Open-Source Data Collection for Activity Studies at Scale

Alexander Hoelzemann, Jana Sabrina Pithan, and Kristof Van Laerhoven

Abstract Activity studies range from detecting key indicators such as steps, active minutes, or sedentary bouts, to the recognition of physical activities such as specific fitness exercises. Such types of activity recognition rely on large amounts of data from multiple persons, especially with deep learning. However, current benchmark datasets rarely have more than a dozen participants. Once wearable devices are phased out, closed algorithms that operate on the sensor data are hard to reproduce and devices supply raw data. We present an open-source and cost-effective framework that is able to capture daily activities and routines, and which uses publicly available algorithms, while avoiding any device-specific implementations. In a feasibility study, we were able to test our system in production mode. For this purpose, we distributed the Bangle.js smartwatch as well as our app to 12 study participants, who started the watches at a time of individual choice every day. The collected data was then transferred to the server at the end of each day.

1 Introduction

Many types of studies focus on capturing activity data from human study participants. We can distinguish these types of studies based on the measurement devices and sensors used, the carrying position of the sensors and the domain of the data.

Alexander Hoelzemann
University of Siegen, Ubiquitous Computing, Hoelderlinstr. 3 - 57076 Siegen - Germany,
e-mail: alexander.hoelzemann@uni-siegen.de

Jana Sabrina Pithan
University of Vechta, Institute of Sports Science, Driverstr. 22 - 49377 Vechta - Germany,
e-mail: jana.pithan@uni-vechta.de

Kristof Van Laerhoven
University of Siegen, Ubiquitous Computing, Hoelderlinstr. 3 - 57076 Siegen - Germany,
e-mail: kvl@eti.uni-siegen.de

The types of devices used go hand in hand with the sensor technology used. For example, sensors worn on the wrist offer the possibility of recording the heart rate via PPG sensors, the skin temperature with a thermometer and the movements with an accelerometer, gyroscope and magnetometer. Studies in which smartphones are mainly used to record the data do not usually offer this supplementary sensor technology. Since the devices are not worn directly on the skin, the data is often limited to basic IMU-Sensors. In contrast, the carrying position of the sensors goes hand in hand with the specific domain of the recorded data. Sensor technology used for medical datasets are often worn on different body positions than sensor technology used for the purpose of activity recognition. As previous studies have shown, for many activities, it is often sufficient to wear the sensors only at key positions such as the wrist [23], [29]. In the medical environment, however, more complex sensors and different wearing positions are often required [20], [1].

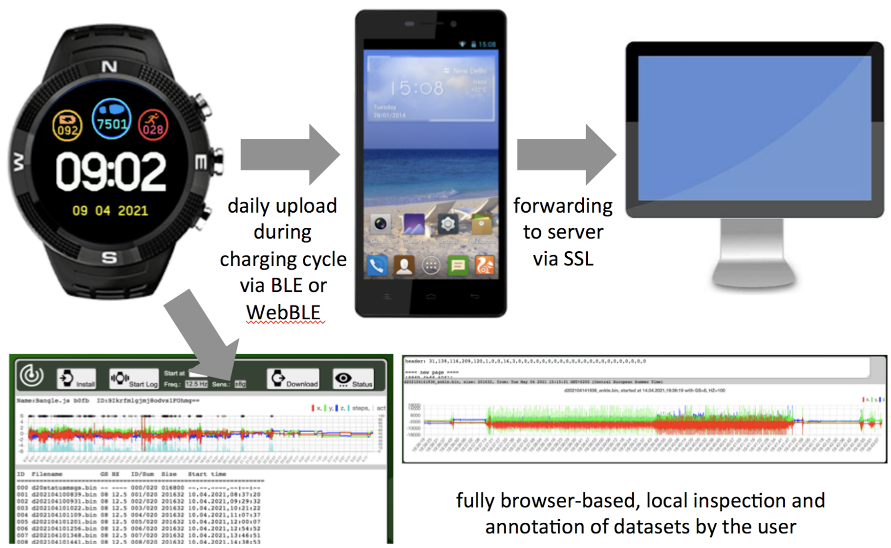

Fig. 1 Our system relies on an open-source smartwatch [37] with custom firmware, smartphone apps, and a server-side database to collect all data centrally. For participants without smartphone or in studies where users need to inspect their data or manually forward their data, a web-based suite (bottom) retrieves the data through WebBLE. The raw sensor data is frequently streamed from the smartwatch either to a nearby computer via web-based control panel, or via the user's smartphone to a dedicated server.

Empirical studies for which activity plays a crucial role use indicators such as steps taken, sedentary periods, activity counts, or detected physical exercises, which often originate from closed-source algorithms. This tends to lock studies to particular devices and makes the use of other devices or comparisons difficult. In contrast, sensors such as accelerometers or inertial measurement units are already widely integrated in many wearables, and tend to produce calibrated sensor data in units such as mg. Restricting studies to particular commercial wearables that also record raw inertial data has the effect that large-scale studies are only possible if the project has a high budget that allows the purchase of commercial hardware and software. In this paper we present the ActiVatE_prevention system, which is based exclusively on open-source components, logs raw inertial data, and also offers subjects a similar wearing comfort as commercially manufactured products. We argue that it therefore lends itself well to the capturing of multiple users simultaneously for activity studies, while being an open source, replicable, and low-cost approach.

2 Related Work

Plenty of studies that log wearable inertial data to capture the activity of a user have been proposed throughout the past two decades. In human activity recognition research, for instance, recently published survey papers, such as Chen et al., table 1, [7] or Demrozi et al., table 6, [8], show that the same datasets, e.g. WISDM [3], OPPORTUNITY [31], PAMAP2 [30] or DSADS [2], are used for many machine learning papers published in recent years. These datasets are limited regarding their nature, in respect to scope, quality, continuity and reliability. We extended these lists of compared datasets by adding the SHL [13] and the RealWorld (HAR) [33] dataset. Taken the numbers into account that were given by the publications, we calculate a median of about 13 activities and 12 subjects for activities of daily living. Datasets that have a significantly higher number of activities or subjects are often recorded using smartphones. However, a smartphone does not provide the same level of control to record data as our open source operating system, since the underlying operating system is in control of when exactly an instruction is executed by the CPU. Most of the published datasets were recorded using more than one sensor attached to the body. These sensors were prototypes developed in a lab and therefore not optimized to being inconspicuous and comfortable to wear. Furthermore, study participants were always conscious of being recorded, thus (unintentionally) changing activity patterns which leads to the recorded data being biased [36].

However, in order to develop machine learning algorithms that are reliable and robust with everyday situations and data recorded in the wild, large and standardized datasets are needed. Several research projects and publications have highlighted the challenges and needs for robust and systematic collection of activity. The ActiServ project [4] presents a smartphone-based software architecture to infer activities from local sensor data and specifically designed for everyday use, enabling flexible placement of the device and requiring minimal effort from the user. The AWARE [12] framework is developed as an open source framework which uses smartphone integrated sensors to record human activity data. It is available for Android and iOS and comes with a server application that uses the rapid preprocessing pipeline for machine learning [35] to preprocess incoming data streams. The SPHERE Sensor Platform [34] is a multi-sensor fusion approach, which is deployed for healthcare supervision in residential housing. IMU-Wristbands as well as environment sensors and 3D Kinect cameras are used to supervise behaviors such as sleep, physical activity, eating, domestic chores and social contact. The system has been deployed in about 100 households. On a smaller scale, Mairittha et al. [28] present a mobile app for crowdsourcing labeled activity data from smartphone integrated sensors. They recorded 1,749 labeled subsets of activity data. This application is neither available on the app stores nor on known repository platforms. E-care@home [22] is an open source collection of software modules for data collection, labeling and reasoning tasks, such as activity recognition or person counting in a smart home environment and is meant to be used for large-scale data acquisition in a home environment. The solution is partly open-source and available to download from their GitLab repository [25]. Several sensor nodes are placed in a smart home and used to record data.

Wear OS from Google [15] was released under this name in March 2018 and also offers developers the possibility access raw sensor data. Methods for activity recognition of different sports can also be integrated via the Google FIT API [14]. For using this service, however, one is tied to Google and their contract terms.

In current publications of data collection frameworks and algorithms, the main focus has been on video and image data based activity recognition, [6], [26], [16] or [17]. Similar open-source systems do not yet exist for IMU data, since most frameworks are either smartphone based or the needed wearables are lab made prototypes that cannot be purchased easily online.

IMUTube [24], an algorithm that is capable of generating artificial IMU data from humans in videos. Such large and publicly available datasets do not yet exist in the area of IMU-based activity recognition.

Extensive datasets with sensor based human activity data have been difficult to record due to the need to use specific hardware with sensors that are often difficult to start, uncomfortable to wear, or data sharing is limited for inexperienced volunteers. Therefore, researchers started to use data augmentation techniques on inertial sensor data to create synthetic data [11] or [10]. These techniques increase the size of the dataset, but are limited if we want to increase quality and variability [18]. With the pervasiveness of inertial sensors embedded in commercial smartwatches, it has become easy to deploy applications that use inertial data locally, but longer recordings of these data in a common format (for instance, using particular sensitivities and sampling rates) remain difficult.

3 Our Proposed Approach

The design of our open source system is shown in Figures 2 and 3. The operating system is installed once on the Bangle.js via Web-BLE and the apps are downloadable via the Apple AppStore and the Google Play Store. The app forwards the data from the smartwatch to the central server. The user interface of the app is kept simple, the users can only select their daily activity goals and retrieve their daily activity statistics.

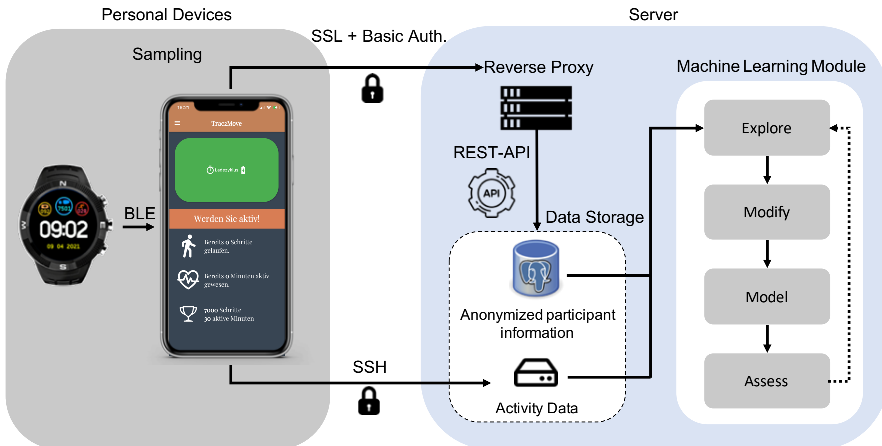

Fig. 2 Open source client-server architecture for recording human activity data. The data is recorded by the Bangle.js smartwatch and is sent to the server daily with our app. Anonymized participant information is sent to the server via a reverse proxy that implements SSL + Basic Authentication This reverse proxy communicates via a REST-Api with the Postgres SQL database. The system is designed in accordance with the SEMMA data process model [32]. (1) Sampling, (2) Explore (3) Modify (4) Model (5) Asses. The model itself can be seen as a cycle.

The sequence diagram (Figure 3) depicts the communication in between the architecture elements. We recorded the execution time for every communication step, which is added to the diagram. On average, it takes 185 seconds to send one file (approx. 200KB and 1 hour of data) from the watch via BLE to the smart device. After ∅ 45 minutes, the complete daily data is sent from the smartwatch to the server.

Smartwatch. To date, there are few open-source smartwatch designs that allow algorithms for detecting activities, from basic ones such as steps, sedentary bouts, and active minutes, to recognition of particular exercise repetitions, to be transparently implemented on a device with integrated inertial sensors. We used the Bangle.js [37] as an affordable (around 50 $) low-power system that is equipped with a Nordic 64MHz nRF52832 ARM Cortex-M4 processor, inertial sensors, a PPG sensor, sufficient internal memory, and an internal BLE module. Our firmware on this open-source platform is capable of storing the sensors' raw data over a full day, and integrate recognition algorithms – currently for steps, active minutes, and exercise intensities – locally on the watch. Users are expected to start the data upload process once a day, either through the web-based platform, or automatically through their smartphone or tablet app.

Since the logging of activity data requires sampling rates from 10Hz up to as high as 100Hz, depending on the activity, the recording of raw inertial data is rarely implemented in a way where local recordings are routinely synchronized and uploaded to a server. The local storage for a day's worth of inertial data and the energy footprint for sending this data tends to be substantial [5]. Instead, the early pre-processing of inertial data in the aforementioned detected features (steps, active

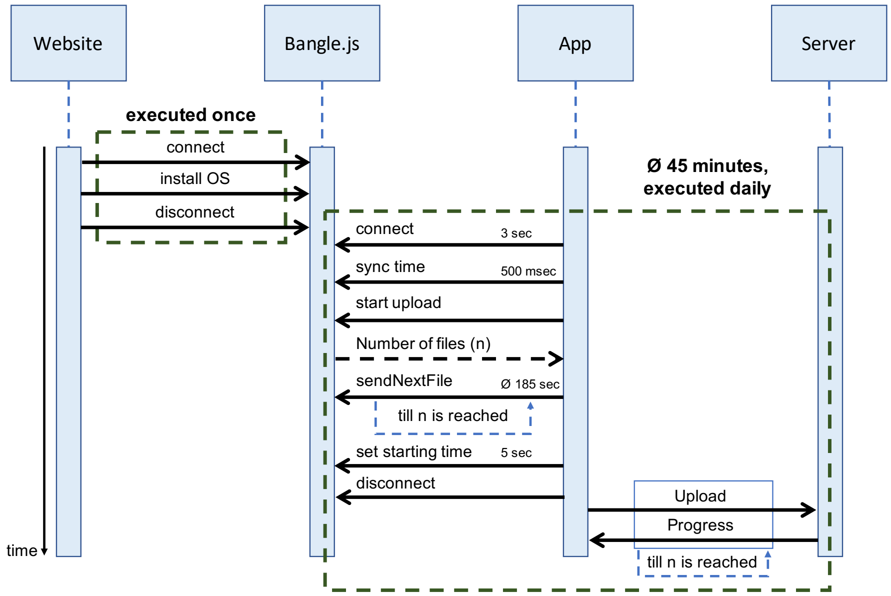

Fig. 3 The operating system is installed from our webtool via Web-BLE on the Bangle.js. This needs to be executed once. The communication between Bangle.js and the app occurs on a daily basis. The procedure needs ∅ 45 minutes for a full day of recording (14 hours of active time) and ∅ 185 seconds for sending one file from the watch to the smartphone. The upload to the server is executed when all files are transferred to the smartphone.

mintes, etc) takes place on the wearable devices and usually solely these aggregated values are stored.

Detected activity-related concepts such as Active Minutes [19] have been deployed locally on the Bangle.js smartwatch and are uploaded together with the raw sensor data to the server through the smartphone app (or via the browser-based tool suite) on a daily basis. We designed to fully use the watch's 4Mb flash memory to losslessly compress 16 bit, 12.5 Hz inertial data at +/-8g, along with other data such as the skin temperature and heart rate.

iOS and Android App. The Activate client is implemented using Flutter. Therefore, we are able to design and implement clients for the two major operating systems, iOS and Android, at once. However, minor code changes are necessary to solve operating system specific issues, especially with regards to the BLE connection.

The interface consists of three main views and is displayed in German language. It was designed to encourage diabetes patients to perform more physical activities in their daily lives. Beyond the recording of raw inertial data, it is planned for the near future to expand this open source app to be able to annotate and detect an arbitrary number of activities as well.

When the app starts, the participant is taken to the home screen, (1) in Figure 4. Here, the user interface visualizes an overview of the day's accumulated number of steps taken and active minutes. When pressing the green button, the study participant

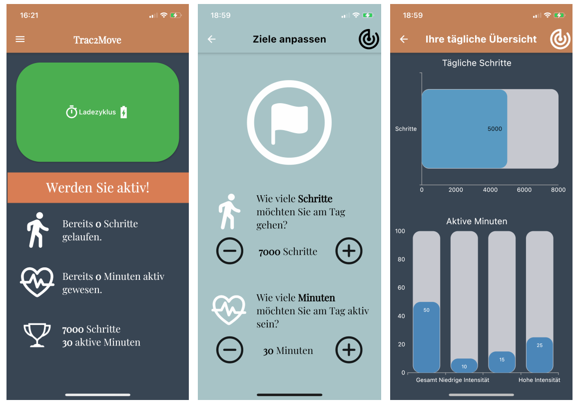

Fig. 4 The smartphone's user interface: (1) Home Screen, (2) Setting daily activity goals, e.g. Daily Steps (Tägliche Schritte) and daily Active Minutes (Aktive Minuten), (3) Graphical overview of daily activities: Daily Steps (Tägliche Schritte), Active Minutes (Aktive Minuten), devided into three intensities - low, moderate and vigorous (niedrige, mittlere, hohe Intensität).

saves the data on the server and sets the starting time for the following measurement (typically the next day). During the first start of the app, an anonymized user account is created and saved in a Postgres SQL database.

On the second screen in Figure 4, the user can set their personal goals for the day within its limits. Screen (3) in Figure 4 gives a graphical overview of the daily metrics and shows, beside the total number of steps and active minutes, also the active minutes sorted by their intensities.

Server. The server communicates with the client via two channels, Figure 2. Private information about the study participants, such as gender or age, and the confirmation of the consent form are sent via SSL and Basic Authentication to a reverse proxy which then sends the information to the database via localhost. The information is stored in an anonymous form. The recorded activity data, as well as daily steps and active minutes, are sent via SSH to the server and stored in binary files with delta compression. The activity data can then be processed and modelled by machine learning algorithms.

Browser-based data analysis. The smartphone or tablet app and server software described above can be complemented with a local analysis and annotation tool that can be used by the study participants. This requires users to simply visit a website that can connect to the watch through WebBLE and download the watch's data locally on the computer for further inspection or manual upload to our central study server, through users' own computers without the need to install software.

4 Performance Analysis

Since our software is distributed between apps that are available as a web-based software suit or downloadable in Apple's App Store and Android's Play Store, the deployment of our system is straightforward. We gave the Bangle.js smartwatches to 12 geographically-distributed study participants and recorded compliance, comfort rating, and reliability performance measures for our presented approach to illustrate the feasibility of our approach, and report our findings below.

We analyzed recordings from participants over a window of five days and decided to let them choose how many hours they recorded by letting them start and stop the smartwatch with the app at a time of their choice. This is important because of the age group and the profession of the subject, which entails certain active and inactive, as well as sleep and wake cycles [9].

During the feasibility study we focused on detecting basic activity concepts such as steps as well as the active minutes divided into the three subclasses, low, moderate and vigorous intensity. The participants wore the smartwatch for an average of 12 hours per day. In total we collected approx. 29 MB (12*202KB*12 participants) of raw compressed data. Basic activity classes are already recognized on the watch without machine learning. However, since the Bangle.js has Tensorflow-Lite already implemented on the hardware, there is an opportunity to deploy a pre-trained neural network or machine learning classifier on the watch in the future. A recent article [27] demonstrates how to implement this for gesture recognition.

We can demonstrate by means of our experiment that the system we have designed can be used for data recordings in the wild without the subject being biased by the technology worn, since the smartwatch is a commercially designed product and looks and feels like a normal watch. Due to the 4 Mb memory limitation of Bangle.js, we limit the inertial measurements to a 12.5 Hz sampling rate so that a full 24-hour day can still be recorded in one cycle. We consider this sampling rate acceptable, since activity detection is still possible at such a low sampling rate. Furthermore, the signal can be interpolated as part of the machine learning preprocessing or the sampling rate can be increased at the cost of shorter recordings (a 100 Hz recorded data set corresponds to about 3 hours).

Occasionally, data uploads are hampered because of problems with a reliable Internet connection and the Bluetooth connection between Bangle.js and app in particular. The communication flow as depicted in figure 3 has therefore been developed for stability and has built-in recovery mechanisms that guarantee that individual files are uploaded reliably. The current version is therefore characterized by a high reliability and accessibility, but also relatively long upload times (around 45 minutes on average for a full day's data set). However, this seems acceptable, as the download process has been integrated with charging the smartphone and Bangle.js smartwatch in the nightly "charging cycle".

Bangle.js Wearing Comfort. In addition to the feasibility study of our open source architecture's ability to accommodate data over multiple users and in a distributed manner, we decided to investigate Bangle.js in terms of its comfort of use. We consider this to be important, since the success of a study is directly dependent on the acceptance of a device. We use the Comfort Rating Scale (CRS), a questionnaire-based method proposed by Knight et al. [21], as a well-known and state-of-the-art method to evaluate the wearing comfort of wearable devices in particular.

The Bangle.js smartwatch was rated (as Figure 5 shows) overall as comfortable to wear without restricting its users. However, users can feel the device on their wrist due to its larger size (5 x 5 x 1.7 cm case) and weight. The device is heavier and

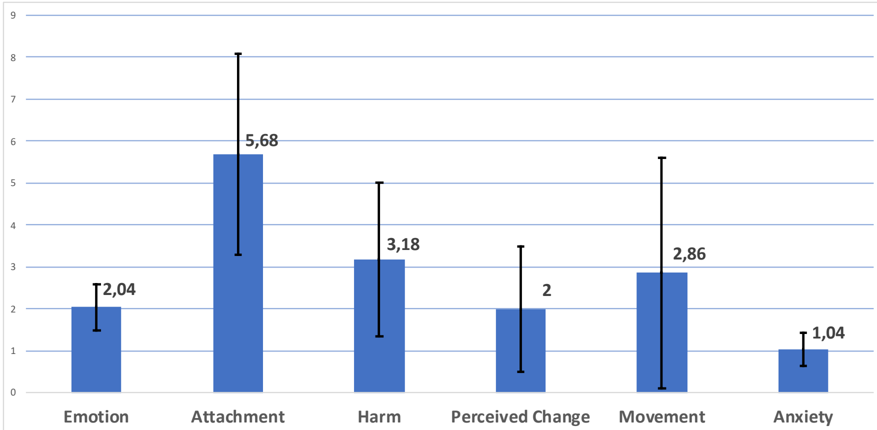

Fig. 5 CRS result means and standard deviation. Emotion: 2.04, 0.56; Attachment: 5.68, 2.39; Harm: 3.18, 1.84; Perceived Change: 2, 1.49; Movement: 2.86, 2.75; Anxiety: 1.04, 0.39.

more bulky than most commercial wrist-worn products, which may lead to slightly negative wearing comfort and perhaps more difficult acceptance in larger future studies. We consider this an acceptable trade-off, as only one person in the study reported that the watch had a strong negative emotional impact on them, and that they would have liked to take it off.

5 Conclusions

The use of low-cost and open-source systems is essential for future machine learning applications. Only through the development and use of such systems it will be possible to generate the required amount of data to train a neural network to be used in a real-world context in a generalized way. Many publications show new and exciting methods in dealing with human activity data, however, these methods are always evaluated on the same datasets mentioned before. This creates a bias in our scientific domain, which can only be eliminated by publicly available, understandable and reusable implementations for data collection.

The already available open-source platforms and systems presented in chapter 2 are either smartphone-based or smart-home based solutions. Smartwatch-based solutions are mostly prototypes, which are not meant to be distributed in scale and not open-source. Due to its open-source architecture, the use of the Bangle.js wristwatch combines the advantages of a product while having an open architecture that is fully documented. Our custom operating system as well as the client-server architecture can serve as a starting point that can later be modified or further developed accordingly. Due to the low purchase price, the device can be used in projects with a smaller budget or in need of a larger group of users. In contrast to a self-developed prototype, where the wearing comfort is often not the main interest, the Bangle.js was confirmed to offers a high acceptance by study participants in our study using the comfort rating scale (CRS). We argue that this aspect also contributes to the long-term success of a scientific study and the scope, quality, continuity and reliability of the produced dataset.

Commercial products tend to not open the algorithms used, and do not give researchers the same insights in recorded data as a fully open-source implementation does. Therefore, we made the source code of the smartphone app as well as the smartwatch operating system available for download and inspection under the MIT licence, to encourage other researchers to replicate and improve on our approach:
https://github.com/ahoelzemann/activateFlutter,
https://github.com/kristofvl/BangleApps/blob/master/apps/activate/app.js

Funding

This publication is part of the project ActiVAtE_prevention which is funded by the Ministry for Science and Culture of the federal state of Lower Saxony in Germany (VW-ZN3426).

References

[1] AlShorman, O., B. AlShorman, M. Alkhassaweneh, and F. Alkahtani, A review of internet of medical things (iomt)–based remote health monitoring through wearable sensors: A case study for diabetic patients, *Indonesian Journal of Electrical Engineering and Computer Science*, *20*(1), 414–422, 2020.

[2] Altun, K., B. Barshan, and O. Tunçel, Comparative study on classifying human activities with miniature inertial and magnetic sensors, *Pattern Recognition*, *43*(10), 3605–3620, 2010.

[3] Anguita, D., A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones., in *Esann*, vol. 3, p. 3, 2013.

[4] Berchtold, M., M. Budde, D. Gordon, H. R. Schmidtke, and M. Beigl, Actiserv: Activity recognition service for mobile phones, in *2010 International Symposium on Wearable Computers (ISWC 2010)*, pp. 1–8, IEEE Computer Society, Los Alamitos, CA, USA, doi:10.1109/ISWC.2010.5665868, 2010.

[5] Berlin, E., M. Zittel, M. Braunlein, and K. V. Laerhoven, Low-power lessons from designing a wearable logger for long-term deployments, IEEE, doi: 10.1109/sas.2015.7133581, 2015.

[6] Caba Heilbron, F., V. Escorcia, B. Ghanem, and J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in *Proceed-*

ings of the ieee conference on computer vision and pattern recognition, pp. 961–970, 2015.

[7] Chen, K., D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities, *ACM Computing Surveys (CSUR)*, *54*(4), 1–40, 2021.

[8] Demrozi, F., G. Pravadelli, A. Bihorac, and P. Rashidi, Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey, *IEEE Access*, 2020.

[9] Espiritu, J. R. D., Aging-related sleep changes, *Clinics in geriatric medicine*, *24*(1), 1–14, 2008.

[10] Esteban, C., S. L. Hyland, and G. Rätsch, Real-valued (medical) time series generation with recurrent conditional gans, *arXiv preprint arXiv:1706.02633*, 2017.

[11] Fawaz, H. I., G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, Data augmentation using synthetic data for time series classification with deep residual networks, *arXiv preprint arXiv:1808.02455*, 2018.

[12] Ferreira, D., V. Kostakos, and A. K. Dey, Aware: mobile context instrumentation framework, *Frontiers in ICT*, *2*, 6, 2015.

[13] Gjoreski, H., M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen, The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices, *IEEE Access*, *6*, 42,592–42,604, 2018.

[14] Google-LLC, Google fit api, google Fit API, .

[15] Google-LLC, Google wear os, google Wear OS, .

[16] Goyal, R., et al., The" something something" video database for learning and evaluating visual common sense, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5842–5850, 2017.

[17] Gtoderici, gtoderici/sports-1m-dataset.

[18] Hoelzemann, A., N. Sorathiya, and K. Van Laerhoven, Data augmentation strategies for human activity data using generative adversarial neural networks, in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 8–13, IEEE, 2021.

[19] Jetté, M., K. Sidney, and G. Blümchen, Metabolic equivalents (METS) in exercise testing, exercise prescription, and evaluation of functional capacity, *Clinical Cardiology*, *13*(8), 555–565, doi:10.1002/clc.4960130809, 1990.

[20] Khan, Y., A. E. Ostfeld, C. M. Lochner, A. Pierre, and A. C. Arias, Monitoring of vital signs with flexible and wearable medical devices, *Advanced Materials*, *28*(22), 4373–4395, 2016.

[21] Knight, J., C. Baber, A. Schwirtz, and H. Bristow, The comfort assessment of wearable computers, in *Sixth International Symposium on Wearable Computers (ISWC 2002),*, IEEE Press, doi:10.1109/iswc.2002.1167220, 2002.

[22] Köckemann, U., M. Alirezaie, J. Renoux, N. Tsiftes, M. U. Ahmed, D. Morberg, M. Lindén, and A. Loutfi, Open-source data collection and data sets for activity recognition in smart homes, *Sensors*, *20*(3), 879, 2020.

[23] Kunze, K., and P. Lukowicz, Sensor placement variations in wearable activity recognition, *IEEE Pervasive Computing*, *13*(4), 32–41, doi: 10.1109/mprv.2014.73, 2014.
[24] Kwon, H., C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Ploetz, Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *4*(3), 1–29, 2020.
[25] Köckemann, U., Projects · uwe köckemann / ecare-pub.
[26] Levine, S., P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, *The International Journal of Robotics Research*, *37*(4-5), 421–436, 2018.
[27] Madsen, A., Running tensorflow lite on nodewatch/bangle.js - nearform, 2020.
[28] Mairittha, N., and S. Inoue, Crowdsourcing system management for activity data with mobile sensors, in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 85–90, IEEE, 2019.
[29] Mannini, A., S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell, Activity recognition using a single accelerometer placed at the wrist or ankle, *Medicine and science in sports and exercise*, *45*(11), 2193, 2013.
[30] Reiss, A., and D. Stricker, Introducing a new benchmarked dataset for activity monitoring, in *2012 16th International Symposium on Wearable Computers*, pp. 108–109, IEEE, 2012.
[31] Roggen, D., et al., Collecting complex activity datasets in highly rich networked sensor environments, in *2010 Seventh international conference on networked sensing systems (INSS)*, pp. 233–240, IEEE, 2010.
[32] Shafique, U., and H. Qaiser, A comparative study of data mining process models (kdd, crisp-dm and semma), *International Journal of Innovation and Scientific Research*, *12*(1), 217–222, 2014.
[33] Sztyler, T., and H. Stuckenschmidt, On-body localization of wearable devices: An investigation of position-aware activity recognition, in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–9, IEEE Computer Society, doi:10.1109/PERCOM.2016.7456521, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7456521, 2016.
[34] Twomey, N., et al., The SPHERE challenge: Activity recognition with multimodal sensor data, *arXiv preprint arXiv:1603.00797*, 2016.
[35] Vega, J., M. Li, K. Aguillera, N. Goel, E. Joshi, K. C. Durica, A. R. Kunta, and C. A. Low, *RAPIDS: Reproducible Analysis Pipeline for Data Streams Collected with Mobile Devices (Preprint)*, doi:10.2196/preprints.23246, 2020.
[36] Vickers, J., A. Reed, R. Decker, B. P. Conrad, M. Olegario-Nebel, and H. K. Vincent, Effect of investigator observation on gait parameters in individuals with and without chronic low back pain, *Gait & posture*, *53*, 35–40, 2017.
[37] Williams, G., The world's first open source hackable smart watch, bangle.js - Hackable Smart Watch.

View publication stats