Smart Devices are Different: Assessing and Mitigating Mobile Sensing Heterogeneities for Activity Recognition

Allan Stisen[‡], Henrik Blunck[‡], Sourav Bhattacharya[*], Thor Siiger Prentow[‡],
Mikkel Baun Kjærgaard[‡], Anind Dey[†], Tobias Sonne[‡], and Mads Møller Jensen[‡]
[‡]Department of Computer Science, Aarhus University, Denmark
[*]Bell Laboratories, Dublin, Ireland
[†]Carnegie Mellon University, USA
{allans, blunck, prentow, mikkelbk, tsonne, mmjensen}@cs.au.dk
sourav.bhattacharya@bell-labs.com, anind@cs.cmu.edu

ABSTRACT

The widespread presence of motion sensors on users' personal mobile devices has spawned a growing research interest in human activity recognition (HAR). However, when deployed at a large-scale, e.g., on multiple devices, the performance of a HAR system is often significantly lower than in reported research results. This is due to variations in training and test device hardware and their operating system characteristics among others. In this paper, we systematically investigate sensor-, device- and workload-specific heterogeneities using 36 smartphones and smartwatches, consisting of 13 different device models from four manufacturers. Furthermore, we conduct experiments with nine users and investigate popular feature representation and classification techniques in HAR research. Our results indicate that on-device sensor and sensor handling heterogeneities impair HAR performances significantly. Moreover, the impairments vary significantly across devices and depends on the type of recognition technique used. We systematically evaluate the effect of mobile sensing heterogeneities on HAR and propose a novel clustering-based mitigation technique suitable for large-scale deployment of HAR, where heterogeneity of devices and their usage scenarios are intrinsic.

Keywords

Mobile Sensing, Activity Recognition

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology; I.5.4 [**Pattern Recognition**]: Signal processing

1. INTRODUCTION

Off-the-shelf modern smartphones readily support an increasingly rich set of embedded sensors such as accelerometer, gyroscope, compass, WiFi, NFC and GPS [30, 45]. This growing ubiquity of sensor rich mobile devices in our everyday lives provides a unique opportunity to unobtrusively capture contextual information from the underlying human behavior in real-time. This growth has also led to an easier development, deployment and a wide proliferation of publicly available mobile sensing applications. Novel mobile sensing applications have also opened up new possibilities for mobile sensing research.

Among the sensors available on mobile consumer device platforms, the accelerometer is one of the earliest and most ubiquitous. The accelerometer has gained immense popularity in HAR research as it allows recognizing a wide variety of human activities, while having a relatively small energy footprint [8]. Accelerometer-based HAR has been deployed in a large number of domains including smart homes [5, 32, 38], health care [31, 37], daily activity tracking [29], fitness tracking [16], fall detection of elderly people [18] and transportation mode detection [9, 24, 42]. Other motion-related sensors, such as compass and gyroscope, are becoming increasingly common-place and often used for assisting and complementing the accelerometer. Motion sensors can be further paired with sensors, e.g., GPS, GSM, WiFi, and barometer, especially for recognizing tasks beyond basic HAR.

Although, a large body of motion sensor-based HAR research exists, real-world performance variations across, e.g., device manufacturers, models, OS types, and CPU load conditions, have been largely overlooked and not been evaluated rigorously yet. When a HAR system is deployed 'in the wild', i.e., across heterogeneous devices and usage situations, recognition performances are often significantly lower than what is suggested in previous research, as noted, e.g., by Amft [2] and Blunck *et al.* [13]. While device placement and orientation and differences in how users perform physical activities have been noted and mitigations for the adverse effects on HAR have been proposed, heterogeneities across devices and their configurations have not been studied rigorously. In this paper, we aim to bridge this gap in HAR research by systematically studying various heterogeneities in motion sensor-based sensing, their impact on HAR, and propose mitigation solutions. Below we elaborate on three major types of heterogeneities, which yield impairments of HAR.

Sensor Biases (SB): To keep the overall cost low, mobile devices are often equipped with low cost sensors, which are often poorly calibrated, inaccurate, and of limited granularity and range, compared to dedicated sensors for HAR, e.g., a dedicated standalone Inertial Measurement Unit (IMU). Furthermore, sensor biases may shift over time through everyday device usage, e.g., accidental device dropping might increase sensor biases.

Sampling Rate Heterogeneity (SRH): As of 2014, on the Android platform alone there are more than 18,000 distinct smartphone models [36]. Often popular smartphones vary in terms of the default and

SenSys'15, November 1–4, 2015, Seoul, South Korea..
© 2015 ACM. ISBN 978-1-4503-3631-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2809695.2809718.

supported sampling frequencies for accelerometer and other sensors. To highlight this in Table 1 we summarize the supported maximum accelerometer sampling frequency across 36 devices, spanning over 13 device models, used in our experiments.

Sampling Rate Instability (SRI): A number of factors, including delays in OS level timestamp attachment to sensor measurements and instantaneous I/O load, affect both the actual as well as the reported sampling rate of sensors on a device. For example, heavy multitasking and I/O load on mobile devices, as exhibited in typical usage scenarios, often lead to unstable sampling rates as the mobile OS frequently fails to attach accurate timestamp information as the measurements arrive. Unpredictability (and inevitability) of such loads makes the multitasking impairments challenging in HAR. Further irregularities in sampling rate may be introduced by modern sensing strategies. For example APIs supporting continuous sensing on the mobile platforms often rely heavily on dynamic duty-cycling to lower the overall power consumption [8].

In this paper, we empirically quantify the above heterogeneities, experimentally evaluate their effects on the HAR performance, and discuss mitigation of these impairments. In the following, we summarize the main contributions of our work:

- We present several sources of intrinsic heterogeneities in mobile sensing, focusing on accelerometer sensors. We conduct an extensive analysis of such heterogeneities using 31 smartphones, 4 smartwatches and 1 tablet, representing 13 different models from 4 manufacturers, running variants of Android and iOS, respectively (see Section 3).
- We systematically study the influence of these heterogeneities on HAR performance by conducting a case study using nine users, each carrying 2 instances of 3 smartphone and 2 smartwatch models. We report the effects of heterogeneities on various feature representation techniques and classifiers popular within HAR research considering a number of cross-validation techniques (see Section 4).
- We propose a novel clustering-based approach as a mitigation technique to improve HAR performance in the presence of heterogeneities (see Section 5).

2. RELATED WORK

Within the field of ubiquitous computing research, sampling rate unpredictability when using motion sensors on smartphones is a known issue: For example, Lane et al. [30] report that for Nokia Symbian and Maemo phones, accelerometers would return samples unpredictably to an application at $25-38$ Hz and Bieber et al. [11] report that such variations can change during normal phone usage, e.g., while making and receiving calls. Using interpolation in pre-processing as a mitigation technique, the authors reported an accuracy of 95% for distinguishing activities such as 'walking', 'running', 'jumping', 'biking' and 'driving' on the phone. However, they did not report on the impact that the impairments, and the proposed mitigations, had on the overall activity recognition performance. Albert et al. [1] show that the accelerometer sampling rate on T-mobile G1 phones running Android varies between $15-25$ Hz. The authors also use interpolation as a potential solution without studying its impact in detail.

Various research investigates motion sensor biases. Among early works, Lötters et al. proposed an online calibration method to mitigate accelerometer drift and offset over a long period of time [33]. The proposed method relies on the fact that under *quasi static* conditions, a measurement vector from a tri-axial accelerometer should equal $1\boldsymbol{g}$. The online approach requires a few minutes to carry out and results in a post-procedure bias of $0.03\boldsymbol{g}$, without requiring explicit orientation information. Gulmammadov also studied the accelerometer bias drifts, in the domain of aerospace engineering, and modeled bias as a combination of time and temperature dependent variables [22]. By conducting a set of experiments in a controlled laboratory setting the author reported that 99% of the drifts could be eliminated. Batista et al. also proposed a calibration technique, based on time-varying Kalman filtering and gravity estimation, for estimating biases observed on *micro-electrical-mechanical-systems* accelerometers (tri-axial) [7]. Through a set of simulation experiments using a *motion rate table* the authors show good performance of their approach. Sensor biases are prevalent not only for accelerometers but also for other sensing modalities: For example, Barthold et al. studied the influence of heavy magnetic environments and motion on gyroscope-based orientation estimations [6]. The authors also point out that offline sensor calibration techniques are hindered by that the sensor heterogeneities, e.g., biases and offsets, are often dependent on time and temperature, thereby limiting the applicability of such calibration techniques.

Within the robotics community, calibration of IMUs via sensor fusion has been investigated, utilizing cameras and multiple sensors modalities, e.g., gyroscopes, magnetometers and accelerometers [26]. Banos et al. report on translating—across several sensor modalities—HAR from one sensor platform domain to another one, e.g., from a Kinect to an IMU [4]. Such methods could be used to align and potentially calibrate sensors to mitigate device heterogeneities.

In regards to the heterogeneities as discussed herein, a number of observations have been made in regards to the challenges these yield for real-world deployments: Amft [2] notes that HAR performances are often overly optimistic and that significant performance impairments may occur when utilized 'in the wild'. Blunck et al. [13] give a taxonomy of device-, user- and project-specific heterogeneities and sketch challenges for mobile app development and deployment projects. They further note that the recent rapid evolution of smartphone technologies increases these challenges.

Within the HAR community, usage heterogeneities in regards to device orientation and on-body placement have been thoroughly investigated. The severe HAR accuracy impairments these can cause have been investigated, initially for on-body-sensors but increasingly also for phones [3, 6, 9, 17, 25, 28, 44, 46]. For usage heterogeneity, various mitigation techniques for HAR have been proposed, e.g., body motion approximations aimed at on-body sensor setups [28], gravity vector estimation techniques for smartphones [25], as well as the training of several classifiers each associated with an individual phone orientation and/or placement, often combined with algorithms for identifying this context information from given data samples [17, 46].

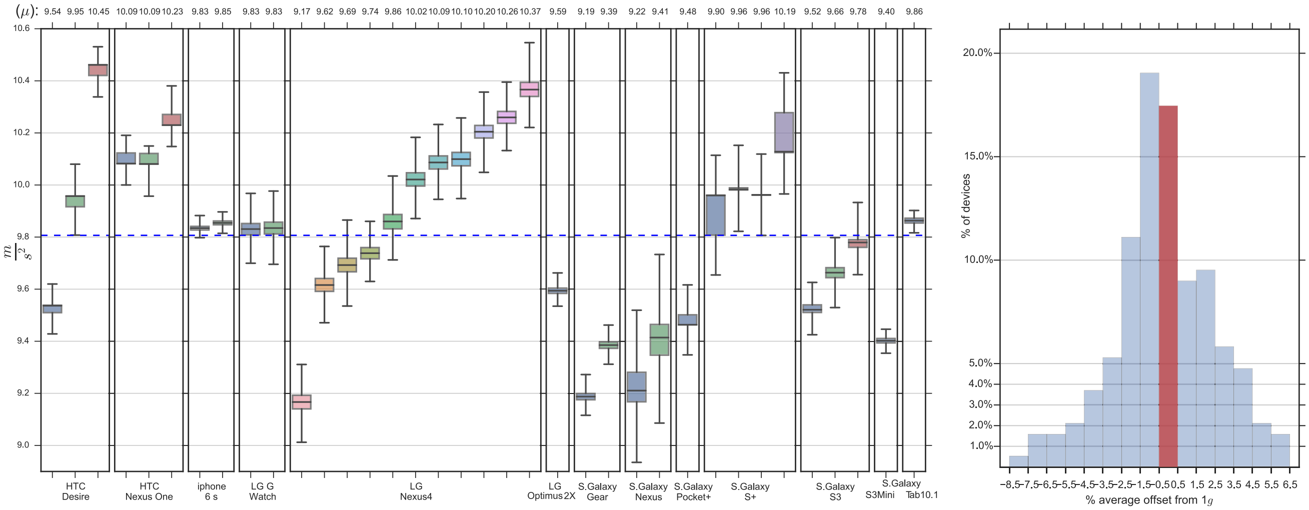
To the best of our knowledge, this paper is the first to rigorously assess both the heterogeneities as described in Section 1 in regards to challenges and impairments they come with, as well as potential mitigations of these for HAR, using a wide range of smart device models.

3. HETEROGENEITIES IN ACCELEROMETER-BASED SENSING

To systematically study heterogeneities in mobile sensing, in this section we report results from a number of static and non-static experiments focusing on accelerometer data from 36 mobile devices covering Android and iOS smartphones and several Android smartwatches. The details of the devices used in our study are listed in Table 1.

3.1 Sensor Biases (SB)

As with any motion sensors, accelerometer models (and instances) differ in precision, resolution, and range, and often suffer from various biases. Initial calibration is often done by the manufacturers using linear models, considering gains and offsets on each of the three accelerometer axes.



(a) Box plots of acceleration magnitudes and respective means (top) for investigated devices when lying on the back, with the whiskers indicating min/max values.

(b) Histogram of all mean deviations from Zero-***g***-offset.

Figure 1: Accelerometer biases of mobile devices in six different orientations under static conditions.

	#	Release	Max. sampling rate (Hz)
Smartwatch			
LG G	2	2014	200
Samsung Galaxy Gear	2	2013	100
Smartphone			
Apple iPhone 6	2	2014	100
Samsung Galaxy Pocket+	1	2013	100
Samsung Galaxy S3 mini	1	2012	100
Samsung Galaxy S3 mini	2	2012	100
LG Nexus 4	11	2012	200
Samsung Galaxy S3*	3	2012	150
Samsung Galaxy Nexus	2	2011	125
Samsung Galaxy S+	4	2011	50
LG Optimus 2X	1	2011	100
HTC Desire	3	2010	50
HTC Nexus One	3	2010	25
Tablet			
Samsung Galaxy Tab 10.1	1	2011	100

Table 1: Device models used in our experiments with number of instances used, year of release and accelerometer sampling rate; models selected for HAR investigation marked by an asterisk.

However, small errors may exist including rotation of the accelerometer package relative to the circuit board and misalignment of the circuit board to the final product. Errors can also be introduced during the soldering and final assembly process. Furthermore, if a device experiences shock, e.g., falling on the ground, the sensor can be misaligned causing unwanted biases.

In our first experiment we measure the *Zero-**g**-Offset* of devices, i.e., the bias observed when a device lies still on a flat surface (quasi static) and is exposed to only gravity, i.e., to an acceleration of $1\boldsymbol{g}\quad 9\ 81m\ \ s^2$. To measure Zero-***g***-Offsets, we collect data from all devices listed in Table 1 for a timespan of one minute while all devices lie still on their back side.[1] Subsequently, we repeat this procedure by cycling through all six sides of the devices, thereby generating six minutes of data per device. All the devices have previously been used in various user-centric research projects, and thus, to some extent, have experienced real life usage scenarios covering a wide range of domains and applications. We also collected ground truth using a highly accurate standalone IMU, an xsens MTx, to detect presence of any unwanted external forces. The mean and standard deviation of the IMU measurements were $\quad=9\ 82m\ \ s^2$ and $\chi=0\ 008$, respectively, when averaged over the entire data collection period. Figure 1(a) shows box plots, i.e., five point summaries (showing min, Q_1 (first quartile), median, Q_3 (third quartile) and max), of the gravity (under quasi static conditions) observed on individual devices. The respective mean readings are given at the top of the figure.

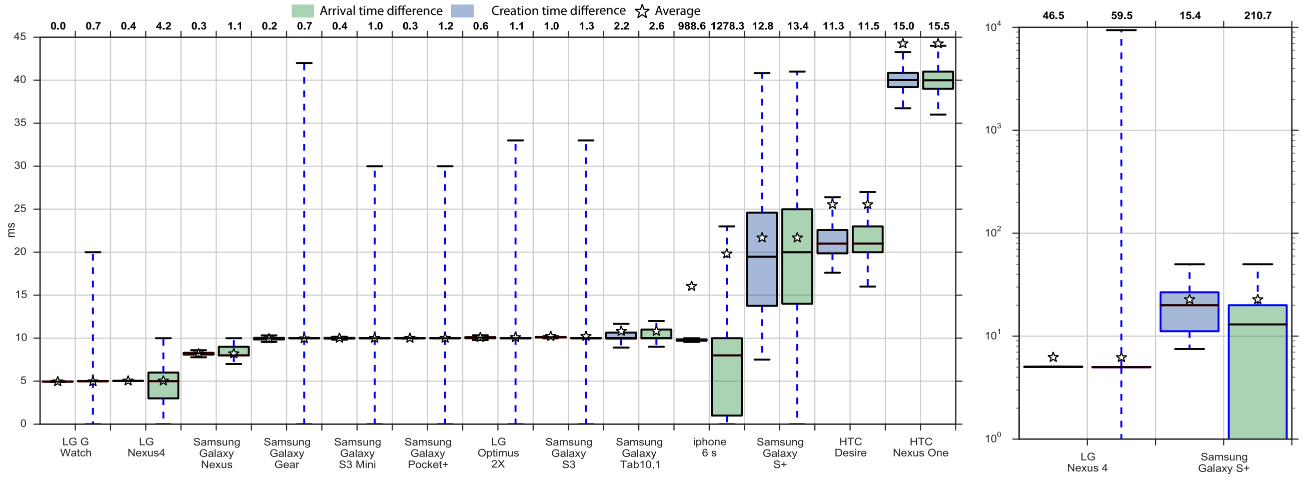
Interestingly, Figure 1(a) highlights not only the (often large) biases for different devices, but also the measurement volatility for some of them (see, e.g., the Samsung Galaxy Nexuses), and that biases and volatility may differ even for the devices of the same model (prominently for, e.g., the Galaxy S+ phones). Figure 1(a) is in line with a recent work stating that smartphones are often well distinguishable by their accelerometer fingerprint, which the authors define via its biases and variations [19]. Note, that for all of the six orientations, similar biases and volatilities were observed. To highlight the significance of the Zero-***g***-Offsets of the devices, Figure 10(b) shows a histogram of the mean deviations of estimated gravities from $1\boldsymbol{g}$ across individual experiments, i.e., for six orientations per device, using a bin width of $0\ 01\boldsymbol{g}$ (1%). Averaged offset biases, i.e., deviations, span from $\quad 8\ 5\%$ to $+7\ 5\%$ of $\boldsymbol{g}$ (i.e., $\quad 0\ 83m\ \ s^2$ to $0\ 74m\ \ s^2$). The figure also indicates a near Gaussian distribution of the biases across all devices with a mean of $9\ 76m\ \ s^2$. The absolute deviation, averaged over all runs was found to be $0\ 35m\ \ s^2$.

Overall, the biases for some tested devices correspond in magnitude to the acceleration as experienced during a train commute. Thus, for a HAR system these biases may easily lead to confusing activities or occupational states.

3.2 Sampling Rate Heterogeneity (SRH)

The maximum sampling rate supported by a mobile device varies significantly across device models and manufacturers. E.g., for the 14 device models investigated we observed 5 different maximal accelerometer sampling rates(see Table 1). With the heterogeneity among sampling rates for training and test devices come challenges for the HAR system design. One naive solution is to train an activity recognition system individually for each sampling rate to be encountered. Unfortunately, this

[1] Devices were configured to record accelerometer measurements on the local file system with their respective maximum sampling rate.



(a) Timespans between consecutive samples as estimated using using creation time (blue) and arrival time (green).

(b) Timespans when under high CPU load (y-axis in log-scale).

Figure 2: Variations in timespans between consecutive samples as estimated using using creation time (blue) and arrival time (green): shown in box plots, with whiskers at 1 5 IQR with outliers removed, and with standard deviation figures (top), for **accelerometer** readings.

solution does not scale because ground truth data collection becomes time consuming and costly [9]. As an alternative, sub- or super-sampling is often used as a pre-processing technique before training or applying a HAR system. We investigate the effects of sub- and super-sampling on HAR in Section 5—since, to the best of our knowledge, a systematic study of these has not been published yet.

3.3 Sampling Rate Instability (SRI)

Contrary to the sampling rate heterogeneity across devices, sampling rate instability is a phenomenon specific to a device. We define sampling rate stability as the regularity of the timespan between successive measurements. On most mobile platforms, there are two common ways to obtain timestamps for sensor measurements: using (i) *creation time*, i.e., the timestamp that the OS or device driver attaches to the sensor measurement and (ii) *arrival time*, i.e., the time when the measurement is received and time stamped by the application code. Ideally, the difference between these two timestamps should be close to zero and constant. This is often not the case for two reasons: Firstly, a small delay exists between the actual recording of a measurement on the chip and it being communicated to various applications. Secondly, the creation timestamp is often not guaranteed to be in the prescribed time format; instead the only guarantee is that the timestamps are monotonic. In this paper we use Network Time Protocol (NTP) to record arrival times, to synchronize with other devices, as well as for activity labeling.

To illustrate the sampling rate instability problem, Figure 2(a) shows for all device models listed in Table 1 tukey box plots for the timespans between consecutive samples—according to creation (blue) and arrival time (green), respectively. Furthermore, the whiskers in the box plots represent here the 1 5 IQR (interquartile range); outliers, i.e., samples outside the 1 5 IQR, are removed for better readability. Stars indicate the average timespan between consecutive samples—which for many devices is set far apart from the median, indicating the there exist not overly many, but long-timespan outliers. For each device model, the timespans aggregated in the box plot are those observed in the Zero-***g***-offset experiment. For most device models, the median timespan is reasonably close to the expected timespan, according to the sampling rates listed in Table 1. Thus, the figure also illustrates the sampling rate heterogeneity (SRH) for the devices listed. Furthermore, it is visible from both the plots, and more so from the standard deviations, that instabilities between measurements are larger for arrival time than for creation time (prominently, e.g., for LG Nexus 4)—and the same holds for the time span ranges.

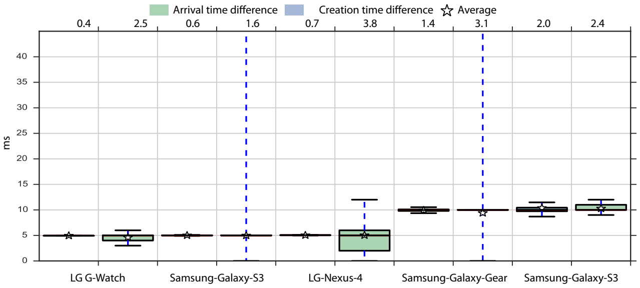


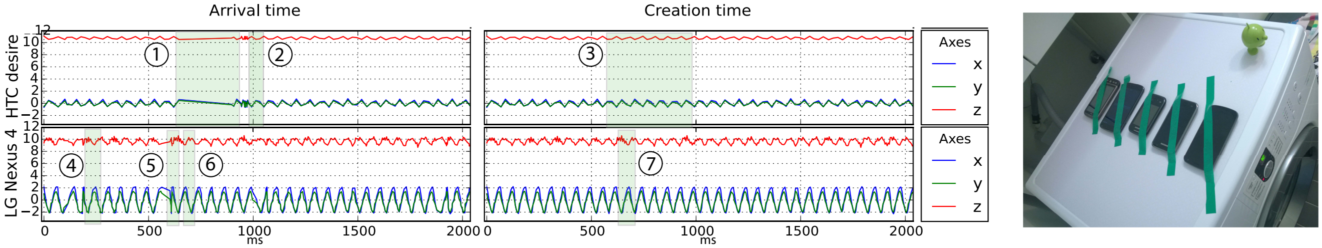
Figure 3: Timespans between consecutive samples—as estimated using using creation time (blue) and arrival time (green); shown in box plots, with whiskers at 1 5 IQR, with outliers removed, and with standard deviation figures (top), for **gyroscope** readings.

The phenomena generalize beyond accelerometer readings. For example, Figure 3 shows results for gyroscope in place of accelerometer readings. The general picture is the same as in Figure 2(a), and also the characteristics for individual devices are similar as well—although the Samsung Galaxy Gear and S3 devices seemingly exhibit even more and larger arrival time outliers, stemming again from some measurements being received in 'bursts'.

Another interesting anomaly is revealed for the iPhone accelerometer data: As the operating systems occasionally decides to suspend the sensing app used (seemingly, often after the phone has been still for several minutes), few very long timespans are introduced, which lead to very high standard deviations. As it turned out, this feature of iOS made it also hard to compare fairly the iPhones' HAR performance with other device models, and we thus decided to not use the data collected with iPhones in the analysis of the HAR experiments to be presented.

Multitasking Effects (ME)

Modern smartphones support running a large number of applications each of which may cause high CPU loads for extended periods of time. The OS often prioritizes among various running tasks and in doing so may affect the sensor sampling of one or several HAR applications



(a) Accelerometer readings on all three axes for two seconds during washing machine run.

(b) Experimental setup.

Figure 4: Effects of heterogeneities on accelerometer readings from phones lying on a running washing machine.

running on the device. To study the multitasking effects on sensor sampling we run a benchmarking application, inducing high loads, on four phones (two instances each of LG Nexus 4 and Samsung Galaxy S Plus) for 4 minutes, while they are configured to sample accelerometer readings at the device's maximum rate. Figure 2(b) shows box plots of the timespans between measurements recorded on the devices (creation and arrival time) under such heavy CPU load. Note that the figure's y-axis yields a logarithmic scale to accommodate the large outliers. The median timespan rises from 5 $2ms$ to 9 $2ms$, almost twice the intended timespan, c.f. Figure 2(a), and that the longest timespans between measurements experienced are over $1s$. As also in the experiment with no additional loads, the range of timespan is lower for creation time than for arrival time for both the device models. Interestingly, for Samsung Galaxy S Plus phones the average timespan between measurements for arrival time (33 $2ms$) is almost twice that for creation time (17 $7ms$). Thus, our experiments indicate that high CPU load impact actual sampling rates and instability very differently across devices models even if they run the same OS version and the same concurrent programs.

3.4 Heterogeneity Effects in Non-static Scenarios

Mostly, to illustrate the impacts of heterogeneities more visually and intuitively, we conducted another experiment, in a dynamic scenario yielding regular, i.e., stable and periodic movement. To this end, we taped several devices on top of a washing machine and recorded data during a regular washing program, see Figure 4(b). Figure 4(a) shows examples of the resulting accelerometer measurements during a two second exampled period. These measurements are given with timestamps according to their arrival time (left) and their creation time (right), respectively. Overall, the periodicity of the accelerometer measurements is clearly visible. However, the measurements show various anomalies, which are related to the heterogeneities described above. For both example devices measurement outages according to their arrival time are observed as measurements are delivered in bulks, see, e.g., the readings highlighted in green and marked (1), (2), (5) and (6). Whereas, the creation timestamps reveal that the measurements were instead taking almost equidistantly in time, e.g., compare (3) to (1) and (2). However, they may not always be collected equidistantly: (7) shows some variations in timespan between samples, smaller though than indicated by arrival time, at (5) and (6). Overall, the plots clearly illustrate some of the pitfalls in mobile sensing, and these may be harmful for HAR, especially when the use case renders it crucial to sensitively recognize and timestamp, e.g., short events, and/or starts or stops of, or breaks during activities.

4. IMPACTS OF HETEROGENEITIES ON HAR

In the following, we analyze the effect of various sensing heterogeneities, as detailed and investigated in Section 3, on the performances of HAR systems, using results from a case study.

4.1 HAR Case Study

We follow Bulling et al.'s exposition of HAR systems using supervised learning approach, which is composed of a sequence of signal processing and machine learning techniques, coined *activity recognition chain* (ARC) [15]. A typical ARC for HAR is composed of five stages: (i) *data collection*, (ii) *pre-processing*, (iii) *segmentation*, (iv) *feature extraction* and (v) *classification*. Although, accelerometer heterogeneities impact throughout the entire stages of an ARC, their effect is significant on the feature extraction stage. Accordingly, we study the impact of heterogeneities on three popular feature types in HAR research: (i) *time-domain*, (ii) *frequency-domain* features [20], and (iii) features extracted from *empirical cumulative distribution functions* (ECDF) [23].

Data Collection

In our case study, we consider the HAR task of detecting, and distinguishing among, six different user activities: 'Biking', 'Sitting', 'Standing', 'Walking', 'Stair Up' and 'Stair down'. Although simple, the selected activity classes (or a subset of them) have been investigated prominently in a large number of HAR publications [10, 11, 14, 27, 29, 41, 47]. We consider example data[2]gathered by nine users (age range 25 30 years). All users followed a scripted set of activities while carrying eight smartphones (2 instances of LG Nexus 4, Samsung Galaxy S+ and Samsung Galaxy S3 and S3 mini) and four smart watches (2 instances of LG G and Samsung Galaxy Gear). All eight smartphones were kept in a tight pouch and carried by the users around their waist, whereas two smart watches were worn on each arm. Each participant conducted five minutes of each activity, which ensured a near equal data distribution among activity classes (for each user and device).

Based on the results presented in the previous section, eight phones from two different manufacturers were selected to investigate diverse sensing scenarios. These smartphone models yielded different maximum sampling frequencies: 200 Hz for LG Nexus 4, 150 Hz for the Samsung Galaxy S3, 100 Hz for Samsung Galaxy S3 mini and 50 Hz for Samsung Galaxy S plus, approximately. The smart watches also varied in the supported maximum sampling rate, e.g., 200 Hz for LG G and 100 Hz for Samsung Galaxy Gear (see Figure 2(a)). Furthermore, the devices exhibit different accelerometer biases and gains.

To further minimize external factors affecting heterogeneities in accelerometer sensing, the data collection for the case study was carried out while keeping the CPU load of all phones minimal, i.e., only running the data collection application. Furthermore, we fixed also for each activity two environments and routes where the activity was to be executed. We then divided the user pool into two groups, each of which used the exact same environment and routes for their activities, respectively.

We limited our experiments to not consider explicitly differences in on-body placements and instead we will discuss these in Section 6. Consequently, we will compare data only among devices with simi-

[2] The data set has been made publicly available at **http://tinyurl.com/pxele7w**

lar on-body placements; in particular, we will analyze HAR results separately for smartphones and for smartwatches, respectively.

Pre-processing and Segmentation

For the scope of the evaluation of the effect of sensing heterogeneity on HAR in this section, we do not perform any preprocessing steps. We will though discuss pre-processing techniques for mitigation purposes in Section 5. Whenever timestamps are used in computations on the sensor measurements, e.g., frame extractions at the later stages of the ARC, we use creation time. In line with the standard approaches [9, 40], we employ a sliding window approach that overcomes the need for explicit semantic segmentations, which is a non-trivial task. In this work, we select a fixed window length of 2 seconds, with 50% overlap between two successive windows. Due to presence of heterogeneity in sensing, our measurement windows contain a variable number of measurements.

Feature Extraction

For evaluating the sensing heterogeneities on HAR for different features, we chose popular features and group them into several types. For the selection of popular time-domain and frequency domain features, we relied on the extensive survey on HAR techniques by Figo *et al.* [20].[3] These features are based on expert's knowledge, specific to the activity recognition domain, and have been used in a number of HAR systems. All the time-domain features, except the correlation-based features, are computed using motion sensor measurements from each axis x, y, z, and from the resulting reading magnitude, i.e., $\sqrt{x^2+y^2+z^2}$. The correlation features are computed by considering all possible pairwise combinations among the three axes.

The frequency features used in our experiments include the normalized spectral coefficients in the range from 0 to 20 Hz (or the maximum frequency available in the signal), with bin width of 0.5 Hz (i.e., 40 features in total), the entropy of the coefficients, the dominating frequency and its normalized magnitude, the normalized sum of the coefficients, and the discrete component.

ECDF features, on the other hand, do not depend on domain knowledge and are computed following the work of Hammerla et al. [23]. However, the ECDF features require setting a parameter value, namely the number of bins used in the inverse function computation. In our experiments we use 30 bins, which was previously shown to perform well in real world activity recognition tasks [23].

Furthermore we investigate the use of Principal Component Analysis (PCA). PCA is a popular dimensionality reduction technique within the field of machine learning and statistics [12]. PCA archives data compression by projecting data onto a linear orthogonal subspace such that the variance in the projected data is maximized. Contrary to the feature-engineering approaches popular within the HAR research, PCA can be used to learn features from sensor measurements in an unsupervised manner [9, 39].

Classification and Performance Metric

To evaluate the impact of heterogeneity on HAR performances we consider four popular classifiers used in activity recognition research, namely nearest neighbor classifier with $k = 5$, $C4.5$ decision tree, support vector machines (SVM) with linear kernels, and random forest classifier. We adopt the F_1-score, which is the harmonic mean of precision and recall, as our primary evaluation metric:

$$\text{F}_1\text{-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Moreover, in line with standard practices in HAR research [9, 43], we report the weighted average of the individual F_1-scores of all classes, which is more resilient to class imbalances in the test dataset, i.e.:

$$\text{Avg. F}_1\text{-score} = \frac{\sum_{i=1}^{c} w_i \cdot F_1^i\text{-score}}{\sum_{i=1}^{c} w_i}$$

where, F_1^i-score is the F_1-score of the i^{th} class and w_i is the number of samples of class i in the test dataset. Lastly, we perform statistical testing to measure variations in recognition performances using the McNemar χ^2-tests with Yates' correction [35].

4.2 HAR Performance Evaluation

To assess impacts of accelerometer heterogeneities on HAR, we perform a number of evaluations in different modes. These modes differ in how training vs. test are generated and separated from each other.

We present recognition performance in terms of F_1-scores in a sequence of figures, starting with Figure 5. Figure 5 shows in different colors the F_1-scores obtained by eight different training/test data definitions, explained further below, using smartphone data only. In particular, we will use the figure to quantify how drastically performance is reduced when the training was agnostic of the specific devices used in the test data. Results are shown in three plots, in each of which is used just one of the three main feature types considered, namely ECDF, frequency-, and time-domain features, respectively. In each of three plots, results are given individually for applying one of four learner types considered, namely C4.5 decision tree, linear support vector machines, K-nearest neighbor, and random forest ensemble learner.

Comparing evaluation modes: The first two training vs. test data generation are very traditional within machine learning and activity recognition, respectively:

Random 10-fold stratified cross validation (CV) randomly assigns one out of 10 folds to each data item. In each of the 10 folds over which performance is averaged afterwards, test data is recruited as the data items labeled with just one number out of 1 to 10. Within activity recognition (and generally with high-frequent time series data input) this evaluation usually leads to overfitting, and thus to overly optimistic recognition performance estimates: for time series, a random labeling implies that for each test data item there will very likely be several training data items which are very close by temporally—and thus very similar, as they sample almost the exact same situation as the test data item. Thus, as expected, the scores for 10-fold CV are the best for each feature type and learner combination considered.

Leave-one-user-out cross validation mode tests in each fold with data from one user, and trains with data from all others; scores are averaged over folds, i.e., test users. This mode provides realistic performance, and results lower than for 10-fold CV are expected, as body, movement, and general execution of the scripted activities will differ among users.

The following four evaluation modes assess the impact of heterogeneities as they are expected in 'in the wild' scenarios: In the *leave-one-model-out cross validation* training data is used from all considered phone models, but one, saved for test data and thus yet unknown to the HAR system. The *one-model* evaluation mode resembles training with only one phone model (as done in many HAR publications)—but testing is done with all the variety of the other considered phone types. As such, this mode resembles a typical low-effort real-world deployment scenario. Cross validation then averages over the training phone models selected. In a variant of this mode, in the *one-model-leave-one-user-out* mode, additionally one user's data is used for testing, but not for training. Cross-

[3] More specifically, we used all time- and frequency domain features, as listed in Table 2 and 3 of [20], respectively, with the exception of the cross-time/frequency-domain Wavelet feature.

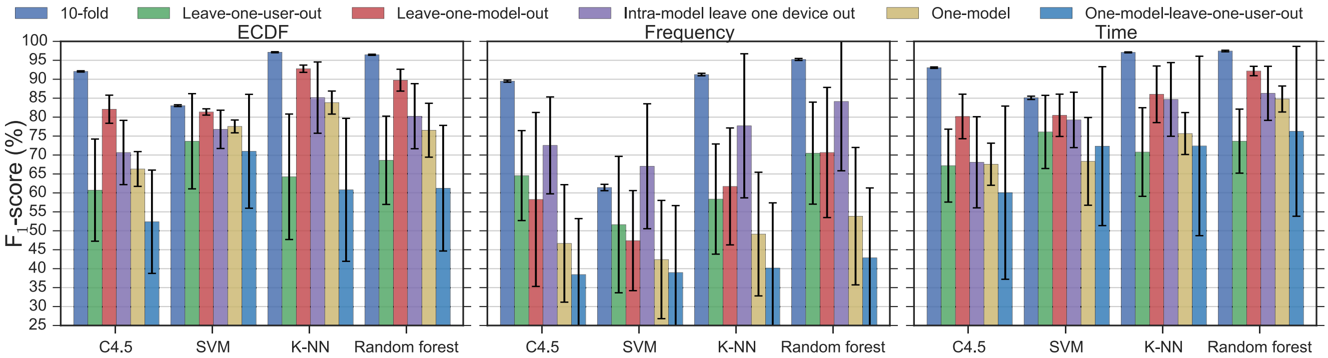


Figure 5: Comparing activity recognition performances on **smartphones** across three feature types (ECDF, Frequency and Time domain), across four classifiers, and across various cross-validation evaluation modes. The error bars represent the standard deviations across folds.

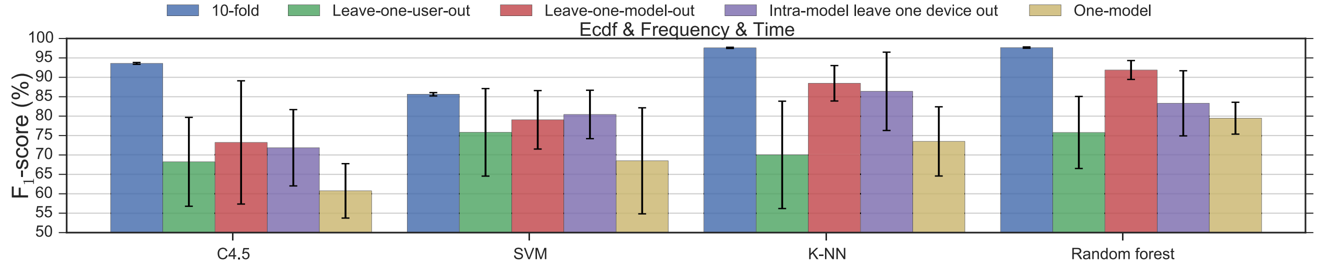


Figure 6: Comparing activity recognition performances on **smartphones** using three feature types (ECDF, Frequency and Time domain) **combined**, across four learner types, and across various cross-validation evaluation modes. The error bars represent the standard deviations across folds.

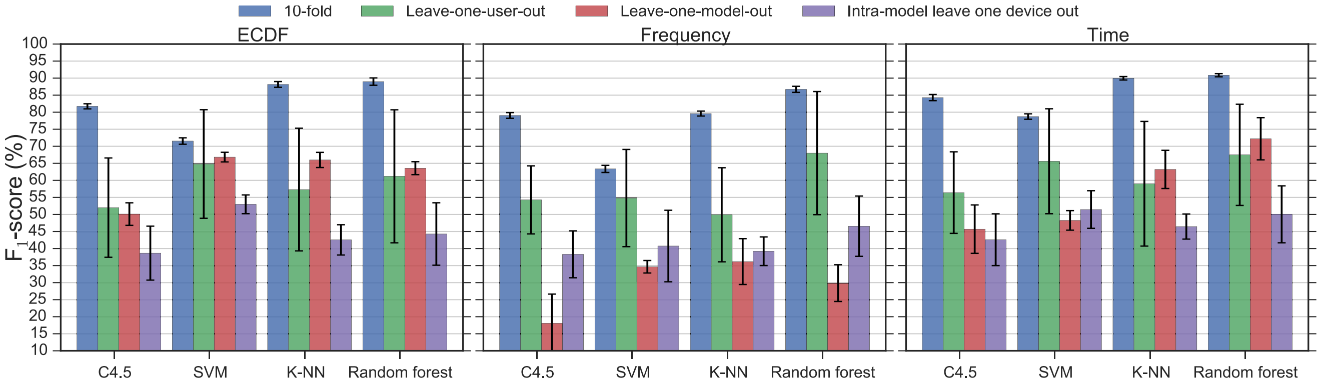


Figure 7: Comparing activity recognition performances on **smartwatches** across three feature types (ECDF, Frequency and Time domain), across four classifiers, and across various cross-validation evaluation modes. The error bars represent the standard deviations across folds.

validation than occurs by cycling through training phone model and test user. From the latter results one can assess the breakdown in F_1-score when training only with one device (56.4%, averaged over all learner and feature types) in comparison to the leave-one-use-out mode (66.7%).

Conversely, in *intra-model-leave-one-device-out cross validation* mode the test model is known to HAR system (and has been trained for specifically) but the particular test phone instance of the model is yet unknown. Despite that, classification with only the model's data yields significantly higher F_1-scores (on average: 77.7%) than for the *one-model* mode. Generally, the comparison of 'intra-model' scores over 'one-model' scores provides an estimate of the benefits of training classifiers specifically for individual phone models, i.e., up to 30 percentage points for some learner and feature types combinations. Such training of a classifier for each phone model individually is infeasible for most app deployment scenarios, and thus in Section 5 we will propose a more feasible training scheme requiring less training devices—via clustering of devices yielding similar characteristics.

Comparing features and learner types: We compare now the performances shown in Figure 5 across the three considered feature types. Complementing, in Figure 6 shows the performance when all three types are used together. From the figures, the combination yields higher scores (for all evaluation modes) than when using any of the three types alone. Using the features types alone, all offer similar scores when evaluated using the traditional *10-fold* or *leave-one-user-out* evaluation—but the frequency-domain features are impacted by far the most by the heterogeneities. Next, for the time features we see that the *leave-one-user-out* mode yields the lowest results compared to other modes. Furthermore, for the tree based learners (random forest and C4.5), the *leave-one-model-out* validations yield higher scores than the intra-model based mode. Thus, the time features are not impacted as much by the sensor heterogeneities as by the variance in the activities across users. Whereas, in contrast to the frequency domain features, sampling rate variability has less of an impact compared to the sensor biases. For the SVM and k-nn the heterogeneities seem to impact equally on the HAR system's score, as the results are quite similar for all but the *10-fold* and *leave-one-user-out* cross validation.

The effect of learner choice is more significant in the presence of heterogeneities. Specifically, ensemble learners do well, likely due to

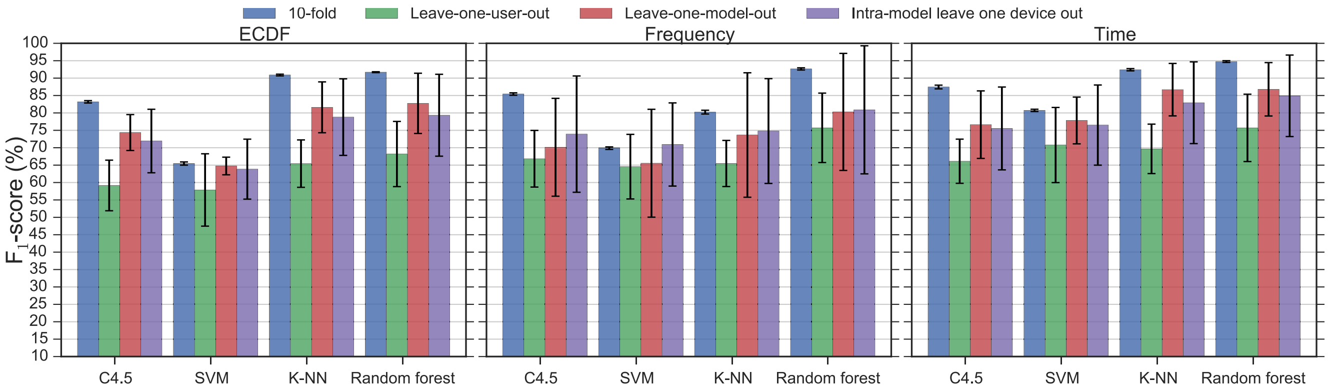


Figure 8: Comparing activity recognition performances on **smartphones** across three feature types (ECDF, Frequency and Time domain), across four classifiers, and across various cross-validation evaluation modes, when using **gyroscope** in place of accelerometer sensor data. The error bars represent the standard deviations across folds.

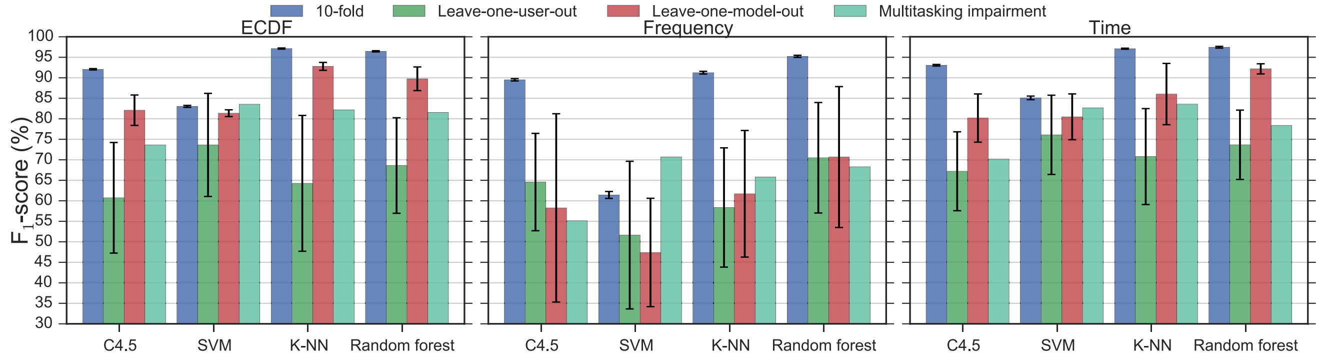
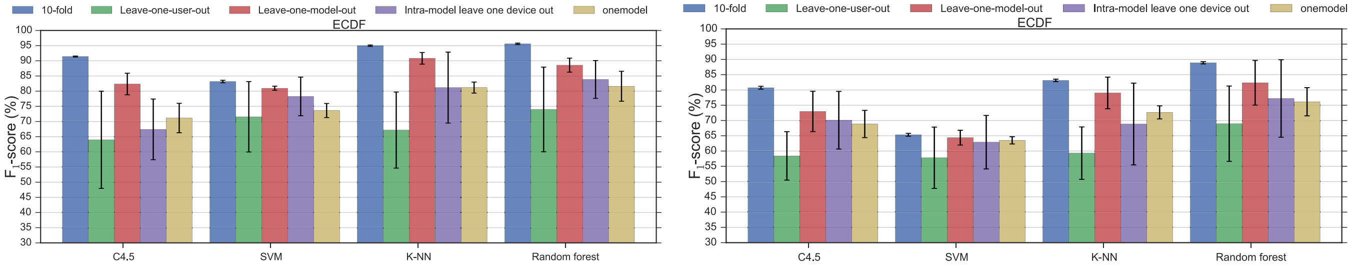


Figure 9: Comparing activity recognition performances on **smartphones** across three feature types (ECDF, Frequency and Time domain), across four classifiers, using the 10-fold, leave-one-user-out,leave-one-model-out and **multitasking-impairment** cross validations. The error bars represent the standard deviations across folds.



(a) Results from feature learning with **accelerometers**

(b) Results from feature learning with **gyroscopes**

Figure 10: Comparing activity recognition performances on **smartphones** across three feature types (ECDF, Frequency and Time domain), across four classifiers, and across various cross-validation evaluation modes, when using **PCA feature-learning**. The error bars represent the standard deviations across folds.

their ability to abstract from (and not overfit to) characteristics particular to the used devices. For the machine-generated ECDF features the picture is different though: performance across learners is more similar, and the k-nn learner outperforms the random forest. In addition, we see for the ECDF and Time domain features that both the leave-one-user-and-model-out yields lower scores than the leave-one-user-out across all learners, whereas for the frequency features it is all learners except the k-nn. This shows that the combined effect of user variance and the sensor heterogeneity impacts the learners more than the effects individually.

Comparing smartphones and smartwatches: As the sensor orientation, position, and dynamics of human arm movements are significantly different between our smartphone and smartwatch data, we repeated the evaluations discussed above for smartwatches in place of smartphones, see Figure 7: compared to the phones' scores, the relative smartwatch HAR scores across learner and feature types are generally impacted in a similar manner; only the absolute impairment levels are somewhat worse than for the phones. The latter observation may though be foremost due to the higher complexity in experienced user motion (which increases the challenge to tell apart the scripted activity types).

Comparing accelerometer vs. gyroscope data: As an indication of the generalizability of the results to other sensing setups, we used the gyroscope data in place of the accelerometer data, see Figure 8: The resulting relative performance across evaluation mode, feature, and learner types is very similar to that of accelerometer data. In absolute performance, the accelerometer scores are slightly higher (and, not shown, slightly lower than for fusing accelerometer and gyroscope data) which fits the intuition that the accelerometer yields somewhat higher amounts of motion clues and information, especially since the features used, while well applicable to gyroscope data, were designed for accelerometer data.

Comparing single- vs. multitasking: We also evaluated the impairments on HAR caused by multitasking: To this end, we trained using single-tasking data, and tested using multitasking-impaired data, see

Figure 9. The impairments are most severe for frequency features—due the unstable sampling frequencies during heavy multitasking.

Comparing with PCA-based feature learning Finally, we consider PCA based feature learning by constructing a feature space retaining 95% of the energy in the data. The results of the PCA experiments are given in Figure 10. PCA suffers from the blind normalization problem [23, 39] and the ECDF feature learning help to overcome it—thus, in Figure 10 these features show the best performance: Especially for easily overfitting learners, such as the C4.5, the results for, e.g., the one-model mode are improved over those without PCA, c.f. Figure 5. For other feature types, PCA application proves harmful, lowering the performance results. Additionally, the dimensionality reduction achieved by PCA helps to run the inference task faster, which helps to improve the overall energy consumption to a low level on resource constraint devices.

Summary of results: The evaluations show that heterogeneities can cause severe performance breakdowns in real-world scenarios, especially when compared to standard evaluation modes, such as the overly optimistic *10-fold* cross-validation, yielding an average F_1-score of 91 0%, or even the more realistic *leave-one-user-out* cross-validation, yielding 66.7%, especially if training is undertaken with only one device model (yielding 56 4% when additionally leaving one-user-out in training). When instead training with diverse (here: all but one investigated model), performances are much improved, and impairments for not training with the one test model are lower than for not training with the test user. Overall, breakdowns are most significant i) for frequency-domain features–and especially when the phone is multitasking heavily, c.f. Figure 9, and ii) when learners are used that are prone to overfitting, such as the C4.5 tree learner. In such learning configurations, the use of PCA may improve performance, see Figure 10. Generally, claims made in this summary still apply when using smartwatches instead of -phones, see Figure 7, and when using gyroscope instead of accelerometer data, see Figure 8.

5. METHODS FOR MITIGATION OF HETEROGENEITY EFFECTS

In the following, we investigate several candidate techniques for mitigating the impairments caused by heterogeneities, as described in the previous section. These candidate techniques pervade several stages of the activity recognition chain (ARC). As the two primary mitigation approaches, we will evaluate first a clustering of devices for similar devices, resulting in a classifier for each identified cluster. Secondly, we investigate the re-sampling of the sensor data from all devices to a common identical sampling rate, resulting also in equidistantly spaced samples for all devices after the preprocessing step. Finally, we also analyze in more detail the respective performance, of various combinations of mitigation techniques with various feature and leaner types, partially in order to arrive at favorable combinations to recommend.

5.1 Sensor Heterogeneity Based Clustering

The comparison of the recognition performance in Figure 5 of *one-model* vs. *intra-model-leave-one-device-out* mode reveals the advantage of training classifiers for similar devices—here: for devices of the same model. However, training for each device model individually comes with great costs and is thus not a practical deployable solution due to the vast amount of different device models. Furthermore, as presented in Section 4.3, also the sensor bias may vary greatly for devices of the same model, and such biases also impact the HAR system's performance—which is not taken into account with device model-based learners. We thus explore a less costly approach to account for diverse devices, based on clustering devices w.r.t. their respective heterogeneities: We train for each identified cluster a targeted classifier, in order to account account for devices' individual bias characteristics as well as for model characteristics such as default sampling rate. To this end, we chose an affinity propagation clustering algorithm [21]. Figure 11 visualizes the resulting clusters and the clustered devices in the feature space spanned by the following two features: median sensor bias, and standard deviation of sampling frequency. The respective feature data is taken from the static experiment described in Section 3. In Figure 12 we present results for employing clustering of devices using the two features as above as well as the standard deviation of the sensor bias and of the median sampling frequency. For the 36 devices, of in total 13 different models, the clustering algorithm chosen generated 5 distinct clusters. The cluster-based approach is evaluated in various *intra-cluster*-based modes in Figure 12—which use classifiers built specifically from the training data of a test device's device cluster. Visibly, modes such as *intra-cluster-leave-one-user-out* show significantly higher F_1-scores than for training with only one device model, c.f. Figure 5. Secondly, performance is also higher than when training with data from all devices (green), at 69.1% vs. 66.7%.[4] Furthermore, the *intra-cluster-leave-one-user-and-device-out* results indicate that training classifiers for whole device clusters is as accurate as the for real-world deployments much more laborious and thus less realistic alternative of training classifiers for each individual phone model: When comparing to the *intra-model-leave-one-user-and-device-out* modes, the F_1-score averaged over all feature and learner types is even higher, at 64.2% vs. 61.7%.[5]

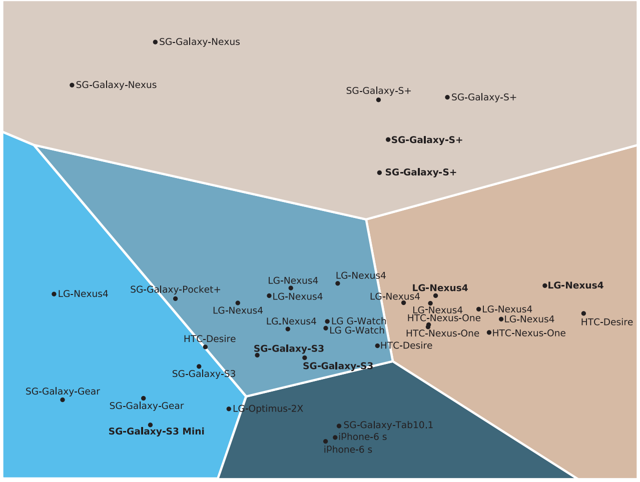


Figure 11: Device instances, clustered by features representing characteristic device heterogeneities. Here shown for the PCA-determined two most significant features, namely sensor bias and sampling frequency stability while devices are lying still on their back. Bold labels mark devices used later for the HAR impact investigation, and white crosses mark cluster centroids.

5.2 Interpolation

In the following, we investigate interpolation as a means to mitigate the heterogeneities in sampling rates across device models, and to improve the performances shown in Figures 5 and following. Specifically, we interpolate via down- or up-sampling of input sensor data to a common target rate of choice for all devices, varying the interpolation methods and target rates. Unifying to a common target rate, as a pre-processing step within ARC for HAR, ensures that i) each time window of data fed

[4] When excluding from the training the data from the test device's cluster (red), performance drops vary and are especially steep for frequency features and for learners prone to overfitting such as C4.5.

[5] Note though, that the respective results suffer from the limits of our data collection: for two out of the three clusters, we can train only with one phone, and thus the learning is prone to overfitting.

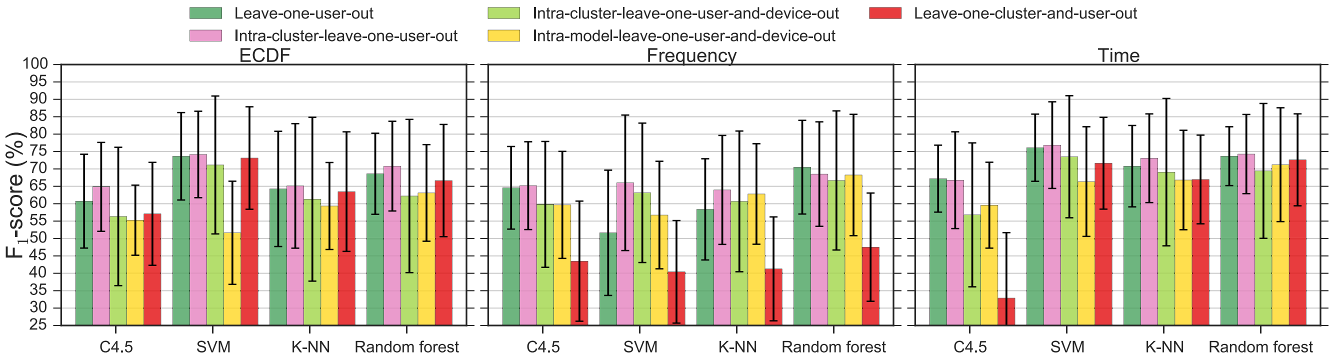


Figure 12: Activity recognition performance using *leave-one-user-out* as a baseline and variants thereof where classifying a test instance uses a classifier trained specifically for the cluster (resp. the model) a respective test device belongs to. The error bars represent the standard deviations across folds.

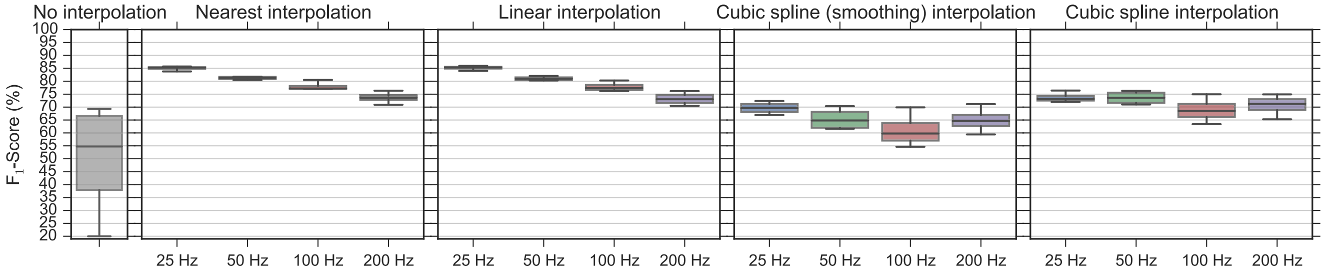


Figure 13: Activity recognition performance using the solely frequency domain features and using the SVM learner; for various interpolation methods and target rates, and using the *leave-one-model-out* cross validation.

into the ARC segment step contains the same number of (interpolated) samples, and ii) that all samples in each data window are equidistantly separated in time.

Interpolation techniques: The following four interpolation schemes were used as part of the preprocessing stage:

Nearest: The interpolation value at time t is set to be the value of the one of the input samples that is closest in time to t.

Linear interpolator: The value at time t is the piecewise-linear interpolation, i.e. the linear interpolation between the input samples adjacent to t in the sequence of input sample timestamps.

Cubic Splines: uses cubic splines, i.e., piecewise polynomial interpolation of degree 3 with continuous first and second derivative.

Smoothed cubic splines: Splines as above but subject to a positive smoothing factor s, which reduces the number of interpolated input samples in favor of fulfilling a smoothness condition: $^{m}_{i=1}(y[i]$ $spl(x[i]))^2$ s , where m is the number of input samples considered, and s is 0 2 m.

The target sample rates which have been evaluated in this case study were the following: 25, 50, 100, and 200 Hz. In Figure 13 we present results for all combinations of these interpolation methods and target sampling rates. The results were obtained using *leave-one-model-out* cross validation and using learner and feature type pairing which was most impaired by sampling rate heterogeneities and instabilities, c.f. Figure 5, namely frequency features and SVM. Clearly, generally interpolation can improve the HAR accuracy: Highest gains (for median accuracy: from 55% to ca. 86%) are achieved for low target rates, and for nearest and linear interpolation. Interestingly, for more complex interpolation methods (i.e. for cubic but also other ones) less gains (or, for less impaired feature/learner combinations, even negative gains) result, and also no clear trend favoring lower target rates can be seen. Both is likely explained by that complex interpolations may introduce noise and artifacts in the data—which impedes learners, as they may learn from artificially introduced characteristics instead of from those present in the actual sensor data. This factor likely also explains why down-sampling yields higher gains than up-sampling: Also up-sampling is prone to introduce artifacts in the data, while down-sampling is likely to just preserve the real data's characteristics.

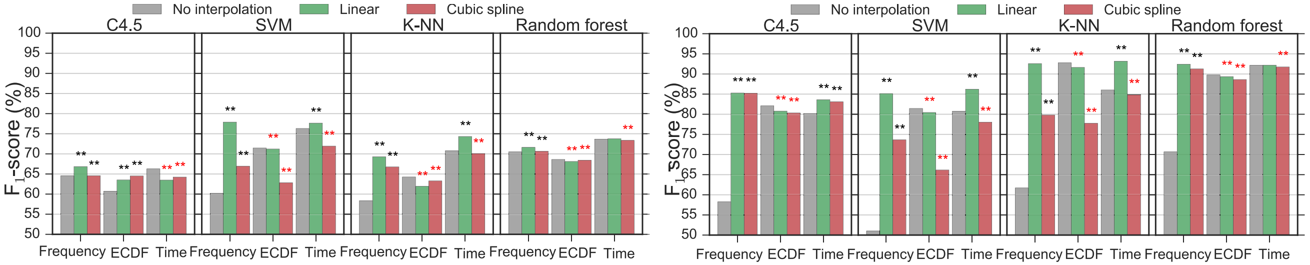
This holds at least for most HAR-typical activity types—as those chosen for our experiments—because their characteristics are rather low-frequent; thus, these characteristics are likely to preserved when down-sampling to 25 Hz. Note though that this may not hold for more exotic HAR tasks and strategies, e.g. for distinguishing (riding in) fossil-fueled from electric vehicles by way of recognizing the characteristic engine vibrations in the former cars.

Impact on Sampling Frequency Heterogeneities

The results presented in Section4 revealed that the differences in sampling frequencies across device models impairs HAR performance, and we now evaluated to which extent these impairments can be mitigated via interpolation. Applying down-sampling to 25Hz in pre-processing, yields results shown in 14(b), here for *leave-one-model-out* cross validations, as used also in Figure 13. In the figure, (**), (*) indicate when improvements were statistically significant at levels $p<0$ 01 and $p<0$ 05, respectively. Conversely, red asterisks are shown instead, if interpolation significantly impairs the performance.

Overall, the impact of the interpolation shows mixed results across learners, feature types and to some extent interpolation methods. Furthermore, in line with Figure 13, linear and nearest interpolation perform better than the spline interpolation variants.

For **frequency features**, interpolation significantly ($p<0$ 001) increased the original performance (labeled 'No interpolation') across all learners and interpolation methods—up to more than 30 percentage points in case of using the SVM learner. Interpolation is though far less helpful for other feature types: For **time domain features** nearest and linear interpolation still improve performance, but spline interpolation impairs it. For **ECDF features**, interpolation has little to no positive effect and in most cases will impair the performance.



(a) Results from the *leave-one-user-out*

(b) Results from the *leave-one-model-out*

Figure 14: Activity recognition performance across all devices using *leave-one-user-out* and *leave-one-model-out* cross validations. (**),(*), for the following significance levels $p < 0$ 01, $p < 0$ 05 comparing interpolated with 25 Hz sampling frequency and non-interpolated data.

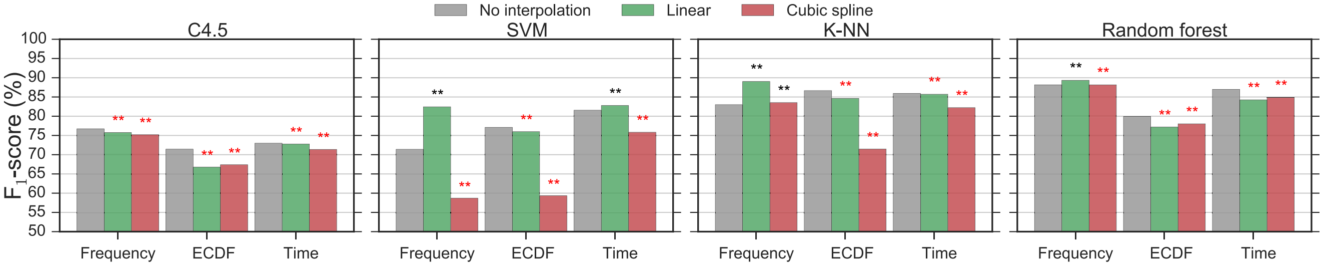


Figure 15: Activity recognition performance using *Intra-cluster-leave-one-device-out* evaluating learners trained with clusters of similar devices. (**),(*), for the following significance levels $p < 0$ 01, $p < 0$ 05 comparing interpolated with 25 Hz sampling frequency and non-interpolated data.

For comparison, Figure 14(a) evaluates the same interpolation options as shown in Figure 14(b) but for the *leave-one-user-out* evaluation mode. Here, since training has seen the devices used in testing, unlike in Figure 14(b), interpolation has lost most of its positive effects, even when applied for frequency features. Note that the interpolation methods nearest (resp. smoothed cubic splines) are not shown in the figure, but perform very similar to linear (resp. cubic spline) interpolation.

5.3 Combining Interpolation and Clustering

We have also studied the impact of interpolating when combined with clustering devices with similar heterogeneities. These results are reported in Figure 15, and—similar to Figure 14(a)—only in a few instances the interpolation actually improves the HAR performance, e.g., for frequency features in SVM, K-NN and random forest ($p < 0$ 001). This poor performance is likely due to that the clusters take into account the sampling frequencies and respective instabilities, and thus devices with similar frequencies and instabilities are in the same cluster and the potential benefits from employing sample rate conversion via interpolation is lowered.

Overall, the results in Figure 14 and 15 indicate that interpolation, and specifically down-sampling, is most useful in case of heterogeneous sampling frequencies w.r.t. training vs. test data. Furthermore, results indicate that interpolation can mitigate the impairments caused by differing sampling frequencies, but not the issues of sampling instabilities. Further evidence of the latter is given by running training with multi-tasking impaired devices, c.f. Section 3: Here, the effects of interpolation on performance are not better (and largely even worse) then when training with non-impaired devices, indicating that instabilities are not mitigated by interpolation.

Summary of results: The evaluations in this section provide evidence that training classifiers specifically for clusters of devices with similar characteristics increases HAR performance cost-effectively. Furthermore, interpolation, specifically down-sampling, can improve HAR performance—specifically when frequency features are used, and when training and test devices differ in sampling frequency. If the latter is already mitigated, e.g., by a clustering based approach, interpolation is only recommendable for pre-processing for frequency-domain features, but should rather not be employed for time-domain or ECDF features.

6. DISCUSSION

In this section we discuss the generalizability of the presented results across new and next-generation classes of devices, as well as further motion and other sensors in place of or additional to accelerometer and gyroscope. Finally, we also discuss assumptions and limitations of the evaluation presented here.

Device Types and Evolution

A natural question to ask is whether the various heterogeneities across devices diminish as the user device technology evolves. Indeed, the results presented here seem to indicate a trend: Newer and more expensive models are likely to yield lower biases; in regards to sensor biases, this trend may though also be caused by the longer wear and tear that the tested instances of older model were exposed to compared to devices of newer models.

An initial intuition of ours that was clearly not backed by the results was that smartwatches, being of smaller form factor and with more limited resources than smartphones, would exhibit far larger sensor biases and sampling instabilities, especially under high CPU load. On the other hand, smaller devices are expected to be more single-purpose-build and less required to fulfill a wide range of tasks. Thus, less multitasking impairments may be expected in real-world use. While biases and instabilities were not stronger, in the actual activity recognition performance smartwatches though showed lower performance than the selected smartphones, c.f. Figure 5 and 7. The results are not conclusive though, in regards to whether that may be attributed to harmful heterogeneities or rather to the difference in learning settings, as the on-body placement is vastly different from the task to learn from the smartphones' data, which were residing in a hip-mounted pouch.

A natural extension of the study is to extend the type of investigated user devices further, beyond tablets, smartphones and -watches and to other, popular or emerging mobile devices, specifically wearables, such as smart earphones and wristbands. Such investigation may be fruitful, as our results on the hypothesis that smaller, less powerful devices suffer more from heterogeneity or its impacts on HAR performance were inconclusive when comparing smartphones with -watches, see above.

A similar extension is to obtain and compare with more results from devices running other OS. While we provide some results for iOS devices, the automatic-sleep phenomena, c.f. Section 3, hindered a fair comparison with other devices in the HAR evaluations. We undertook a preliminary investigation, collecting data from Android but also some iPhones of some 50 CS students in a setup as described in Section 3. The results showed that the frequency irregularities are of similar magnitude than for the average over the investigated Android phones.. Similarly, also the quality of and biases within the acceleration measurement sets themselves were comparable to those stated in Section 3, with deviation from 1G of up to a $\pm 2.5\%$, and an average standard deviation of 0.039G.

Feature types

In this paper we have shown that the performance of three feature types, i.e., time-domain, frequency-domain and ECDF features, have greatly varying performances in HAR in the presence of sensor heterogeneities. Furthermore, the frequency features, without preprocessing, have been shown to be most vulnerable in heterogeneous settings. Thus, based on this case study, especially ECDF but also time features are strongly recommended for HAR when sensor heterogeneities are present.

However, other domains of HAR uses domain specific knowledge to look for specific frequencies, e.g., for detecting freezing of gait in patients with Parkinson's disease [34]. Thus, interchanging features might not be an applicable strategy in all use cases, as it was in our case study. For these instances, based on our case study we have shown that preprocessing the signal with simple interpolation methods will significantly increase the performance, when sensor heterogeneities are present.

Sensor Types

Another extension of the study presented here is to consider popular in-device sensor types other than the accelerometer and gyroscope. Regarding issues with heterogeneities of sampling frequencies across devices, and regarding irregularities of sampling frequency, a natural expectation is that it will affect sampling from other in-device sensors equally as the thoroughly investigated accelerometer sensor. Our results for investigating the gyroscope support this hypothesis. On the other hand, we expect these impairments to be less severe on HAR for sensors which sample at significantly lower frequency or for which high-frequency sampling and sampling stability is less crucial, e.g. for location sensors or for the magnetometer when used to obtain mere device orientation.

Furthermore, our evaluation of sensor biases focused largely on static setups. Regarding varying and mediocre quality of sensing, other sensor types have varying characteristics. Additionally, for some sensors, such as the magnetometer or GNSS-based location sensors, the heterogeneities induced by the user's current environment are much more severe: While the accelerometer is biased only to a small extent, specifically by the ambient temperature, a magnetometer is heavily biased by magnetic inference as induced by, e.g., building elements and installations, or motor vehicles. Furthermore, for many sensors, biases such as gain and offset are typically temperature dependent [26], and, e.g., during the stress test of the phones, c.f. Section 3 the temperatures were noticeable hotter, and the authors are unaware whether the phones have built-in control of temperature dependent calibration. Thus, during the stress test the phones' biases and offsets may have changed due to the higher temperature.

Combined Mitigating of Further Heterogeneities

Beyond the device-centric heterogeneities focused on in this paper, see Section 1, further heterogeneity types are present in most real-world HAR scenarios [2, 13], prominent among which are device orientation and on-body placement. Several of the mitigation techniques for the latter, see Section 2, follow a divide&conquer approach via training classifiers for similar configurations (e.g., w.r.t. placement and orientation) as does our clustering technique (applied for device characteristics). The same concept can be applied to both these dimensions (and to even more) simultaneously—whereas the number of classifiers to be trained then grows exponentially with the dimensions, i.e. heterogeneity types, considered. Also the estimation (and the thereby facilitated 'subtraction') of the gravity vector from acceleration data could be combined with the mitigation techniques described herein, as a pre-processing step.

7. CONCLUSIONS

In this paper, we have presented and analyzed heterogeneities present in the motion sensor output of common smartphones, -watches and tablets. We furthermore analyzed the impairments these heterogeneities cause for human activity recognition tasks. To this end, we presented results from several experiments investigating datasets, which are made public, involving in total 36 devices.

The investigation presented identifies and analyses the following three sensor heterogeneity categories, focusing on accelerometer and gyroscope sensors: Firstly sensor biases, which for some investigated devices showed in stillness deviation of 8% deviation from the sole exerted force, gravity—a bias large enough to account for the acceleration of a fast train. Secondly, severe sampling instabilities occur on many investigated devices, especially when these are running other tasks, which yield high loads. Finally, also the heterogeneous nominal sampling rates of different devices yield a big challenge for activity recognition on device models not yet seen in the training phase. Furthermore, we have investigated, using a case study, the impairments these heterogeneities yield for human activity recognition, as well as several techniques for mitigating these. As mitigation techniques in the form of preprocessing methods we investigated various interpolation schemes and low-pass filtering, as well as the clustering of devices according to their heterogeneity characteristics—which then allows to train classifiers targeted at individual device clusters. Additionally, we evaluate both impairments and mitigation effects for different feature types, namely time domain, frequency domain and ECDF features, and four learner types, namely C4.5 trees, SVMs, k-NN learners, and random forests. The impairments in the case study were significant, lowering the recognition accuracy in our case study by up to a third for some choices of feature and learner types, most notably for frequency-domain features. The mitigation techniques were shown to regain much of these heterogeneity-caused accuracy losses. Finally, we have discussed implications and potential extensions of the presented work.

8. ACKNOWLEDGEMENTS

This work has been supported by the Danish Advanced Technology Foundation under J.nr. 076-2011- 3., and by The Danish Council for Strategic Research as part of the EcoSense project (11-115331). Parts of this work were done during stays of the first author at Carnegie Melon, US, and of the second author at the University of Münster, Germany.

Furthermore, We would like to thank the reviewers and especially our sheperd Lama Nachman for their helpful suggestions and comments.

9. REFERENCES

[1] M. V. Albert, S. Toledo, M. Shapiro, and K. Kording. Using mobile phones for activity recognition in parkinson's patients. *Frontiers in neurology*, 3, 2012.

[2] O. Amft. On the need for quality standards in activity recognition using ubiquitous sensors. In *How To Do Good Research In Activity Recognition. Workshop in conjunction with Pervasive*, 2010.

[3] S. A. Antos, M. V. Albert, and K. P. Kording. Hand, belt, pocket or bag: Practical activity tracking with mobile phones. *Journal of Neuroscience Methods*, 231:22 – 30, 2014.

[4] O. Banos, A. Calatroni, M. Damas, H. Pomares, I. Rojas, H. Sagha, J. del R Millán, G. Troster, R. Chavarriaga, and D. Roggen. Kinect= imu? learning mimo signal mappings to automatically translate activity recognition systems across sensor modalities. In *IEEE Int. Symp. Wearable Computers (ISWC)*, 2012.

[5] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *2nd Intl. Conf. Pervasive Computing (Pervasive)*, pages 1–17, 2004.

[6] C. Barthold, K. Subbu, and R. Dantu. Evaluation of gyroscope-embedded mobile phones. In *IEEE Intl. Conf. Systems, Man, and Cybernetics (SMC)*, Oct 2011.

[7] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira. Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation. *IEEE Trans. Control Systems Technology*, 19(1):1128–1137, 2011.

[8] S. Bhattacharya, H. Blunck, M. Kjærgaard, and P. Nurmi. Robust and energy-efficient trajectory tracking for mobile devices. *IEEE Trans. Mobile Computing (TMC)*, 14(2):430–443, 2015.

[9] S. Bhattacharya, P. Nurmi, N. Hammerla, and T. Plötz. Using unlabeled data in a sparse-coding framework for human activity recognition. *Pervasive and Mobile Computing*, 15(0):242–262, 2014.

[10] G. Bieber, P. Koldrack, C. Sablowski, C. Peter, and B. Urban. Mobile physical activity recognition of stand-up and sit-down transitions for user behavior analysis. In *Intl. Conf. Pervasive Technologies Related to Assistive Environments (PETRA)*. ACM, 2010.

[11] G. Bieber, J. Voskamp, and B. Urban. Activity recognition for everyday life on mobile phones. In *Universal Access in Human-Computer Interaction (UAHCI)*. Springer, 2009.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] H. Blunck, N. O. Bouvin, T. Franke, K. Grønbæk, M. B. Kjærgaard, P. Lukowicz, and M. Wüstenberg. On heterogeneity in mobile sensing applications aiming at representative data collection. In *UbiComp '13 Adjunct*, pages 1087–1098. ACM, 2013.

[14] T. Brezmes, J.-L. Gorricho, and J. Cotrina. Activity recognition from accelerometer data on a mobile phone. In *Intl. Work-Conf. Artificial Neural Networks (IWANN)*. Springer, 2009.

[15] A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *Computing Surveys (CSUR*, 46(3), 2014.

[16] F. Buttussi and L. Chittaro. Mopet: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine*, 42(2):153 – 163, 2008.

[17] Y. Chen, Z. Zhao, S. Wang, and Z. Chen. Extreme learning machine-based device displacement free activity recognition model. *Soft Computing*, 16(9), 2012.

[18] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan. Mobile phone-based pervasive fall detection. *Personal Ubiquitous Comput.*, 14(7), 2010.

[19] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. *Network and Distributed System Security Symp. (NDSS)*, 2014.

[20] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.

[21] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[22] F. Gulmammadov. Analysis, modeling and compensation of bias drift in mems inertial sensors. IEEE 4th Intl. Conf. Recent Advances in Space Technologies (RAST), June 2009.

[23] N. Y. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *ISWC*. ACM, 2013.

[24] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *11th ACM Conf. Embedded Networked Sensor Systems (SenSys)*, 2013.

[25] A. Henpraserttae, S. Thiemjarus, and S. Marukatat. Accurate activity recognition using a mobile phone regardless of device orientation and location. In *IEEE Body Sensor Networks Conference (BSN)*, 2011.

[26] J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, Ultra-Wideband and GPS*. PhD thesis, Linköping University, Sweden, 2011.

[27] Y. Kawahara, H. Kurasawa, and H. Morikawa. Recognizing User Context Using Mobile Handsets with Acceleration Sensors. In *IEEE Intl. Conf. Portable Information Devices (PORTABLE)*, pages 1–5, 2007.

[28] K. Kunze and P. Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *ACM Intl. Joint Conf. on Pervasive and Ubiquitous Computing (UbiComp)*, pages 20–29. ACM, 2008.

[29] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. In *SIGKDD Explorations Newsletter*. ACM, 2011.

[30] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Communications Magazine, IEEE*, 48(9), 2010.

[31] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. In *4th Intl. Conf. Pervasive Computing (Pervasive)*, 2006.

[32] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille. A long-term evaluation of sensing modalities for activity recognition. In *Proceedings of the 10th international conference on Ubiquitous computing (UbiComp)*, 2007.

[33] J. C. Lötters, J. Schipper, P. H. Veltink, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. *Sensors and Actuators A: Physical*, 68(1-3):221–228, 1998.

[34] S. Mazilu, M. Hardegger, Z. Zhu, D. Roggen, G. Troster, M. Plotnik, and J. M. Hausdorff. Online detection of freezing of gait with smartphones and machine learning techniques. In *IEEE Intl. Conf. Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 123–130. IEEE, 2012.

[35] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[36] OpenSignal. Android Fragmentation Visualized. **`http://opensignal.com/reports/2014/android-fragmentation/`**, 2014. Accessed 17-Mar-2015.

[37] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen. Activity classification using realistic data from wearable sensors. *Biomedicine*, 10(1):119–128, 2006.

[38] C. Pham and P. Olivier. Slice&dice: Recognizing food preparation activities using embedded accelerometers. In *Intl. Conf. Ambient Intelligent (AmI)*, 2009.

[39] T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Intl. Joint Conf. Artificial Intelligence (IJCAI)*, volume 22, page 1729, 2011.

[40] T. Plötz, P. Moynihan, C. Pham, and P. Olivier. Activity recognition and healthier food preparation. In *Activity Recognition in Pervasive Intelligent Environments*. Atlantis Press, 2011.

[41] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Trans. Biomedical Engineering*, 56(3):871–879, 2009.

[42] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, 2010.

[43] H. Sagha, S. Digumarti, J. del R Millan, R. Chavarriaga, A. Calatroni, D. Roggen, and G. Tröster. Benchmarking classification techniques using the Opportunity human activity dataset. In *IEEE Intl. Conf. Systems, Man, and Cybernetics (SMC)*, 2011.

[44] P. Siirtola and J. Röning. Recognizing human activities user-independently on smartphones based on accelerometer data. *Intl. Journ. Interactive Multimedia and Artificial Intelligence*, 1(5), 2012.

[45] X. Su, H. Tong, and P. Ji. Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3):235–249, 2014.

[46] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li. Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations. In *Ubiquitous intelligence and computing (UIC)*. Springer, 2010.

[47] J. Yang. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *1st Intl. Workshop Interactive Multimedia for Consumer Electronics*, pages 1–10. ACM, 2009.