SenseHAR: A Robust Virtual Activity Sensor for Smartphones and Wearables

Jeya Vikranth Jeyakumar
University of California, Los Angeles
Los Angeles, California, USA
vikranth94@g.ucla.edu

Naveen Suda
Arm Inc.
San Jose, California, USA
naveen.suda@arm.com

Liangzhen Lai
Arm Inc.
San Jose, California, USA
liangzhen.lai@arm.com

Mani Srivastava
University of California, Los Angeles
Los Angeles, California, USA
mbs@ucla.edu

ABSTRACT

Modern smartphones and smartwatches are equipped with inertial sensors (accelerometer, gyroscope, and magnetometer) that can be used for Human Activity Recognition (HAR) to infer tasks such as daily activities, transportation modes and, gestures. HAR requires collecting raw inertial sensor values and training a machine learning model on the collected data. The challenge in this approach is that the models are trained for specific devices and device configurations whereas, in reality, the set of devices carried by a person may vary over time. Ideally, one would like activity inferencing to be robust of this variation and provide accurate predictions by making opportunistic use of information from available devices. Moreover, the devices may be located at different parts of the body (e.g. pocket, left and right wrist), may have different sets of sensors (e.g. a smartwatch may not have gyroscope while a smartphone might), and may differ in sampling frequencies. In this paper, we provide a solution which makes use of the information from available devices while being robust to their variations. Instead of training an end-to-end model for every permutation of device combinations and configurations, we propose a scalable deep learning based solution in which each device learns its own sensor fusion model that maps the raw sensor values to a shared low dimensional latent space which we call the ‘SenseHAR’-a virtual activity sensor. The virtual sensor has the same format and similar behavior regardless of the subset of devices, sensor’s availability, sampling rate, or a device’s location. This would help machine learning engineers to develop their application specific (e.g., from gesture recognition to activities of daily life) models in a hardware-agnostic manner based on this virtual activity sensor. Our evaluations show that an application model trained on SenseHAR achieves the state of the art accuracies of 95.32%, 74.22% and 93.13% on PAMAP2, Opportunity(gestures) and our collected datasets respectively.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SenSys '19, November 10–13, 2019, New York, NY, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6950-3/19/11...\$15.00

<https://doi.org/10.1145/3356250.3360032>

CCS CONCEPTS

• **Human-centered computing** → **Smartphones; Mobile devices**; • **Computing methodologies** → **Machine learning**; • **Hardware** → Sensor devices and platforms;

KEYWORDS

Human Activity Recognition, Sensor Fusion, Deep Learning

ACM Reference Format:

Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. 2019. SenseHAR: A Robust Virtual Activity Sensor for Smartphones and Wearables. In *The 17th ACM Conference on Embedded Networked Sensor Systems (SenSys '19)*, November 10–13, 2019, New York, NY, USA. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3356250.3360032>

1 INTRODUCTION

In the past decade, the market surrounding smartphones and wearable technology has grown exponentially [65]. Smartphones and other personal wearables used for fitness and health monitoring are cheap, ubiquitous and a survey shows that they are owned by more than 85% and 20% of the United States population [19, 31] respectively. These devices are equipped with multiple sensors [39] and Inertial measurement units (IMUs) are the most prominent sensors that have a combination of microelectromechanical (MEMS) accelerometers, gyroscopes, and magnetometers that enable the devices to infer the activities of a person. While, prior research identifies smartphones as the widely used sensing modality in HAR [44, 55], smartwatches are also being utilized extensively to infer diverse gestures for various applications such as sign language [8] and medical rehabilitation [9]. Today most people own multiple smart devices such as smartphones, smartwatch, fitness bands, smart shoes, smart belts, etc. and we should be able to leverage on the information from these multiple devices to improve the accuracy of activity recognition. This process of combining information from multiple devices and sensors is called Multimodal sensor fusion. There exist multiple ways to do multimodal sensor fusion for HAR. Previous studies suggest feature concatenation of the sensor data [10, 27, 48], an ensemble of models [36, 79] for multimodal sensor fusion and propose frameworks for training data from multiple inertial sensors.

But the existing approaches fail to take into consideration the heterogeneity of the devices. Stisen et al. [66] explain that all devices are different and vary in their hardware and operating systems.

They can have a different number of sensors, different sampling frequencies and be located at different locations on the body. Research by Amft [4] and Blunck et al. [7] show that when these HAR systems are deployed in real life, i.e., across heterogeneous devices and usage situations, recognition performances are often significantly lower than what is suggested in research. Furthermore, the existing methods for HAR assume that all the sensing devices exist at all times to make the inference and but in reality, the set of devices carried by a person tends to vary over time even if they own multiple devices.

Unlike the existing solutions, we need a machine learning framework to exploit the information from the available sensors at any given time to give a constant stream of correct activity predictions. Hence, we propose a solution that makes opportunistic use of the information from available devices while being robust to their variations. We provide a scalable deep learning based framework in which each device learns its own sensor fusion model that maps the raw inertial sensor values to a shared low dimensional latent space which we call the 'SenseHAR'- virtual activity sensor. SenseHAR is robust to the heterogeneity in devices, i.e., it has the same format and similar behavior regardless of the subset of devices, sensor's availability, sampling rate, or a device's location. Since the devices get mapped to the same latent space, we combine information from the available devices by aggregating the SenseHAR values from each device. SenseHAR decouples the hardware from software and helps researchers to train their application specific (e.g., from gesture recognition to activities of daily life) algorithms in a hardware-agnostic way. In other words, the same application model trained on SenseHAR values for daily activity recognition can be used with any device or any combination of devices. Finally, we evaluate our method on the PAMAP2, Opportunity and our collected Multi-Modal Activity Recognition dataset (MMAR) which have data from multiple wearables containing inertial measurement units and located in different parts of the body. We compare the performance and find that our solution to train an application specific model using SenseHAR achieves the state-of-the-art accuracy and f1-score when compared with the end-to-end models trained from scratch on raw sensor data.

To summarize, our main contributions of the paper are as follows:

- We propose a Sensor Fusion model for each device that maps the raw sensor values to a shared low dimensional latent space- SenseHAR, the virtual activity sensor.
- We implement a training framework that enables each device to learn the proposed sensor fusion model and to obtain a robust SenseHAR that is invariant to the device heterogeneity.
- We provide a method to calibrate SenseHAR in a new device or hardware setup with the help of the SenseHAR from the pre-trained device.
- We perform thorough evaluations on PAMAP2, Opportunity and on our collected multimodal dataset (MMAR) to show that the SenseHAR captures maximum information from all the sensors from the different available devices and hence can be used to train models for various HAR applications.

2 BACKGROUND

2.1 Human Activity Recognition

The broad field of study to classify sequences of sensor data into known well-defined movements or actions of a person is called Human Activity Recognition (HAR). Prior research has indicated that smartphones and wearables as the widely used modalities in sensor-based HAR [33, 62] that can infer numerous activities including where they go, when they exercise [44] and how well they sleep [16]. HAR is of great importance because it encourages users to adopt a healthier lifestyle by increasing personal awareness about physical activities and its positive consequences on health. It also has a great significance in designing artificially intelligent human-computer interface [22] for various applications such as sign language [8] and medical rehabilitation [9]. While accelerometer is the most prominent sensor used in activity detection [6, 63, 69], gyroscope and magnetometer [3, 41] are also integrated to improve the probability of correct predictions and to identify activities such as the mode of locomotion [27, 34, 55, 70], posture [78], gait [33], fall detection [72], exercise [61] and sleep pattern [46]. Sensor data from smartwatches are utilized to infer different gestures such as finger movements [71], brushing habits [2], smoking habits [51], eating [20, 62] and cooking [81].

2.2 Multimodal Sensor Fusion

In HAR, to ensure the consistent detection of activities and also to recognize fine gestures multiple devices with different types of sensors are usually employed and they are called multi-modal data sources. To exploit the advantages of the multiple sensors, it is necessary to fuse the data delivered by these sensors. The fused data should contain information from all the sensors and ensure a higher certainty about the detection of the activities performed. Sensor fusion is an existing concept and is widely applied in the field of computer vision (merging video with audio signals) [47, 74], in modern cars [24, 26], medical EEG devices [1, 30] and robotics to detect the existence of objects and free space [49, 60], which is used by mobile robots for obstacle detection or path planning. The fusion of these sensor data, therefore, plays a very crucial aspect of the perception process in almost all automated, assistance and autonomous systems. Since this fusion is based on a time series of data from several different sensing modalities, it is called multi-modal sensor fusion. Multi-modal sensor fusion offers advantages in terms of being able to sense various complementary aspects of an object with the different modalities, i.e. it increases the information gained by the system.

2.2.1 Traditional Machine Learning. In the traditional method, each channel or axis of each sensor is considered as an individual sequence. Then, for each input sequence, the list of features in both time and frequency domain are extracted from the raw data. The frequency domain features and Time domain features are explained in detail in the prior works [18, 21]. These features are just taken together and classical machine learning models like SVM, Random forests and XGBoost or sequence-based models like Hidden Markov models and Conditional Random Fields [5, 11, 40, 68] are applied to make the inferences on the activities.

2.2.2 Deep Learning. With the advent of deep learning more sophisticated algorithms emerged to make predictions from multiple sensor streams. S.Yao et al. [77] suggest a deep learning framework by exploiting local interactions within each sensing modality. They apply Fourier transform to obtain better local frequency patterns, then merge local interactions of different sensing modalities into global interactions and extract temporal relationships. But recent research has shown that we don't have to extract specific features from the raw sensor values anymore as the deep neural networks are capable of learning that by themselves. There have been works using CNNs [80], LSTM [15] and hybrid models like Convolutions LSTMS [34] to make the activity predictions. The existing solutions to perform deep learning based multimodal sensor fusion are as follows:

Feature Concatenation: This is a straightforward technique that combines multimodal information in the feature level. Data obtained from different modality data are firstly concatenated into a long feature vector. For example, Bulling et al. [10] combined the features of accelerometers and gyroscopes. Similarly, combining accelerometers with physiological sensors [76], microphone [43], or location sensors [52] is also investigated to improve HAR performance. A few drawbacks of this approach is that these methods have the tendency to neglect the intra-sensor correlations as stated in [54]. Sometimes it's not possible to perform concatenation at feature level because of the difference in the sampling rates of the modalities and the general procedure to mitigate this is to resample the data to the required frequency.

An ensemble of classifiers: This is another commonly adopted method in multimodal activity models [36, 79]. Ensemble classifiers make the final prediction by combining the predictions from multiple base models and the common techniques used are bagging, boosting and stacking. Prior research generally confirms that a diverse set of models increases predictive accuracy in comparison to single models combination [23, 50]. However, a fundamental weakness of EC as stated in [54] is that because fusion takes place at the end of the model, a lot of potential information and cross-sensor relationships are already lost.

Modality-Specific Architectures: Recent work by Radu et al. [54] proposes Modality-Specific Architecture (MA) in Deep Learning which comprises of two types of hidden layers – hidden layers related to a specific sensor type and hidden layers that capture unified concepts across the sensor. Separate architectures are built for each modality to first learn sensor-specific information before their generated concepts are unified through representations that bridge across all the sensors (i.e., shared modality representations) later in the network.

2.3 Heterogeneity in devices

Set of sensors: All smartphones and wearables do not have the same set of inertial sensors. The inertial measurement unit can have only an accelerometer (3 DOF) or accelerometer and gyroscope (6 DOF), or accelerometer, gyroscope, and magnetometer (9 DOF). Wearables usually are equipped with only an accelerometer and don't have gyroscope and magnetometer to reduce energy consumption.

Sampling frequency: The sampling rates of the inertial sensors in different devices are different. Smartphones can collect inertial sensor readings in the frequency range of 50 - 200 Hz. On the other hand, wearables like smartwatches and health bands collect inertial sensor readings at a much lower frequency of 10-100Hz as low frequency helps in increasing their battery life.

Location of devices: A person might own multiple smart devices and they are usually located in different locations on the body. (E.g., Smartphones in the pockets, smartwatches in the wrists and smart shoes in the feet.)

3 SENSEHAR: VIRTUAL ACTIVITY SENSOR

3.1 Overview

We introduce SenseHAR, a robust virtual activity sensor for wearables and smartphones and Figure 1 shows the overview of SenseHAR. The inertial sensor data from the available heterogeneous devices are fused using a sensor fusion network to a shared low dimensional latent space. This shared latent space is called SenseHAR- a virtual activity sensor. Fusing information from different devices to a shared latent space combines the advantages of both feature concatenation and ensemble classifiers as it captures both inter-device and intra-device correlations. SenseHAR is robust to variations in hardware configurations such as the set of sensors, sampling frequency, device location, and device combinations. Therefore, application-specific machine learning models for activity recognition can be trained on the SenseHAR values in a hardware agnostic manner. The same application model trained on SenseHAR values for daily activity recognition can be used with any device or any combination of devices. This enables to infer activities continuously even while the set of devices carried by the person varies over time, with negligible to no loss in performance.

3.2 Requirements

An Ideal virtual activity sensor should have the following requirements:

- Similar format and behavior regardless of the number of available devices.
- Generate values at a constant frequency irrespective of the sampling rates of the individual devices.
- Provide all the required information for activity detection so that it can be used to train models for different applications like identifying gestures and detecting daily activities.
- Have only a few channels. In our paper, the virtual activity sensor has three channels similar to the axes in the IMU sensors and the reason for restricting to only three dimensions is explained in section 5.2 Capture the correlation across different sensors and at different locations

3.3 Training framework

A naive way to construct SenseHAR would be to use convolutional layers on the sensor data from each device to extract features in the latent space but this would not result in a shared common latent space for all devices and would be specific to each device. Instead, we propose a branched, parallel training architecture to obtain robust SenseHAR. Figure 2 shows the training framework. Let 'n'

Figure 1: SenseHar Overview: Data from the inertial sensors of heterogeneous devices are fused together to construct a shared latent space (SenseHAR). Machine Learning models for diverse applications are trained using SenseHAR in a hardware agnostic way.

be the total number of devices and $\mathcal{D}$ refers to the set of available devices $\mathcal{D} = \{D_i\}, i \in \{1, \dots, n\}$. Let 's' be the number of sensors in each device and it can have the values $\{1, 2, 3\}$. Eg. The value of 's' is '1' if the IMU has only an accelerometer, '2' if accelerometer and gyroscope or '3' if accelerometer, gyroscope, and magnetometer. $\mathcal{F}$ refers to the set of Sensor fusion models for each device. $\mathcal{F} = \{F_i\}, i \in \{1, \dots, n\}$. The first part of the training framework is the sensor fusion network which consists of 'n' Sensor Fusion models corresponding to each device. Let $\mathcal{B}$ be the set of branches in the training framework that follows the sensor fusion network. There are a total of 'n+1' branches $\mathcal{B} = \{b_i\}, i \in \{1, \dots, n+1\}$, one for each device and one branch for the combination of all devices. Each branch is mapped to the common shared latent space—SenseHAR. The final 'n+1'th branch aggregates the output of the sensor fusion model from all the individual devices by taking the arithmetic mean (μ). This helps in constraining the values of SenseHAR to lie within the same latent space while combining the information from multiple devices. All the branches share the same application model 'M-shared'. The models are trained using the activity labels by minimizing the sum of the cross-entropy loss of all the branches. The loss function is given in Equation 1 where 'm' is the total training samples, 'b' the number of branches, 'c' is the total number of output classes, 'y' is the true label and 'a' is the activation of the output layer. This method of training constructs a shared latent space between the Sensor Network and Application model, which contains the high-level features from the sensor streams of all devices making it robust and invariant to heterogeneity.

$$Loss(L) = \frac{1}{m} \sum_{i=1}^m \sum_{b=1}^{n+1} \left(\log \sum_{j=1}^c e^{a_{bj}(x)} - a_{by(i)}(x^{(i)}) \right) \quad (1)$$

3.4 Sensor fusion network

The sensor fusion network comprises the sensor fusion model of each individual device as shown in Figure 2. This network is responsible for mapping the raw sensor streams to the shared low dimensional latent space. Each sensor fusion model consists of three stages. Figures 3, 4, 5 show the three stages of the sensor fusion model.

3.4.1 Stage 1. Each sensor stream is considered separately and passed through a series of one-dimensional convolutional layers (1-D Conv). The 1-D convolutional layer is used for extracting local features from 1D patches in every sensor sequence and can identify local patterns within the window of convolution. And because the same transformation is applied on every patch identified by the window, a pattern learned at one position can also be recognized at a different position, making 1D convolution layers translation invariant. Figure 3 shows the data-flow in stage 1. The input to the first stage is the data from available sensors and is of the shape $[(s*3) \times t]$ where '3' denotes the 3 axes for each sensor and 's' is the number of sensors in each device. The output of the first stage is a collection of feature maps corresponding to each filter for every sequence and is of the shape $[(s*3) \times t \times k_1]$ where 'k₁' is the number of 1-D kernels.

3.4.2 Stage 2. The second stage is of the sensor fusion network is to capture the correlation across the corresponding axes in different sensors. So we concatenate the feature maps to the shape $[3 \times t \times (s * k_1)]$ and feed it to the second set of convolutional layers. The output of the second convolutional layer is of the shape $[3 \times t \times k_2]$ where 'k₂' is the number of convolution kernels in the 2nd stage. Figure 4 shows the data-flow in stage 2.

3.4.3 Stage 3. The initial layer of this stage is a K-Max pooling layer [35]. K-max pooling over a linear sequence of values returns a subsequence of k maximum values in the sequence, instead of the single maximum value and it helps to sample down different length vectors into a fixed length. The output from the pooling layer is fed to a Time-Distributed Dense layer which applies the same Dense (fully-connected) operation to every time-step of a tensor, i.e., it allows you to apply that Dense function across every input over time. Thus the output of this layer is a sequence of the same length as the input sequence. In our paper, we use a Time-distributed Dense layer with 3 neurons to capture the correlation across all the feature maps and finally we get a sequence of values of constant dimension $[3 \times t]$. The reason for choosing 3 can be understood from section 5.2 and Figure 5. Intuitively it corresponds to the 3 virtual axes of the virtual sensor.

Figure 2: Branched Training Framework to construct SenseHAR: It has (a) Sensor Fusion Network comprising 'n' Sensor fusion models which maps to the shared latent space 'SenseHAR' (b) Shared Application model to predict activities.

Figure 3: Sensor Fusion Model - Stage 1: Contains 1-D Conv layer to extract features from each sensor stream

Figure 4: Sensor Fusion Model - Stage 2: Captures the correlation across the corresponding axes in different sensors.

3.5 Application model

The second part of the training framework is the application model (M-Shared). This application model is shared by all the branches in the training framework. It includes two stacked LSTM layers followed by an output dense layer to give the activity predictions. The LSTM layers capture the temporal information from the SenseHAR sequence. Dense layer has 'c' neurons corresponding to the number of output classes. And since this is a classification problem we use softmax (σ) as the activation of the final dense layer which is given

Figure 5: Sensor Fusion Model- Stage 3: Fuses the extracted features from the previous stages to a low-dimensional latent space

in Equation 2. Softmax function is used to impart probabilities to the logits 'a' when we have multiple classes and we get the probability distribution of output classes. We consider the most probable occurrence with respect to other outputs as the predicted class. Figure 6 shows the application model architecture.

$$\sigma(\mathbf{a})_j = \frac{e^{a_j}}{\sum_{k=1}^c e^{a_k}} \quad (2)$$

4 EVALUATION METHODS

4.1 Constructing SenseHAR

Inertial sensor data is collected from a person wearing multiple devices while performing various daily activities. We then use the training framework discussed in section 3.3 to train the Sensor fusion models for each device and the Application model by training with the collected data. With the trained sensor fusion models each device can map to the shared low dimensional latent space and generate SenseHAR values.

Figure 6: Application Model: LSTM layers to capture the temporal information and output Dense layer

4.2 Testing on Subset of devices

Though a person might own multiple devices, the set of devices carried by them varies over time. Example: Alice might own a smartphone, a Fitbit, a smartwatch and smart shoes. While going for a run she might have all of them (a smartphone in the pocket, a smartwatch on her left hand, Fitbit on her right hand and smart shoes on her feet), while she's heading to her office she might have only her smartphone and smartwatch and while reading she might just be having one of these devices. So, when Alice has only one device at a given time, we directly use the pre-trained Application model on the data from that device's SenseHAR to make inferences but when she has multiple devices, we take the arithmetic mean of the SenseHAR values from each device to combine their information and use the same pre-trained Application model on the aggregated values to infer the activities.

4.3 Calibrating for new Hardware

When a person buys a new device, it usually has different device hardware specifications (e.g. Different sensor sampling rate or missing the magnetometer sensor). The new device has to be calibrated such that it generates similar SenseHAR values as the existing device. To achieve that, we first need to obtain a new sensor fusion model specific for that device which has different inertial sensor modalities than that were included in the training phase. We do this by following these two steps:

4.3.1 Synchronous Data Collection Phase. We combine our new device or hardware setup with an existing device whose sensor fusion model is already available and synchronously collect data as shown in Figure 7. This provides us the raw sensor values from the new hardware and the SenseHAR values from the existing device which will be used as the ground truth soft labels.

4.3.2 Calibration Phase. We then initialize a new sensor fusion model for the new device which is trained to generate SenseHAR values that resemble the values obtained from the existing device by minimizing the mean squared error (regression) as shown in Figure 8. Once we obtain the sensor fusion model for the new device, we combine it with the same pre-trained Application model to infer the activities. This method eliminates the need to collect a large amount of labeled data for that particular hardware setup making it easier than training a new end-to-end model from the ground up.

Figure 7: Collecting raw data from new device and SenseHAR values from pre-trained device synchronously.

Figure 8: Calibrating the new sensor fusion model by training it with the SenseHAR values of the pre-trained device.

4.4 Training for different applications

For a given device with a trained sensor fusion model, we first collect labeled SenseHAR data for the new application and then train a new Application model for that application (Eg. gesture recognition) using the SenseHAR values and new activity labels. This method while providing similar performance as training from scratch a model on the raw sensor data has two additional benefits—1. Saves time as compared to training a model from scratch since we are only training a few layers. 2. The application model can be designed in a hardware-agnostic way since SenseHAR has the same configuration on all devices.

5 RESULTS

5.1 Datasets

There are plenty of publicly available datasets for human activity recognition and we selected the PAMAP2 and Opportunity datasets because they had the most number of inertial sensor modalities. We also collected our own multimodal dataset to perform more concrete evaluations. The location of the inertial sensors in the three datasets is shown in Figure 9. A short description of each dataset used in this paper is as follows:

5.1.1 PAMAP2 [56, 57]. This dataset had 9 subjects and each subject had to follow a particular protocol that contained 12 different activities. There were three inertial measurement units used in this study and they were placed in three different locations on the body:

Devices	Sampling Frequency	Sensors	Activities
Smartphone (Pocket)	100Hz	A,G,M	Sitting idle
Smartwatch (Right wrist)	50Hz	A,G	Sitting(using computer)
Wristband (Left wrist)	25Hz	A,G	Standing
			Walking
			Running

Table 1: Description of our collected Multimodal Activity Recognition (MMAR) dataset.

wrist, ankle, and chest. The IMUs contained accelerometer, gyroscope, and magnetometer and the data were collected at a sampling frequency of 100 Hz. In our evaluations, we used subject four as our test data and the remaining subjects were used as training data.

5.1.2 Opportunity dataset [58]. This dataset had four trials and each trial had data from so many different sensors including body sensors, object sensors, and ambient sensors. Since we focus only on the inertial sensor data we consider the five inertial body-worn sensors which were worn in five different locations on the body: Left lower arm (LLA), Left upper arm (LUA), Right lower arm (RLA), Right Upper arm (RUA) and Back of the torso. These inertial units recorded accelerometer, gyroscope and magnetometer values. The activities performed had two different hierarchy of labels– High-level activities which considered the 4 major locomotion activities(sit, stand, walk, lie, random) and Low-level activities which had 17 different micro activities such as opening and closing doors, shelves, drawers and drinking tea. In our evaluation, we removed the activities with null labels and considered trial 1 to 3 as training data and trial 4 as test data.

5.1.3 Our Collected dataset - Multimodal Activity Recognition dataset (MMAR). We prepared a multimodal dataset using three commonly available devices– a MotionsenseHRV wrist band worn in our left wrist that collects accelerometer and gyroscope data at a sampling frequency of 25Hz, an Android smartwatch (Asus Zenwatch 2) on right wrist that collects accelerometer and gyroscope data at 50Hz and an Android smartphone (Pixel 3) in the right pocket which records data from accelerometer, gyroscope and magnetometer at 100Hz. We collected data for the following five activities: sitting idle, sitting and using a computer, standing, walking and running. The total duration of our collected dataset was for two and a half hours, with 30 minutes approx. for each activity. The data was split into two sets such that 70% was considered for training and 30% for testing. Table 1 summarizes MMAR dataset.

5.2 Number of SenseHAR Channels

The number of channels for SenseHAR was a hyper-parameter and it depends on the number of neurons in the Time-Distributed dense layer of the sensor fusion model as discussed in section 3.4.3. We wanted to find the smallest number of channels that can capture information and the correlation across different sensor modalities without the loss in performance of the model. This is because the number of channels affect the model size and hence having fewer channels results in fewer parameters and a simpler model. So to find the smallest number of channels for the virtual activity sensor, we created different models having the SenseHAR channels ranging

Figure 9: Placement of Inertial sensors in the different datasets. PAMAP2 dataset has 3, Opportunity has 5, and our collected dataset (MMAR) has 3 inertial sensors placed at different locations on the body.

Figure 10: Number of SenseHAR channels vs F1-score for PAMAP2 and Opportunity datasets. There was not a significant gain in performance when the SenseHAR dimensions increased beyond three.

from 1 to 10 based on our sensor fusion model architecture. We evaluated this on Opportunity and PAMAP2 datasets and plotted the f1-scores achieved vs the number of channels, to obtain the graph shown in Figure 10. From our analysis, we found that increasing the number of channels beyond three did not have a significant increase in the performance of the model for both the datasets and hence we decided to use three channels to represent SenseHAR.

5.3 Baseline models

To evaluate the performance of our proposed method using SenseHAR we used the following existing deep learning based multimodal architectures as our baseline models.

Feature Concatenation methods: The sensing modalities are concatenated to a single modality vector and we considered the three common deep learning architectures– Dense Neural Network(DNN), Convolutional Neural Network(CNN) and a hybrid Convolutional LSTM neural network.

Modality Specific Architecture: We implemented the two modality specific architectures MA-DNN and MA-CNN as described by Radu et al. in [54] which achieved state of the art performance on

multi-modal datasets. These architectures have a dedicated DNN and CNN for each sensing modality whose output is then combined through a fully connected layer to identify the target class.

Evaluation Metrics We used the comparison of classification accuracy and mean F1-Score with the baseline models as our evaluation metrics. Classification accuracy is the number of correct predictions made as a ratio of all predictions made and the F1-score can be interpreted as a weighted average of precision and recall. The F1 scores can be calculated using the equation 3.

$$F1 - score = 2 * \frac{precision * recall}{precision + recall} \quad (3)$$

Table 2 compares the multimodal activity recognition performance of the different baseline models for the three datasets.

5.4 Training to obtain SenseHAR values

First, we obtain the SenseHAR values and the Application model using all the devices and sensor modalities by training them in the branched manner as explained in Section 3.3 for each dataset. We modified the K-max-pool layer of the sensor fusion model such that the sensor fusion model outputs SenseHAR values at a constant frequency of 25Hz (the smallest sampling frequency in the considered datasets). The training framework has one branch for every device and one branch for the combination of devices as shown in Figure 2 and all branches use the same Application model. Therefore, the framework for PAMAP2 dataset and MMAR dataset has 4 branches and Opportunity dataset has 6 branches.

After the training process, we get a sensor fusion model for every device that maps to SenseHAR and an Application model that works on the SenseHAR values. We obtain the accuracy and F1-scores on the test data and report the performance obtained by each branch in Table 3. Since the final branch in our framework corresponds to the Sensor fusion of all the available modalities we compare it with the performance of the baseline models shown in Table 2 and observe that our model achieves the state of the art performance of 95.32%, 74.22% and 93.13% on PAMAP2, Opportunity(gestures) and MMAR datasets respectively.

5.5 Different Combination of devices

We consider different scenarios where only a subset of the devices is available in each dataset. In these cases, we first obtain the SenseHAR values by using the sensor fusion model of the available devices. Then we take the arithmetic mean of the SenseHAR values from the available devices to combine their information and feed it to the trained Application model to get the activity predictions. We also train from scratch the baseline models for this different combination of devices to compare with our approach. Table 4 shows that our method of averaging the SenseHAR values to combine information and using the same Application model to make predictions performs equally well as training a new model from scratch for different combinations. This is of great significance because it eliminates the need to train a new model for every possible permutation of devices.

Figure 11: Performance comparison of the baseline models and SenseHAR-M (Model trained on SenseHAR values) for activity recognition in a new device with different sensors across datasets.

5.6 A new device with Different Sensors

In this case, we consider that the user has a new device with only a subset of the sensors. In our experiments, the new devices are– 1. Chest sensor without gyroscope from PAMAP2 dataset, 2. RUA sensor without gyroscope and magnetometer from Opportunity dataset, 3. A smartphone without gyroscope and magnetometer from MMAR dataset. As seen from the table 3, these devices were not considered in the training phase. So now we initialize a new Sensor fusion model for each of the new devices which can take in the data from available sensors as input and train it using the SenseHAR values from a pre-trained device using the method explained in section 4.3. The Sensor Fusion model of the new devices learn to map to the same low dimensional latent space and the SenseHAR values obtained from this model is combined with the same previously trained Application model. We evaluate the performance on the test data and compare this performance to the performance of the baseline models (i.e.) the model that is trained from scratch on the available sensors and using the activity labels. We observe that even without using the true activity labels but training only on the SenseHAR values, this method is able to come very close to the performance of a baseline model trained on the true labels. There is a small performance drop of less than 1% as shown in Figure 11.

5.7 A new device with a Different Sampling frequency

In this evaluation, we considered only the PAMAP2 and MMAR datasets since these datasets had frequencies higher than 25Hz. For this experiment, we considered two new devices from PAMAP2 dataset– 1. Wrist sensor that collects data at 50 Hz, 2. Ankle sensor that collects data at 25Hz and one new device from the MMAR dataset– 3. Smartwatch on the left wrist at 25Hz. The data for this experiment was obtained by downsampling the original data to from initial frequency to 50Hz and 25Hz respectively. We create a new sensor fusion model for each of these new devices such that they are able to support the new frequencies. This is done by adjusting the max-pooling layer of the sensor fusion model such that it maintains the same SenseHAR frequency of 25Hz. Then the

Datasets	DNN		CNN		Conv-LSTM		MA-DNN		MA-CNN	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
PAMAP2	85.82	0.8622	90.40	0.9036	94.68	0.9446	92.16	0.9204	95.14	0.9499
Opportunity (Gestures)	67.21	0.5842	71.13	0.6431	74.80	0.6792	72.86	0.6684	74.40	0.6754
Opportunity (Locomotion)	85.67	0.8494	87.25	0.8778	89.56	0.8968	88.37	0.8841	90.52	0.9044
MMAR	88.23	0.8831	91.14	0.9107	93.20	0.9298	90.86	0.9082	93.28	0.9306

Table 2: Performance comparison of the baseline models for activity recognition in PAMAP2, Opportunity and MMAR datasets.

Dataset	Branch	Sensors	Sampling Rate	Device Location	Accuracy	F1-score
PAMAP2	1	AGM	100	Wrist (W)	83.35	0.8347
	2	AGM	100	Ankle (A)	86.81	0.8664
	3	AGM	100	Chest (C)	87.24	0.8736
	4	AGM	100	W, A, C	95.32	0.9508
Opportunity (Gestures)	1	AGM	30	Back	65.14	0.5994
	2	AGM	30	RUA	67.32	0.6088
	3	AGM	30	RLA	63.81	0.5845
	4	AGM	30	LUA	66.74	0.6077
	5	AGM	30	LLA	62.05	0.5763
	6	AGM	30	Back, RUA, RLA, LUA, LLA	74.22	0.6748
MMAR	1	AGM	100	Pocket (P)	86.88	0.8658
	2	AG	50	Left Wrist (LW)	89.51	0.8955
	3	AG	25	Right Wrist (RW)	87.39	0.8747
	4	AGM	100,50,25	P, LW, RW	93.13	0.9308

Table 3: Performance of each branch from SenseHAR training framework for activity recognition in PAMAP2, Opportunity and MMAR datasets. The application model trained on the combination of SenseHAR values from all devices is able to achieve the state-of-the-art performance

Dataset	Device Combination	F1-Scores					
		DNN	CNN	Conv-LSTM	MA-DNN	MA-CNN	SenseHAR-M
PAMAP2	Ankle, Chest	0.8433	0.8923	0.9218	0.8809	0.9220	0.9226
	Ankle, Wrist	0.8478	0.8735	0.8982	0.8677	0.8996	0.8968
	Wrist, Chest	0.8465	0.8931	0.9008	0.8845	0.9016	0.8992
Opportunity (Gestures)	RUA+LUA+Back	0.5776	0.6323	0.6583	0.6230	0.6568	0.6550
	RLA+LLA	0.5446	0.5952	0.6204	0.6018	0.6198	0.6210
	RUA+LLA	0.5672	0.6338	0.6392	0.6222	0.6420	0.6404
MMAR	Smartphone +Smartwatch	0.8632	0.8955	0.9120	0.8851	0.9202	0.9208
	Smartphone +Wristband	0.8688	0.8845	0.9095	0.8739	0.9106	0.9092
	Wristband +Smartwatch	0.8646	0.8811	0.9152	0.8812	0.9158	0.9175

Table 4: Performance comparison of the baseline models and SenseHAR-M (Model trained on SenseHAR values) for activity recognition considering different combination of devices in PAMAP2, Opportunity and MMAR datasets.

Figure 12: Performance comparison of the baseline models and SenseHAR-M (Model trained on SenseHAR values) for activity recognition in a new device with different sampling frequency across datasets.

sensor fusion models are trained in the same way as mentioned in section 4.3, combined with the Application model and evaluated on the test user data. We compare the performance with the baseline models trained from scratch on these new frequencies using the true activity labels. We find that our method is able to achieve similar performance as the baseline models though we trained it without using true activity labels. The performance metrics are shown in Figure 12.

5.8 A new hardware at a Different Location

To evaluate different locations, we used the Opportunity dataset since they had five different modalities based on locations (i.e) they had inertial sensors on five different locations on the body as seen in Figure 9. Instead of using all the five devices in the training framework as mentioned in section 5.4 we now use only four devices by leaving out one device. The left out device is then introduced as the new device in a different location. The experiment was done five times with the new device being in a different location each turn. A new sensor fusion model was initialized and calibrated for the new device. Figure 13 shows the performance of the new device when introduced in each of the five locations. We observe that, introducing an inertial sensor in a totally new location, calibrating a new sensor fusion model using the SenseHAR values from existing devices and combining it with the pre-trained Application model has an average performance drop of 5.2% and the worst performance drop of 6% (for the new device in 'Back' location) when compared to training a new baseline models from scratch using true activity labels. The drop in performance can be attributed to the bias in SenseHAR to the locations included in the training framework and can be mitigated by considering more devices located at different locations while training to obtain SenseHAR.

5.9 Different Activities

Usually, we initialize and train a new model on the sensor data for every new application and this is an extremely time and resource consuming process. So to avoid always training a model from scratch, we use the same Sensor fusion model for a particular

Figure 13: Performance comparison of the baseline models and SenseHAR-M (Model trained on SenseHAR values) for activity recognition in a new device at a new location in Opportunity dataset.

device to obtain the SenseHAR values and then train an Application model which is a simple LSTM network on the SenseHAR values for the specific application. Training this simple LSTM model takes significantly less time than training a new model on raw sensor data since the SenseHAR already captures the high-level features corresponding to the activities. To test the generalizability of the SenseHAR values, we did it in two ways: (i) Train an Application model for a different application using the same SenseHAR values (ii) Test the performance across datasets with similar hardware setup.

Different Application: Opportunity dataset has two different labels for each sensor stream- one for low-level gestures and another for higher level locomotion activity. In the training framework, while mapping to the shared latent space, we used the labels from low-level gestures and hence the Application model obtained will correspond to inferring the gestures. Now to infer the modes of locomotion, we train a new Application model on the SenseHAR data by using the same sensor fusion models. Table 5 shows the comparison of the performance metrics obtained by training the baseline models from the start for a different application and our proposed method.

Across Datasets: Data collection of MMAR dataset was similar to the PAMAP2 dataset which also had a wrist sensor worn on the participant's right wrist and hence, we obtained a Sensor fusion model from the PAMAP2 dataset for the wrist sensor sampled at 50Hz. We then used that Sensor fusion model on MMAR dataset to obtain the SenseHAR data for the smartwatch on the right wrist and trained a new Application model on the SenseHAR data for MMAR dataset. The performance of the new Application model was compared with baseline models from scratch on the training data. The performance comparison is shown in Table 5. We found that the F1-score achieved by the Application model trained on SenseHAR values for each activity was similar to that of the baseline model. This shows that SenseHAR is able to capture the information of these activities performed by the participant. This is of importance because we are able to achieve similar performance as state of the art models in a hardware-agnostic way.

Dataset	Activites	F1-score	
		SenseHAR Model	Baseline
Opportunity (Locomotion)	Stand	0.9114	0.9048
	Sit	0.8346	0.8107
	Walk	0.9330	0.9600
	Lie	0.9466	0.9589
MMAR	Sit(idle)	0.9021	0.8992
	Sit(use Computer)	0.9045	0.9088
	Stand	0.9412	0.9488
	Walk	0.9227	0.9210
	Run	0.9574	0.9586

Table 5: Comparing individual activity recognition performance between the Application model trained on SenseHAR values and baseline model for Opportunity and MMAR datasets.

Figure 14: Comparison of time taken to train baseline models and to calibrate SenseHAR on a new device.

5.10 Training Time

We used GTX 1060 GPU to train our models. The initial training time to obtain the robust shared latent space representation is longer when compared to training an end-to-end model since the training framework involves optimization of the sum of the loss functions from multiple branches. It took 80 minutes for PAMAP2 and 50 minutes for Opportunity and MMAR datasets to obtain SenseHAR. But once we obtain the SenseHAR, as discussed in section 3 we do not have to train a new end-to-end model for different combination of devices since we can use the same application model and this saves time considerably. Additionally, the time taken to calibrate a new device to map to the same latent space is three to four times shorter than training a new end-to-end model as only a few convolutional layers in the sensor fusion model have to be trained. Figure 14 compares the time taken to calibrate the SenseHAR with the time taken to train the baseline models for a new device. It's important to note that we don't have to collect labeled data for calibrating SenseHAR which further saves the time and effort that would be required to label the data for every new device.

6 RELATED WORK

In deep learning, before we use a neural network for a task (classification, regression), the usual architecture is to extract features through many layers (convolutional, recurrent, pooling, etc.). These layers map the inputs to the latent space on which the last classification layer is applied. Latent space has gained a lot of importance in Natural Language Processing(NLP) for word embeddings [25], [53], in social network analysis for random graph models [29] and in computer vision for applications like style transfer [82], data compression [28] and data interpolation [73]. The work by Ngiam et al. [47] shows how to learn a shared representation between modalities and evaluate it on a unique task, where the classifier is trained with audio-only data but tested with video-only data and vice-versa.

For cross-domain activity recognition, the concept of transfer learning where certain parameters can be transferred while training a new model for a different application has been explored [17, 45] but this method still requires true activity labels to train the new model. Chen et al. [14] introduce stratified transfer learning approach that uses pseudo labels from existing sensors instead of true labels. It performs intra-class knowledge transfer between domains iteratively to transform them into the same subspaces. Rokni et al. [59] introduce an autonomous training method where, pseudo labels obtained from existing sensors are concatenated with features from the new sensor and then clustered to improve accuracy. But these approaches only work on the traditional method of extracting features from the raw sensor values. Xing et al. [75] show that the shared latent space representation exists for time series sensory data, and it can help transfer knowledge from ambient edge devices to wearable edge devices and vice versa. Radu et al.'s deep learning based modality specific architecture [54] does sensor fusion in latent space. But this work does not take into consideration the heterogeneities in the devices and sensors and hence, a new end-to-end model has to be trained for each device and application. Also, these models assume that all the modalities are present at any given time which is generally not the case in real life implementations. The work by Khan et al. [37] assumes that the distribution of weights and biases in the convolutional layers remains largely unchanged across different activities, and thus automatically adapts and learns the model across different domains with minimal labeled data. This approach does not address the problem of a new device having a different hardware configuration than the existing device. The conventional procedure to take care of the difference in sampling rates of the devices is to interpolate or resample the collected data to the required frequency. The two common methods of interpolation used for HAR are linear [13] and cubic spline [67]. This means additional preprocessing has to be done before feeding it to the HAR models. Stisen et al. [66] propose a clustering-based approach as a mitigation technique to improve HAR performance in the presence of heterogeneities but their work is limited to devices with a single modality(accelerometer).

One of the early efforts in creating a virtual sensor from sensor fusion was the work on virtual gyroscope [12] which proposed the theory of gyro-free IMU. The virtual gyroscope with multigyro-scope and accelerometer array (MGAA) configuration merges the outputs of multi-gyroscopes and specifically placed accelerometers through a Kalman filter. The work by LiKamWa et al. [42] proposed

Moodscope, a virtual mood sensor for a smartphone that measures an important mental state of the user based on the interactions with the smartphone. These papers relied on traditional signal processing and decision-based algorithms and hence do not generalize well for different devices and different users.

7 DISCUSSION & FUTURE WORK

Training Process: For each dataset, the sensor sequences were divided into window size of five seconds with one-second overlap. When the number of sensor modalities changes, we change the input shape of the first convolutional layer in the Sensor fusion model to handle this difference. When the sensors had different sampling frequencies, we adjusted the K-max-pool layers to support the different frequencies. In our implementation, the sensor fusion model had one or two max-pool layers with a pool size of two if the input sensor data was sampled at 50Hz or 100Hz respectively. When the sampling frequency was 30 Hz, we adjusted the K-max-pool layer such that it returns the maximum five values from a pool size of six. Thus when high frequency sensor values are fed into the sensor fusion model, the output, which is the SenseHAR values, will always have a constant frequency of 25Hz. Though the frequency of the SenseHAR is less than the input raw sensor frequencies, the relevant high-frequency features from the sensors are captured by the convolutional and pooling layers of the sensor fusion model, and is rich in information. In addition to the convolutional, K-max-pool, LSTM and dense layers, we also used batch normalization [32] and dropout layers [64] in our models. The use of batch normalization reduces the impacts of earlier layers by keeping the mean and variance fixed, which makes the layers independent with each other and helps the model to train faster. We used dropout layers as the method of regularization to prevent over-fitting of our models. In the dropout layer, the neurons are randomly dropped out of the network with a certain probability during training. This makes the other neurons to step in and handle the representation required to make predictions for the missing neurons resulting in multiple independent internal representations being learned by the network. We used Adam optimizer [38], an extension of the classical stochastic gradient descent procedure, which maintains an adaptive, per-parameter learning rate which helps the network to converge faster.

Pre-Processing: Since our method works on raw sensor streams it requires no pre-processing before the training phase. Hence, we don't require domain-specific expertise to do feature engineering. This saves the effort required to do identify the important features and also the time required to compute those features. We used a window size of five seconds as inputs to our model so that we have sufficient information from each sensor to capture the high-level features and map them to a shared latent space.

Limitations and Future work: To make sure that the SenseHAR values are not biased to the applications it is trained on, we need a significantly large dataset with diverse activities. In our experiments, since the maximum sampling frequency of the devices we considered was 100Hz, the performance of SenseHAR was evaluated only on activities that last for a few seconds. We did not consider any minute fine-grained activities that occur in the millisecond scale (lasts for only a few milliseconds) as it requires data

sampled at high frequencies to capture the fine grained information. In the current method, our sensor fusion model does not work for devices having sensors with sampling frequencies less than 25Hz. So in the future, we aim to address this issue by trying different techniques like zero padding to the input sequence or adding de-convolutional layers to the sensor fusion model to support any arbitrary sampling rates. Also, we used the arithmetic mean for combining information because we wanted the shared latent space obtained from our training framework to be independent of the type of device or the device configuration or location and only contain information related to the activities and hence we gave equal weights to all the devices. In future we plan to incorporate techniques like adaptive weighted averaging and explore different normalization techniques at the sensor fusion step, which might help in improving the performance.

8 CONCLUSION

In this paper, we explore the different challenges and heterogeneities in smartphones and wearables for HAR and construct SenseHAR – a virtual activity sensor that is robust to the variations and availability of the inertial sensors using deep learning. SenseHAR enables the inferring of human activities even when the set of available devices varies over time by mapping the information from the available devices to a shared low-dimensional latent space. We explain the method to calibrate the SenseHAR for a new device in the absence of labeled data with the help of an existing calibrated device. Finally, we do a comprehensive evaluation of SenseHAR and compare its performance with the state-of-the-art multimodal architectures for various device combinations and configurations. Our results indicate that SenseHAR generalizes well for a different set of devices, different sampling frequency, different sensors and for different applications. It shows a small degradation in performance when calibrating for a device introduced in a new location on the body which wasn't included in the training framework. This drop in performance can be minimized if we include a wide array of devices located at different locations on the body in the training framework. Our virtual activity sensor - SenseHAR decouples hardware from software and will help application developers to not worry about the hardware specifications of the device while creating activity recognition models.

ACKNOWLEDGMENTS

The research reported in this paper was supported by a summer internship at Arm. The first and the last authors would also like to acknowledge support of their research from the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, from the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-16-3-0001, and from the NIH Center of Excellence for Mobile Sensor Data-to-Knowledge (MD2K) under award 1-U54EB020404-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

- [1] Mohammad ABS Akhonda, Yuri Levin-Schwartz, Suchita Bhinge, Vince D Calhoun, and Tülay Adalı. 2018. Consecutive independence and correlation transform for multimodal fusion: Application to EEG and fMRI data. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2311–2315.
- [2] Sayma Akther, Nazir Saleheen, Shahin Alan Samiei, Vivek Shetty, Emre Ertin, and Santosh Kumar. 2019. mORAL: An mHealth Model for Inferring Oral Hygiene Behaviors in-the-wild Using Wrist-worn Inertial Sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1.
- [3] Kerem Altun and Billur Barshan. 2010. Human activity recognition using inertial/magnetic sensor units. In *International workshop on human behavior understanding*. Springer, 38–51.
- [4] OD Amft. 2010. On the need for quality standards in activity recognition using ubiquitous sensors. In *Workshop in conjunction with pervasive 2010, how to do good research in activity recognition*.
- [5] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*. Springer, 216–223.
- [6] Louis Atallah, Benny Lo, Rachel King, and Guang-Zhong Yang. 2010. Sensor placement for activity detection using wearable accelerometers. In *2010 International Conference on Body Sensor Networks*. IEEE, 24–29.
- [7] Henrik Blunck, Niels Olof Bouvin, Tobias Franke, Kaj Grønbaek, Mikkel B Kjaergaard, Paul Lukowicz, and Markus Wüstenberg. 2013. On heterogeneity in mobile sensing applications aiming at representative data collection. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 1087–1098.
- [8] Helene Brashear, Thad Starner, Paul Lukowicz, and Holger Junker. 2003. Using multiple sensors for mobile sign language recognition. Georgia Institute of Technology.
- [9] Hans-Peter Brückner, Benjamin Krüger, and Holger Blume. 2014. Reliable orientation estimation for mobile motion capturing in medical rehabilitation sessions based on inertial measurement units. *Microelectronics Journal* 45, 12 (2014), 1603–1611.
- [10] Andreas Bulling, Jamie A Ward, and Hans Gellersen. 2012. Multimodal recognition of reading activity in transit using body-worn sensors. *ACM Transactions on Applied Perception (TAP)* 9, 1 (2012), 2.
- [11] Pierluigi Casale, Oriol Pujol, and Petia Radeva. 2011. Human activity recognition from accelerometer data using a wearable device. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 289–296.
- [12] Honglong Chang, Liang Xue, Wei Qin, Guangmin Yuan, and Weizheng Yuan. 2008. An integrated MEMS gyroscope array with higher accuracy output. *Sensors* 8, 4 (2008), 2886–2899.
- [13] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digi-marti, Gerhard Tröster, José del R Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.
- [14] Yiqiang Chen, Jindong Wang, Meiyu Huang, and Han Yu. 2019. Cross-position activity recognition with stratified transfer learning. *Pervasive and Mobile Computing* 57 (2019), 1–13.
- [15] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. 2016. Lstm networks for mobile human activity recognition. In *2016 International Conference on Artificial Intelligence: Technologies and Applications*. Atlantis Press.
- [16] Zhenyu Chen, Mu Lin, Fanglin Chen, Nicholas D Lane, Giuseppe Cardone, Rui Wang, Tianxing Li, Yiqiang Chen, Tanzeem Choudhury, and Andrew T Campbell. 2013. Unobtrusive sleep monitoring using smartphones. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Æ 145–152.
- [17] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems* 36, 3 (2013), 537–556.
- [18] Walteneus Dargie. 2009. Analysis of time and frequency domain features of accelerometer measurements. In *2009 Proceedings of 18th International Conference on Computer Communications and Networks*. IEEE, 1–6.
- [19] Deloitte. 2018. Global mobile consumer survey: US edition. <https://www2.deloitte.com/us/en/pages/technology-media-and-telecommunications/articles/global-mobile-consumer-survey-us-edition.html>. [Online; accessed 09-April-2019].
- [20] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. 2014. Detecting periods of eating during free-living by tracking wrist motion. *IEEE journal of biomedical and health informatics* 18, 4 (2014), 1253–1260.
- [21] Ç Berke Erdaş, İslim Atasoy, Koray Açıcı, and Hasan Oğul. 2016. Integrating features for accelerometer-based activity recognition. *Procedia Computer Science* 98 (2016), 522–527.
- [22] Elisabetta Farella, Augusto Pieracci, Luca Benini, Laura Rocchi, and Andrea Acquaviva. 2008. Interfacing human and computer with wireless body area sensor networks: the WiMoCA solution. *Multimedia Tools and Applications* 38, 3 (2008), 337–363.
- [23] Steven Finlay. 2011. Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research* 210, 2 (2011), 368–378.
- [24] Daniel Göhring, Miao Wang, Michael Schnürmacher, and Tinosch Ganjineh. 2011. Radar/lidar sensor fusion for car-following on highways. In *The 5th International Conference on Automation, Robotics and Applications*. IEEE, 407–412.
- [25] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
- [26] Jeff Hecht. 2018. Lidar for self-driving cars. *Optics and Photonics News* 29, 1 (2018), 26–33.
- [27] Samuli Hemminki, Petteri Nurmi, and Sasu Tarkoma. 2013. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM conference on embedded networked sensor systems*. ACM, 13.
- [28] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.
- [29] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. 2002. Latent space approaches to social network analysis. *Journal of the American Statistical Association* 97, 460 (2002), 1090–1098.
- [30] Xue-Qin Huo, Wei-Long Zheng, and Bao-Liang Lu. 2016. Driving fatigue detection with fusion of EEG and forehead EOG. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 897–904.
- [31] IDC. 2018. New Wearables Forecast from IDC Shows Smartwatches Continuing Their Ascendancy While Wristbands Face Flat Growth. <https://www.idc.com/getdoc.jsp?containerId=prUS44000018fdeep>. [Online; accessed 09-April-2019].
- [32] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- [33] Takuya Isho, Hideyuki Tashiro, and Shigeru Usuda. 2015. Accelerometry-based gait characteristics evaluated using a smartphone and their association with fall risk in people with chronic stroke. *Journal of stroke and cerebrovascular diseases* 24, 6 (2015), 1305–1311.
- [34] Jeya Vikranth Jeyakumar, Eun Sun Lee, Zhengxu Xia, Sandeep Singh Sandha, Nathan Tausik, and Mani Srivastava. 2018. Deep convolutional bidirectional LSTM based transportation mode recognition. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1606–1615.
- [35] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).
- [36] Ashish Kapoor and Rosalind W Picard. 2005. Multimodal affect recognition in learning environments. In *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM, 677–682.
- [37] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. 2018. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–9.
- [38] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [39] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. 2010. A survey of mobile phone sensing. *IEEE Communications magazine* 48, 9 (2010), 140–150.
- [40] Young-Seol Lee and Sung-Bae Cho. 2011. Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer. In *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 460–467.
- [41] Qiang Li, John A Stankovic, Mark A Hanson, Adam T Barth, John Lach, Gang Zhou, et al. 2009. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information.. In *BSN*, Vol. 9. 138–143.
- [42] Robert LiKamWa, Yunxin Liu, Nicholas D. Lane, and Lin Zhong. 2013. MoodScope: Building a Mood Sensor from Smartphone Usage Patterns. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA, 389–402. <https://doi.org/10.1145/2462456.2464449>
- [43] Paul Lukowicz, Jamie A Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. 2004. Recognizing workshop activity using body worn microphones and accelerometers. In *International conference on pervasive computing*. Springer, 18–32.
- [44] Fen Miao, Yi He, Jinlei Liu, Ye Li, and Idowu Ayoola. 2015. Identifying typical physical activity on smartphone with varying positions and orientations. *Biomedical engineering online* 14, 1 (2015), 32.
- [45] Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*. ACM, 92–99.
- [46] Vincenzo Natale, Maciek Drejak, Alex Erbacci, Lorenzo Tonetti, Marco Fabbri, and Monica Martoni. 2012. Monitoring sleep with a smartphone accelerometer. *Sleep and Biological Rhythms* 10, 4 (2012), 287–292.

- [47] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 689–696.
- [48] Trung Thanh Ngo, Yasushi Makihara, Hajime Nagahara, Yasuhiro Mukaigawa, and Yasushi Yagi. 2015. Similar gait action recognition using an inertial sensor. *Pattern Recognition* 48, 4 (2015), 1289–1301.
- [49] Domen Novak and Robert Riener. 2015. A survey of sensor fusion methods in wearable robotics. *Robotics and Autonomous Systems* 73 (2015), 155–170.
- [50] Giuseppe Paleologo, André Elisseeff, and Gianluca Antonini. 2010. Subagging for credit scoring models. *European Journal of Operational Research* 201, 2 (2010), 490–499.
- [51] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. 2014. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 149–161.
- [52] Juha Parkka, Miikka Ermes, Kari Antila, Mark Van Gils, Ari Manttari, and Heikki Nieminen. 2007. Estimating intensity of physical activity: a comparison of wearable accelerometer and gyro sensors and 3 sensor locations. In *2007 29th annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 1511–1514.
- [53] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
- [54] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2018. Multimodal deep learning for activity and context recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 157.
- [55] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)* 6, 2 (2010), 13.
- [56] Attila Reiss and Didier Stricker. 2012. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 40.
- [57] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *Proceedings of the 2012 16th Annual International Symposium on Wearable Computers (ISWC '12)*. IEEE Computer Society, Washington, DC, USA, 108–109. <https://doi.org/10.1109/ISWC.2012.13>
- [58] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczeck, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkel, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 233–240.
- [59] Seyed Ali Rokni and Hassan Ghasemzadeh. 2018. Autonomous training of activity recognition algorithms in mobile sensors: A transfer learning approach in context-invariant views. *IEEE Transactions on Mobile Computing* 17, 8 (2018), 1764–1777.
- [60] Fendy Santoso, Matthew A Garratt, and Sreenatha G Anavatti. 2017. Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology. *IEEE Transactions on Automation Science and Engineering* 14, 1 (2017), 260–275.
- [61] Christian Seeger, Alejandro Buchmann, and Kristof Van Laerhoven. 2011. myHealthAssistant: a phone-based body sensor network that captures the wearer's exercises throughout the day. In *Proceedings of the 6th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 1–7.
- [62] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. 2016. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors* 16, 4 (2016), 426.
- [63] Jørgen Skotte, Mette Korshøj, Jesper Kristiansen, Christiana Hanisch, and Andreas Holtermann. 2014. Detection of physical activity types using triaxial accelerometers. *Journal of physical activity and health* 11, 1 (2014), 76–84.
- [64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [65] Statista. 2018. Global smartphone shipments forecast from 2010 to 2022 (in million units). <https://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/>. [Online; accessed 09-April-2019].
- [66] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 127–140.
- [67] Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. 2007. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Wearable Computers, 2007 11th IEEE International Symposium on*. IEEE, 37–40.
- [68] Douglas L Vail, Manuela M Veloso, and John D Lafferty. 2007. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 235.
- [69] Peter H Veltink, HansB J Bussmann, Wiebe De Vries, WimL J Martens, Rob C Van Lummel, et al. 1996. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering* 4, 4 (1996), 375–385.
- [70] Lin Wang, Hristijan Gjoreskia, Kazuya Murao, Tsuyoshi Okita, and Daniel Roggen. 2018. Summary of the sussex-huawei locomotion-transportation recognition challenge. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, 1521–1530.
- [71] Hongyi Wen, Julian Ramos Rojas, and Anind K Dey. 2016. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3847–3851.
- [72] Waskitho Wibisono, Dedy Nur Arifin, Baskoro Adi Pratomo, Tohari Ahmad, and Royyana M Ijtihadie. 2013. Falls detection and notification system using tri-axial accelerometer and gyroscope sensors of a smartphone. In *2013 Conference on Technologies and Applications of Artificial Intelligence*. IEEE, 382–385.
- [73] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*. 82–90.
- [74] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. 2016. Multi-stream multi-class fusion of deep networks for video classification. In *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 791–800.
- [75] Tianwei Xing, Sandeep Singh Sandha, Bharathan Balaji, Supriyo Chakraborty, and Mani Srivastava. 2018. Enabling Edge Devices that Learn from Each Other: Cross Modal Training for Activity Recognition. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*. ACM, 37–42.
- [76] Sung-Ihk Yang and Sung-Bae Cho. 2008. Recognizing human activities from accelerometer and physiological sensors. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*. IEEE, 100–105.
- [77] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 351–360.
- [78] O Yurur, C-H Liu, and W Moreno. 2013. Unsupervised posture detection by smartphone accelerometer. *Electronics Letters* 49, 8 (2013), 562–564.
- [79] Piero Zappi, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Troster. 2007. Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness. In *2007 3rd international conference on intelligent sensors, sensor networks and information*. IEEE, 281–286.
- [80] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. 2014. Convolutional neural networks for human activity recognition using mobile sensors. In *6th International Conference on Mobile Computing, Applications and Services*. IEEE, 197–205.
- [81] Mi Zhang and Alexander A Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 1036–1043.
- [82] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint* (2017).