, performance and compression rate during development. The front end has been implemented as a Single-page Application (SPA) with React [43], React Router [44], and Redux [45] as main technologies. To provide a familiar user interface Material-UI [46], a React implementation of Google's Material design guidelines, is used. In order to display large time series data, only the currently displayed segment is requested from the back end and downsampled, if it contains more samples than defined by an arbitrarily fixed threshold. All services are containerized and managed by Docker [47].

F. Plug-Ins

Since there is a vast amount of wearable devices available and new ones are constantly being developed, a key design goal of ALPS was to provide a simple way to extend its compatibility with new file formats as well as to add new filtering and analysis methods. That is why these functionalities have been developed with a basic plugin architecture. As depicted in Fig. 8, there are interfaces defined by abstract base classes that need to be implemented by new plugins. All plugins are imported automatically by registries for sources as well as filter and analysis methods. A base registry provides the functionality of discovering subclasses of the respective `base_class`. A new source is responsible for providing information about expected files via `fileOptions()` and to parse these files to save them in a standardized way. Filter and analysis methods can declare a set of configuration parameters with `options()` and need to `process()` the provided signal samples, which can be a full signal or multiple labeled analysis samples (excerpts).

To add a plugin for a new source, it has to be a child class of the `SourceBase` class. Each source needs to define its name by returning the appropriate string from the `name()` method. The `fileOptions()` method should return a list of dictionaries, where each dictionary is the definition of a file, that can be uploaded by the user for that source. A file is defined by a `label` and optional `description`, that is shown to the user in the front end. A `pattern` can be declared for file validation, and two boolean flags to define if this file is `required` and a `timestamp` can be provided by the user. The application will call the `parse()` method after initializing the source. During initialization, references to the uploaded files will be provided as a list of `RawFile` instances via the `raw_files` attribute. A dictionary is expected as return value of `parse()`. It contains a dictionary for each signal under its name as property, that has been extracted from the files. The values of the signal have to be passed as a `pandas.Series` with a timezone-aware `DateTimeIndex`. Empty rows of the series will not be saved. Additionally, the `frequency` in Hz, a `unit` string and the `raw_file_id` can be passed optionally. The `type`

[6]https://parquet.apache.org/

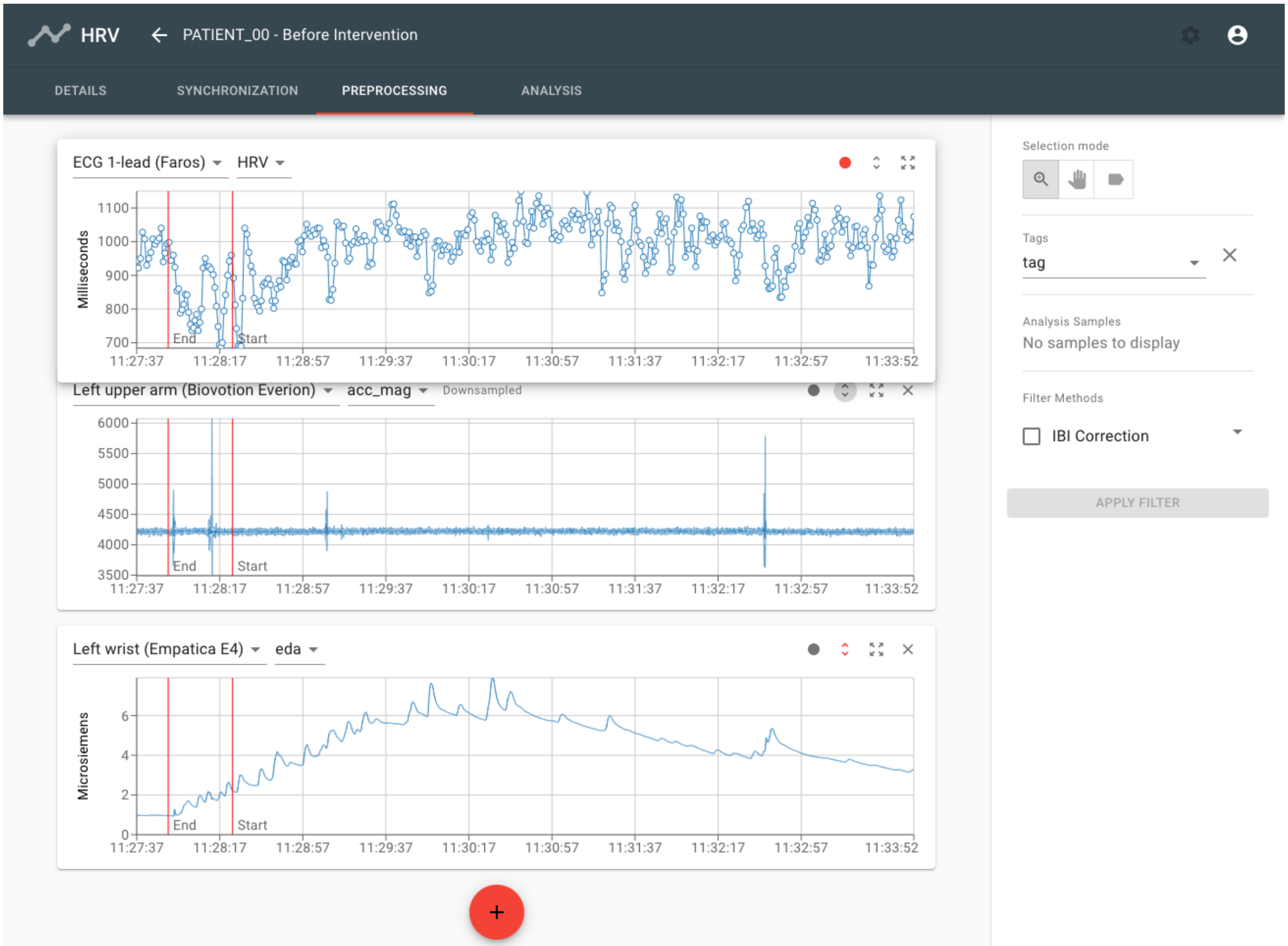


Fig. 5: Example of the screen for preprocessing and exploring signals from multiple devices with interbeat intervals from Faros™180, the accelerometer signal of Biovotion Everion and electrodermal activity from Empatica E4. The sidebar offers different modes of interaction, the selection of textual tags (displayed as red vertical lines), a list of marked analysis samples and filter methods.

as defined in `datasets.constants.signal_types` is used to determine if the time series data is saved in the database or as a binary file.

New analysis and filter methods can be added by subclassing the respective base class. Just like a source, the `name()` function should return a distinct name for the method, that will be shown in the User Interface (UI). With `options()` the developer can allow the user to configure parameters that influence the results. This method should return a dictionary of dictionaries, where each dictionary represents an option under its `id`. Per option `type`, `title`, `unit`, and `default` value should be provided. The `type` can be *boolean*, *number*, *range*, or *select*. For *number* and *range* `limits` can be provided to set a minimum and maximum value for front-end validation. *Select* expects a dictionary of available `items` the user can choose from. To execute the method the application will call the `process()` method, after initializing the instance.

IV. Case Study

ALPS has been developed to provide an integrated tool that facilitates the research in digital health. A case study was performed to illustrate the functionalities of the web platform and how the platform can be used to quickly evaluate parameters across multiple subjects, devices and sensors in a study. Two hypotheses were tested: (1) with an increasing mental workload the stress in a person rises and this is reflected as trends in the HRV indices, and (2) the HRV indices derived from an ECG signal are correlated with the ones from a PPG signal. Therefore, subjects in a sitting position were recorded during a cognitive test with three types of workload (low, medium and high) plus a baseline. Some of the most used HRV analysis methods from time, frequency, and non-linear domain were implemented in the platform. Except for the statistical analysis, all steps were accomplished with the web platform.

Two off-the-shelf wearable devices for different anatomical

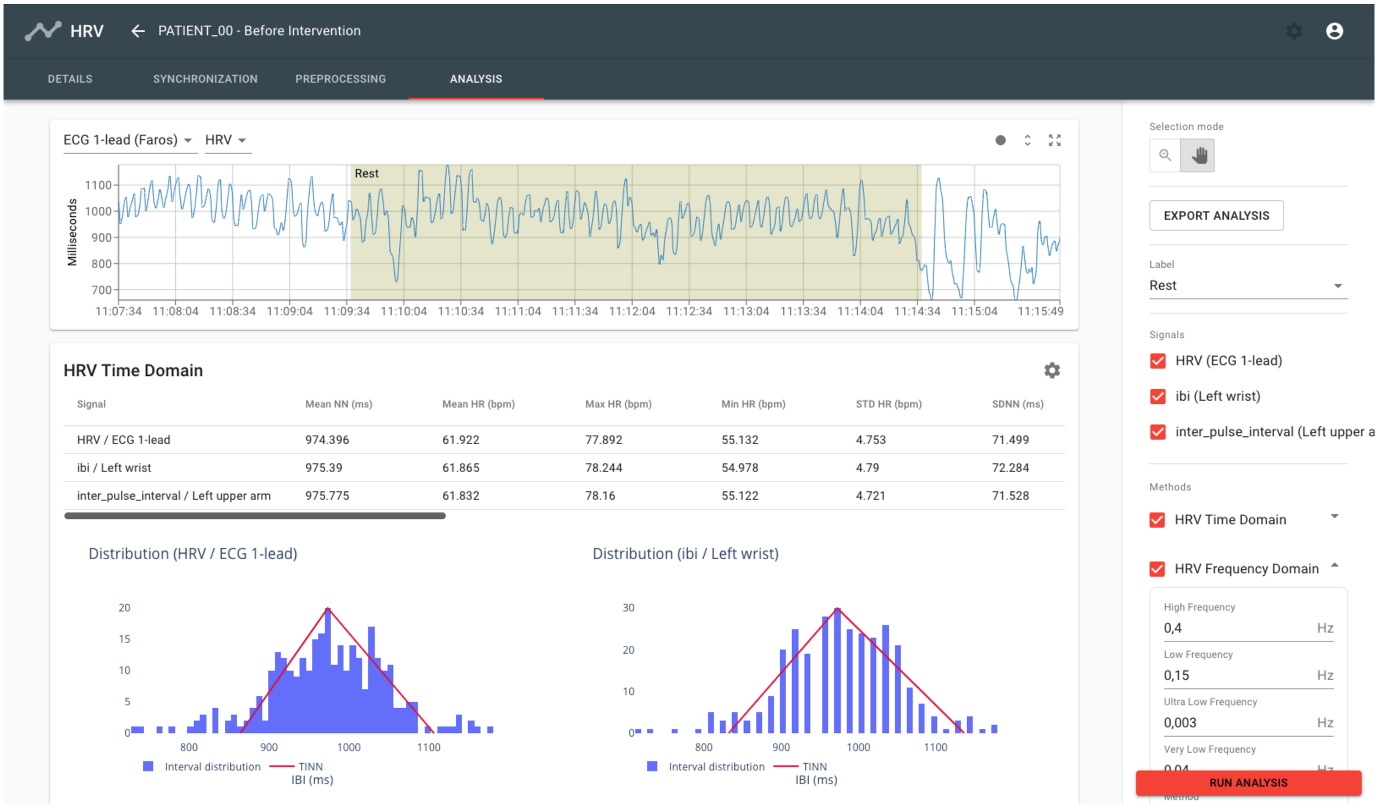


Fig. 6: Screen for signal analysis and comparison with three signals selected simultaneously. Tables containing a flexible amount of parameters can be scrolled horizontally, if necessary, and plots can be displayed. The sidebar offers a selection of label, signals and methods for the analysis.

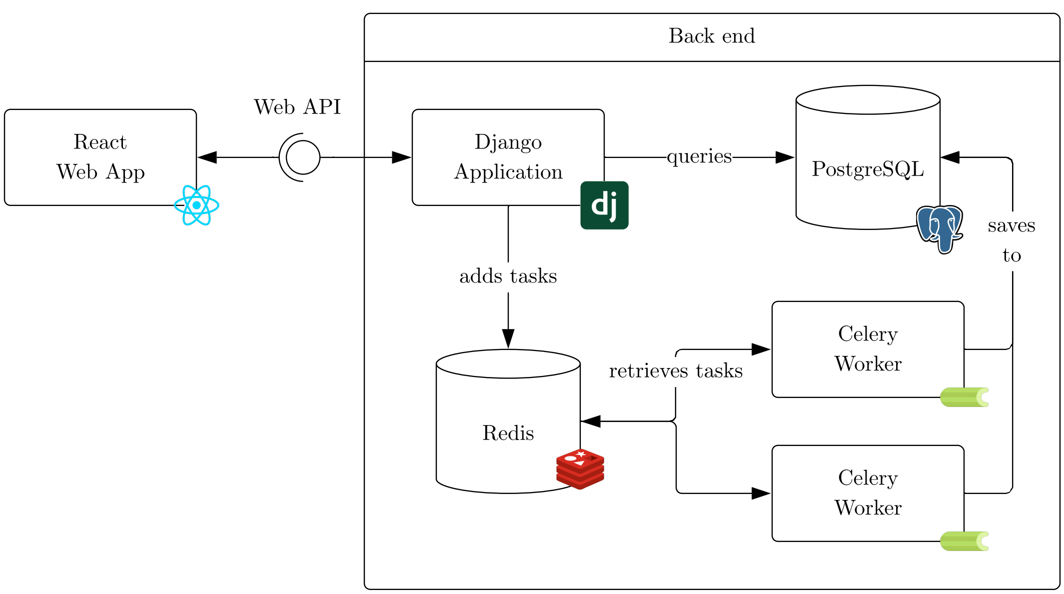


Fig. 7: Overview of system architecture with front and back end.

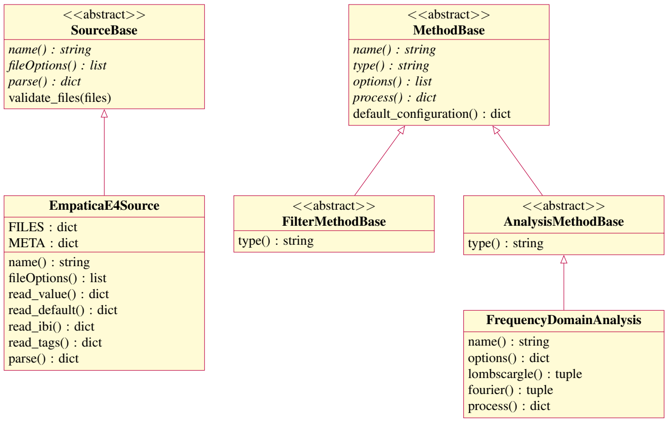
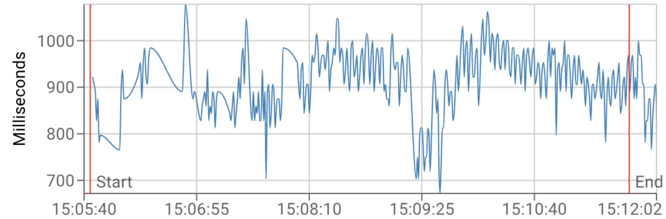
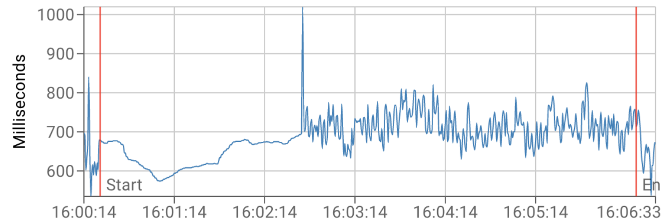


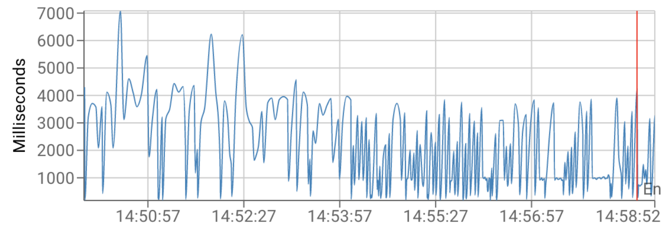
Fig. 8: Class diagram of plugin base classes with examples.



(a) Empatica E4 has a lot of missing beats in the first two minutes



(b) Biovotion Everion was unable to measure IBI correctly and deduced samples in the beginning



(c) Faros™180 recorded no valid IBI values, possibly due to loose electrodes

Fig. 9: Examples of Interbeat Interval (IBI) time series from each device that were excluded from evaluation.

locations, i.e. the wristband Empatica E4 (FW 2.1.0.4911) and the arm band Biovotion Everion (FW 03.06), were compared to the 1-lead chest ECG monitor Faros™180 from Bittium (FW 3.5.1). All devices contain a 3-axis accelerometer, which is used to synchronize them. Empatica as well as Biovotion use CSV files with their own data formatting. Faros data is stored in the European Data Format (EDF).

After uploading the data to ALPS, each session was synchronized with the event-based algorithm. Since the objective of the experiment was to compare the HRV indices obtained from off-the-shelf ECG and PPG based devices, the IBI signals were not filtered. The quality of the IBI signals was manually inspected with the help of the ECG, PPG and accelerometer signals, and for Everion, a quality estimation signal was factored in. With the help of manual inspection some of the recordings could already be discarded, due to noise caused by movement of the participants or poor contact of the electrode to the skin. Some examples taken from the web platform can be seen in Fig. 9. Five minute excerpts from each workload were labeled for each suitable session and HRV indices were computed. The results of all sessions and workloads were exported as CSV for further statistical analysis. The statistical analysis is out of the scope of the current paper.

V. Conclusion

This paper presented ALPS, a web-based platform for signal analysis of multi-device sensor data in the context of clinical studies. The main objectives were the synchronization of data from multiple devices, extensibility and an user-friendly interface to cater to the needs of researchers with and without programming skills. An event-based synchronization algorithm has been implemented, which is tailored to detect deliberate shakes in the accelerometer signal. Extensibility has been achieved by building a plugin architecture to easily add new sources and new methods to filter and analyze time series data. A graphical user interface was built with `React` to allow fast and effortless exploration and processing of sensor data. The use case demonstrated that the platform provides all tools needed, from data management to data analysis, to empower clinical studies using multiple sensors and wearables, paving the way for the Internet of Health Things (IoHT).

VI. Future Work

The implementation of ALPS is prototypical and it is supposed to be improved in future iterations. This concerns not only the analytical features, but also the performance as well as the data security. To protect the data uploaded to the platform, currently a user has to login and a permission management system makes sure that no other user has access to the uploaded data. However, due to the scope of information privacy and data security, these issues were not further examined and no policies exceeding the token-based authentication and permission management are currently implemented. In addition, the usability of the web platform needs to be tested with researchers from different backgrounds to further adjust the UI to their needs and prioritize new features. For this, a pluralistic walkthrough could be applied, where developer, usability expert, and user discuss together the interface elements and the workflow of the software [48].

Appendix

A. Shake Detection Algorithm

$signal \leftarrow$ Array of normalized samples with timestamps
$window \leftarrow$ Time from start and end in seconds
$threshold \leftarrow$ Minimum peak height

$maxDistance \leftarrow$ Maximal time between peaks in milliseconds
$minLength \leftarrow$ Minimum number of peaks in sequence
$peaks \leftarrow$ Samples of $signal$ above $threshold$
$startTime \leftarrow$ Minimum timestamp in $signal$
$endTime \leftarrow$ Maximum timestamp in $signal$
$startSequences \leftarrow$ Empty list
$endSequences \leftarrow$ Empty list
for every $peak$ in $peaks$ **do**
    $startPeak \leftarrow peakTime < startTime + window$
    $endPeak \leftarrow peakTime > endTime - window$
    if $startPeak$ **then**
        $sequences \leftarrow startSequences$
    else if $endPeak$ **then**
        $sequences \leftarrow endSequences$
    end if
    if $startPeak$ or $endPeak$ **then**
        if $prevPeakTime + maxDistance >= peakTime$ **then**
            $sequences[-1].add[peak]$
        else
            $sequences.add([peak])$
        end if
        $prevPeak \leftarrow peak$
    end if
end for
Remove sequences with length $< minLength$ from $startSequences$ and $endSequences$
$startShake \leftarrow$ sequence with highest weight from $startSequences$
$endShake \leftarrow$ sequence with highest weight from $startSequences$
return $startShake, endShake$

B. Time Shift Calculation

$refSignal \leftarrow$ Array of samples with timestamps
$syncSignal \leftarrow$ Array of samples with timestamps
$maxFrequency \leftarrow maxFrequency(refSignal, syncSignal)$
$refSignal \leftarrow$ resample $refSignal$ to $maxFrequency$
$syncSignal \leftarrow$ resample $syncSignal$ to $maxFrequency$
$refTs \leftarrow shakeDetection(refSignal)$
$syncTs \leftarrow shakeDetection(syncSignal)$
for each $event$ in $[start, end]$ **do**
    $refShake \leftarrow$ Samples within $refTs[event]$ from $refSignal$
    $syncShake \leftarrow$ Samples within $syncTs[event]$ from $syncSignal$
    $crossCorrelations \leftarrow correlate(refShake, syncShake)$
    $shiftInSamples \leftarrow indexOfMax(crossCorrelations) - length(syncShake) - 1$
    $startIndex \leftarrow refSignal.indexOf(min(refTs[event])) + shiftInSamples$
    $maxCorrTimestamp \leftarrow refSignal[startIndex].timestamp$
    $timeShifts[event] \leftarrow maxCorrTimestamp - min(syncTs[event])$
end for
return $timeShifts$

Acknowledgment

The authors would like to thank Suparno Datta, M.Sc. from the Hasso Plattner Insititute for Digital Health as well as Matteo Danieletto, Ph.D. from the Hasso Plattner Institute for Digital Health at Mount Sinai for their feedback and assistance in developing the platform and during the HRV analysis.

References

[1] International Data Corporation, *IDC Reports Strong Growth in the Worldwide Wearables Market, Led by Holiday Shipments of Smartwatches, Wrist Bands, and Ear-Worn Devices*, Mar. 2019. [Online]. Available: https : / / www . idc . com / getdoc . jsp ? containerId = prUS44901819 (visited on 11/12/2019).

[2] World Health Organization, *Integrated chronic disease prevention and control*. [Online]. Available: https://www.who.int/chp/about/integrated_cd/en/ (visited on 02/12/2019).

[3] G. A. Roth, M. D. Huffman, A. E. Moran, *et al.*, "Global and regional patterns in cardiovascular mortality from 1990 to 2013," *Circulation*, vol. 132, no. 17, pp. 1667–1678, 2015.

[4] International Diabetes Federation, "The IDF consensus worldwide definition of the metabolic syndrome," *IDF Communications*, pp. 1–23, 2006.

[5] B. M. L. Miller and L. Brennan, "Measuring and reporting attrition from obesity treatment programs: A call to action!" *Obesity Research and Clinical Practice*, vol. 9, no. 3, pp. 187–202, 2015.

[6] E. Burgess, P. Hassmén, and K. L. Pumpa, "Determinants of adherence to lifestyle intervention in adults with obesity: a systematic review," *Clinical Obesity*, vol. 7, no. 3, pp. 123–135, 2017.

[7] M. I. Stuckey, M. P. Tulppo, A. M. Kiviniemi, and R. J. Petrella, "Heart rate variability and the metabolic syndrome: a systematic review of the literature," *Diabetes/Metabolism Research and Reviews*, vol. 30, no. 8, pp. 784–793, Nov. 2014.

[8] E. Gil, M. Orini, R. Bailón, J. M. Vergara, L. Mainardi, and P. Laguna, "Photoplethysmography pulse rate variability as a surrogate measurement of heart rate variability during non-stationary conditions," *Physiological Measurement*, vol. 31, no. 9, pp. 1271–1290, 2010.

[9] S. Kumar, W. Nilsen, M. Pavel, and M. Srivastava, "Mobile Health: Revolutionizing Healthcare Through Transdisciplinary Research," *Computer*, vol. 46, no. 1, pp. 28–35, Nov. 2012.

[10] L. E. Dunne, "Wearable technology," *Encyclopædia Britannica*, Mar. 2016. [Online]. Available: https://www.britannica.com/topic/Wearable-Technology-2008498.

[11] S. Park, K. Chung, and S. Jayaraman, "Wearables: Fundamentals, advancements, and a roadmap for the future," in *Wearable sensors*, Elsevier, 2014, pp. 1–23.

[12] IDC, *Wearables - Absatz weltweit bis 2018*, Mar. 2019. [Online]. Available: https://de.statista.com/statistik/daten/studie/515723/umfrage/absatz-von-wearables-weltweit/ (visited on 12/10/2019).

[13] A. Kamišalić, I. Fister Jr., M. Turkanović, and S. Karakatič, "Sensors and Functionalities of Non-Invasive Wrist-Wearable Devices: A Review," *Sensors*, vol. 18, no. 6, Jun. 2018.

[14] M. Iosa, P. Picerno, S. Paolucci, and G. Morone, "Wearable inertial sensors for human movement analysis," *Expert Review of Medical Devices*, vol. 13, no. 7, pp. 641–659, Jul. 2016, ISSN: 1743-4440.

[15] P. Sarcevic, Z. Kincses, and S. Pletl, "Online human movement classification using wrist-worn wireless sensors," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 1, pp. 89–106, Jan. 2019.

[16] X. Wang, Z. Liu, and T. Zhang, "Flexible Sensing Electronics for Wearable/Attachable Health Monitoring," *Small*, vol. 13, no. 25, p. 1 602 790, Jul. 2017.

[17] *BME680 - Datasheet*, Rev. 1.3, Bosch SensorTec, Jul. 2019.

[18] M. Elgendi, R. Fletcher, Y. Liang, *et al.*, "The use of photoplethysmography for assessing hypertension," *npj Digital Medicine*, vol. 2, no. 60, pp. 1–11, Jun. 2019.

[19] M. Boyer and B. A. Koplan, "Atrial Flutter," *Circulation*, Nov. 2005.

[20] S. Bagha and L. Shaw, "A real time analysis of PPG signal for measurement of SpO2 and pulse rate," *International journal of computer applications*, vol. 36, no. 11, pp. 45–50, 2011.

[21] D. M. Mirvis and A. L. Goldberger, "Electrocardiography," *Heart disease*, vol. 1, pp. 82–128, 2001.

[22] V. Montesinos Canovas, F. I. T. Dell'Agnola, A. Arza Valdes, A. Aminifar, and D. Atienza Alonso, "Multi-Modal Acute Stress Recognition Using Off-the-Shelf Wearable Devices," *Infoscience*, Apr. 2019.

[23] D. R. Seshadri, R. T. Li, J. E. Voos, *et al.*, "Wearable sensors for monitoring the physiological and biochemical profile of the athlete," *npj Digital Medicine*, vol. 2, no. 72, pp. 1–16, Jul. 2019, ISSN: 2398-6352. DOI: 10.1038/s41746-019-0150-9.

[24] J. Laird, "Clock Synchronization Terminology," *InterOperability Laboratory*, Jun. 2012. [Online]. Available: https://www.iol.unh.edu/knowledge/clock-synchronization-terminology.

[25] L. Filipe, F. Fdez-Riverola, N. Costa, and A. Pereira, "Wireless body area networks for healthcare applications: Protocol stack review," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 213 705, 2015.

[26] D. Bannach, O. Amft, and P. Lukowicz, "Automatic Event-Based Synchronization of Multimodal Data Streams from Wearable and Ambient Sensors," *ResearchGate*, vol. 5741, pp. 135–148, Sep. 2009.

[27] *Ahrv - hrv sfotware*, Nevrokard. [Online]. Available: http://www.nevrokard.eu/maini/hrv.html (visited on 12/10/2019).

[28] T. Kaufmann, S. Sütterlin, S. M. Schulz, and C. Vögele, "ARTiiFACT: a tool for heart rate artifact processing and heart rate variability analysis," *Behavior Research Methods*, vol. 43, no. 4, pp. 1161–1170, Dec. 2011.

[29] L. Rodríguez-Liñares, M. J. Lado, X. Vila, A. J. Méndez, and P. Cuesta, "gHRV: Heart rate variability analysis made easy," *Computer methods and programs in biomedicine*, vol. 116, no. 1, pp. 26–38, 2014.

[30] J. T. Ramshur, "Design, evaluation, and application of heart rate variability analysis software (HRVAS)," Master's thesis, University of Memphis Memphis, TN, 2010.

[31] P. Perakakis, M. Joffily, M. Taylor, P. Guerra, and J. Vila, "KARDIA: A Matlab software for the analysis of cardiac interbeat intervals," *Computer Methods and Programs in Biomedicine*, vol. 98, no. 1, pp. 83–89, 2010, ISSN: 01692607. DOI: 10.1016/j.cmpb.2009.10.002. [Online]. Available: http://dx.doi.org/10.1016/j.cmpb.2009.10.002.

[32] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen, "Kubios HRV - Heart Rate Variability Analysis Software," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 210–220, 2014.

[33] *Using LabVIEW for Heart Rate Variability Analysis*, National Instruments, Jun. 2019. [Online]. Available: http://www.ni.com/product-documentation/30832/en/.

[34] C. A. García Martínez, A. Otero Quintana, X. A. Vila, *et al.*, *Heart Rate Variability Analysis with the R package RHRV*. 2017.

[35] P. Gomes, P. Margaritoff, and H. Plácido da Silva, "pyHRV: Development and Evaluation of an Open-Source Python Toolbox for Heart Rate Variability (HRV)," in *International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*, Jun. 2019.

[36] A. N. Vest, G. D. Poian, Q. Li, *et al.*, "An open source benchmarked toolbox for cardiovascular waveform and interval analysis," *Physiological Measurement*, vol. 39, no. 10, p. 105 004, Oct. 2018.

[37] *Django Documentation*. [Online]. Available: https://docs.djangoproject.com/en/2.2/ (visited on 11/10/2019).

[38] *PostgreSQL: Documentation*. [Online]. Available: https://www.postgresql.org/docs/11/index.html (visited on 11/10/2019).

[39] KDnuggets, *Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis*, 2019.

[Online]. Available: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html (visited on 11/10/2019).

[40] *Pandas: Powerful python data analysis toolkit*. [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/ (visited on 11/13/2019).

[41] *Celery - Distributed Task Queue — Celery 4.3.0 documentation*. [Online]. Available: https://docs.celeryproject.org/en/latest (visited on 11/10/2019).

[42] *Redis Documentation*. [Online]. Available: https://redis.io/documentation (visited on 11/10/2019).

[43] *React Documentation*. [Online]. Available: https://reactjs.org/docs/getting-started.html (visited on 11/10/2019).

[44] *React Router Web Documentation*. [Online]. Available: https://reacttraining.com/react-router/web/guides/quick-start (visited on 11/10/2019).

[45] *Redux Documentation*. [Online]. Available: https://redux.js.org/introduction/getting-started (visited on 11/10/2019).

[46] *Material-UI Documentation*. [Online]. Available: https://material-ui.com/ (visited on 11/10/2019).

[47] *Docker Documentation*. [Online]. Available: https://docs.docker.com/ (visited on 11/10/2019).

[48] J. Nielsen, R. L. Mack, *et al.*, *Usability inspection methods*. Wiley New York, 1994, vol. 1.