Machine Learning Based Physical Activity Extraction for Unannotated Acceleration Data

UNIVERSITY OF TURKU
Department of Computing
Master of Science in Technology Thesis
Artificial Intelligence
May 2021
Tanja Vähämäki

Supervisors:
Antti Airola
Iman Azimi

UNIVERSITY OF TURKU
Department of Computing
TANJA VÄHÄMÄKI: Machine Learning Based Physical Activity Extraction for Unannotated Acceleration Data
Master of Science in Technology Thesis, 87 p.
Artificial Intelligence
May 2021

Sensor based human activity recognition (HAR) is an emerging and challenging research area. The physical activity of people has been associated with many health benefits and even reducing the risk of different diseases. It is possible to collect sensor data related to physical activities of people with wearable devices and embedded sensors, for example in smartphones and smart environments. HAR has been successful in recognizing physical activities with machine learning methods. However, it is a critical challenge to annotate sensor data in HAR. Most existing approaches use supervised machine learning methods which means that true labels need be given to the data when training a machine learning model. Supervised deep learning methods have outperformed traditional machine learning methods in HAR but they require an even more extensive amount of data and true labels.

In this thesis, machine learning methods are used to develop a solution that can recognize physical activity (e.g., walking and sedentary time) from unannotated acceleration data collected using a wearable accelerometer device. It is shown to be beneficial to collect and annotate data from physical activity of only one person. Supervised classifiers can be trained with small, labeled acceleration data and more training data can be obtained in a semi-supervised setting by leveraging knowledge from available unannotated data. The semi-supervised En-Co-Training method is used with the traditional supervised machine learning methods K-nearest Neighbor and Random Forest. Also, intensities of activities are produced by the cut point analysis of the OMGUI software as reference information and used to increase confidence of correctly selecting pseudo-labels that are added to the training data. A new metric is suggested to help to evaluate reliability when no true labels are available. It calculates a fraction of predictions that have a correct intensity out of all the predictions according to the cut point analysis of the OMGUI software.

The reliability of the supervised KNN and RF classifiers reaches 88 % accuracy and the C-index value 0,93, while the accuracy of the K-means clustering remains 72 % when testing the models on labeled acceleration data. The initial supervised classifiers and the classifiers retrained in a semi-supervised setting are tested on unlabeled data collected from 12 people and measured with the new metric. The overall results improve from 96-98 % to 98-99 %. The results with more challenging activities to the initial classifiers, taking a walk improve from 55-81 % to 67-81 % and jogging from 0-95 % to 95-98 %. It is shown that the results of the KNN and RF classifiers consistently increase in the semi-supervised setting when tested on unannotated, real-life data of 12 people.

Keywords: human activity recognition, wearable sensors, acceleration data, machine learning, semi-supervised learning, unlabeled data

Table of Contents

Abbreviations and Acronyms

AE	Autoencoder
ARI	Adjusted randomizing index
C-index	Concordance index
CNN	Convolutional neural network
DBSCAN	Density-based spatial clustering of applications with noise
DNN	Deep neural network
GMM	Gaussian mixture model
GRU	Gated recurrent unit
HAR	Human activity recognition
HIER	Hierarchical agglomerative clustering
KNN	K-nearest neighbor algorithm
LSTM	Long short-term memory
MET	Metabolic equivalent of task
NMI	Normalized mutual information
NN	Neural network
PCA	Principal Component Analysis
RF	Random forest algorithm
RNN	Recurrent neural network
SVM	Support Vector Machine
TCN	Temporal convolutional network

1 Introduction

Sensor-based human activity recognition (HAR) is an emerging and challenging research area. The goal in HAR is to recognize physical activities of people by monitoring their daily lives. It is important to ensure the quality and quantity of physical activity that has been associated with many health benefits like maintaining physical fitness and even reducing the risk of different diseases [5]. The embedded sensors in smartphones, wearable devices and smart environments have made the sensor data stream more accessible, and HAR is used in many real-life applications in areas like health management, smart assistive technologies, and human computer interaction [1].

HAR applications can use the data of wearable devices, such as accelerometers and gyroscopes. The data can be processed by machine learning methods to recognize and analyze physical activities like sitting, walking, and jogging or for example, activities of daily living such as sleeping and doing domestic tasks [2].

The light, non-invasive and low-cost wearable accelerometer devices, such as Axivity accelerometer [11], play a significant role in remote health monitoring [16]. They can continuously and remotely monitor physical activities of the users. The devices collect acceleration values of body movements in three dimensions over time and save the values in X, Y and Z axes in a defined frequency. For example, the accelerometer can be placed on a thigh and the frequency can be set to 100 Hz. In that case, the device collects acceleration values caused by the gravity (9,81 m/s/s) and the movements of the thigh a hundred times per second.

Machine learning methods allow extracting information from data. A model is trained based on data using a machine learning algorithm. Machine learning algorithms can learn from the data (and the corresponding true labels of the data) by minimizing the error and maximizing the likelihood of the predictions being true [6]. A good HAR model learns to predict labels and thus, learns to recognize activities, from the new sensor data of the

wearable device. The model learns to find patterns in the data related to physical activities performed by the people using the wearable device.

The human physical level can be interpreted from the physical, often regular activities, that people perform in their daily lives. For example, the activities can be grouped into sedentary time and light, moderate and vigorous activity. If a machine learning model predicts activities like sleeping and sitting, it can be assumed that these activities correspond to sedentary time or light activity. If jogging is predicted, the activity level of the person has likely been vigorous activity [10].

1.1 Motivation

Various studies in the literature have proposed machine learning methods for HAR applications [1,2,15]. However, it is still an attractive and challenging research topic. The existing approaches mostly use supervised machine learning methods that require annotation, which means that true labels need be given to the data when training a machine learning model. However, the majority of the sensor data has no labels and acquiring annotated sensor data of wearable devices is especially challenging in HAR [1]. It is even more challenging to annotate sensor data for long-term HAR applications.

The objective of this thesis is to build a machine learning solution to recognize physical activity (e.g., walking, and sedentary time) from unannotated acceleration data collected using an Axivity accelerometer positioned on a thigh. The solutions are tested on real-life acceleration data collected from 12 people who were asked to wear an Axivity accelerometer on a thigh for one week. Although no true labels and no ground truth are available, the performance and the reliability of the new model should be evaluated.

The existing HAR solutions are studied to define the current state of the research related to the task of recognizing physical activities from unannotated sensor data. The characteristics and challenges of HAR and the used approaches to recognize physical

activities with different machine learning methods are examined. Approaches that use supervised machine learning with annotated data and unsupervised machine learning with no true labels are studied. Also, semi-supervised learning, that can use both unannotated data and a smaller annotated dataset, is investigated.

1.2 Research questions

This thesis aims to fulfil the following research questions:

RQ1: Can different activity levels be reliably extracted from an accelerometer device with machine learning using only unlabeled acceleration data?

RQ2: Can machine learning models that are trained with new labeled acceleration data from a single person be used to annotate unlabeled acceleration data reliably?

RQ3: How can both unlabeled and new labeled acceleration data be used together when extracting activities from unlabeled acceleration data?

RQ4: How to get information about the performance of the solution without true labels and the ground truth?

1.3 Contributions

In this thesis, the following contributions are made.

- The current state-of-the-art HAR studies are reviewed and discussed.

- Several types of machine learning solutions are developed based on unsupervised, supervised, and semi-supervised approaches to recognize physical activities in unannotated sensor data of the Axivity accelerometer device.

- New acceleration data of one person is gathered and annotated for one week to acquire annotated data for evaluating the performance of the solutions and to study how to benefit from the new annotated data when developing them.

- The solutions are tested on unlabeled, real-life data collected from 12 participants of the study.

1.4 Thesis structure

The rest of the thesis is organized as follows: Chapter 2 describes challenges and commonly used methods and metrics in HAR. Chapter 3 introduces related work in HAR. Chapter 4 describes the solutions that are developed to extract activities from the Axivity accelerometer device. Chapter 5 explains experiments with the new solutions. In Chapter 6, the results are discussed, and in Chapter 7 conclusions of the study are made.

2 Background

2.1 Challenges in HAR

Machine learning methods have been successfully used in HAR in areas like healthcare and wellness [5]. However, the existing approaches in HAR mostly use supervised machine learning methods that require annotated data, while the majority of the sensor data has no labels. The annotation of the ground truth is a critical challenge for HAR and may not always be feasible [2]. The number of sensor data records is usually huge. If the sampling rate is for example 100 Hz, the number of records is 360 000 for an hour. It is time consuming to label the records and difficult to remember the activities performed at a specific time. It is especially challenging to assign a correct label for short periods or at the boundary of consecutive activities [8]. Alternative solutions are to use camera-based methods to monitor individuals' physical activities. However, the methods are privacy-invasive and thus not suitable [13].

Other challenges in HAR are intraclass variability, interclass similarity and class imbalance. The data captured for the same activity from different users of the device may not be similar in nature, for example because of gender or age, and the data related to different activities may be similar, for example for jogging and running. The duration of various activities may differ and cause class imbalance. There are also heterogeneities across the sensing devices and device positioning [2]. In addition, segmenting a continuous data stream and preserving complete activities is difficult. It is challenging to find the precise start and end time of the activities that are not clearly separated by a predefined posture or pause [1].

2.2 Methods used in HAR

2.2.1 Filtering

The sensor data that has been collected with wearable devices is usually preprocessed with filtering methods because the raw sensor data is scattered and noisy. In signal processing, a filter is a device or process that removes unwanted parts of the signal such as random noise or components lying within a certain frequency range [20]. Useful signal for HAR usually lies in low frequencies, while noise and random dithering usually lie in high frequencies [23]. For example, Butterworth low-pass filtering is used to keep the frequencies that are important to recognize human physical activities and to discard higher frequencies [20].

2.2.2 Segmentation

To associate a sensor data stream of wearable devices to physical activities, the sensor data needs to be divided into smaller segments of the signals. Each segment can then be labeled and recognized as one physical activity. The sliding window approach is the most widely used segmentation method in HAR because of simplicity and lack of preprocessing. In this approach, a window with a fixed size and a fixed shift slides over the signal data with no inter-window gaps. There may be an overlap between adjacent windows to handle transitions of activities more accurately [19].

The window lengths from 0,08 seconds to 30 seconds are commonly used in HAR [16]. The size of the window is often considered to be a tradeoff between recognition speed and accuracy where small windows allow a faster activity recognition and large windows are beneficial to recognize complex activities. However, very small window lengths may

be effective in recognizing activities and should be considered especially in cases when speed is prioritized over the best possible accuracy [19].

There are also activity-defined and event-defined window approaches used in HAR, but they require pre-processing of the sensor data and often laboratory settings. For example, in the activity-defined approach activity changes in the sensor data are detected with methods like analyzing variations of the features or asking feedback from users. In the event-defined approach, specific events are located and used for example with gait analysis detecting heel strikes and toe-offs or with external mechanisms like human supervision [19].

2.2.3 Feature extraction

In traditional machine learning in HAR, the features are manually extracted from the segments of the sensor data. They may include statistical features, such as mean, variance and entropy. The features may be extracted in the time domain, where the data is represented with respect to time, or in the frequency domain, where the data has been transformed into values corresponding frequencies using for example fast Fourier transform [21], discrete cosine transform [22] or wavelet transform [23]. The advantage of these features is that they can be derived from the signal easily and have been effective in the HAR systems [1]. However, this is dependent on human knowledge of the domain and restricts extending the models to other domains [2].

The development of deep neural network (DNN) architectures has allowed learning the features directly from the segments of the raw sensor data without the need to extract the features manually [3]. In DNN, there is an input layer, many hidden layers, and an output layer. The input layer receives the input data, the hidden layers extract patterns within the data, and the output layer produces the results. The layers of DNN can progressively extract higher-level features from the raw input data. However, training DNN models require large volumes of labeled data to get reliable results on new data and not to overfit

on the training data. They also need high computational capacity, because they are complex compared to traditional shallow machine learning methods [6].

2.2.4 Machine learning algorithms

Machine learning algorithms that have successfully been used in sensor based HAR are introduced in this chapter. They can be defined as supervised, unsupervised, or semi-supervised methods. In supervised methods, true labels are needed. Unsupervised methods can be applied on unlabeled data. In semi-supervised methods, both unsupervised machine learning with unannotated data, and supervised machine learning with a smaller annotated dataset are used [12]. Semi-supervised methods aim to reduce the need to annotate sensor data and still train models that can make predictions more accurately than unsupervised learning.

Deep learning methods are also machine learning methods and can be used in unsupervised, supervised, and semi-supervised machine learning. Deep learning methods work well on unstructured data and achieve higher accuracy than traditional machine learning methods. However, most deep learning methods used in HAR are supervised methods. They need an extensive amount of data to avoid overfitting and acquiring a large volume of labeled data is a challenge in HAR [6].

2.2.4.1 Supervised machine learning algorithms

In supervised machine learning, true labels of the training data set are available. A supervised machine learning algorithm is applied on the training data to make predictions by minimizing the error between the predicted and true labels. The model learns to find patterns in the training data related to the given labels and in this way learns to predict labels for new data [12].

2.2.4.1.1 K-nearest neighbor

K-nearest neighbor algorithm (KNN) is an instance-based learning algorithm that predicts labels straight from the data instances in the training data, where the labels of the training data instances are known. The idea is that similar data instances should have similar labels and similarity can be determined with a distance between the instances [9].

In KNN the data instances are represented in a multi-dimensional space where each feature extracted from the data illustrates one dimension. The parameter k (the number of neighbors) is chosen. When the model predicts a label for a new data instance, KNN searches k training data instances that are nearest to the new one. The predicted label is based on majority voting between the labels of the found instances. The parameter k tunes the complexity of the model and the distance can be determined by using any distance metric like Euclidean distance [9].

KNN is a simple algorithm to implement, and it can learn complex nonlinear functions. KNN has reached good accuracy in many domains. However, it has computational and memory complexity and irrelevant features may decrease the accuracy of the model because all features contribute equally to distance [9].

2.2.4.1.2 Random forest

Random forest classifier (RF) is an ensemble of decision tree classifiers illustrated in Figure 1. A decision tree is a hierarchical flow chart algorithm. It uses branches of a tree to describe every possible decision based on the attribute values in the training data. The tree is constructed by decision nodes that symbolize the attributes, branches that mean decisions based on the value of the attribute and leaf nodes that are the labels. Every branch of the tree ends up with a leaf node and the leaf node of the selected branch is the predicted label [7].

In RF, multiple decision tree classifiers are trained simultaneously, and each of them independently predicts labels for the data instances. The idea is that combining independent decision trees increases the stability of the model by reducing variance of the results. The model more unlikely predicts a label incorrectly than a single decision tree. An ensemble of weak classifiers results in a strong classifier [7].

The most commonly used parameters for a RF classifier are the number of trees and maximum depth of the trees. The training data is first randomly divided into subsamples. Features are also randomly selected for the selected number of trees. A decision tree is then formulated from each subsample. The prediction of the label for a new data instance is based on majority voting between the decision trees. The idea behind randomly selecting subsamples and features is to reduce the correlation between the decision tree classifiers in the ensemble helping them to predict labels more independently from each other [7].

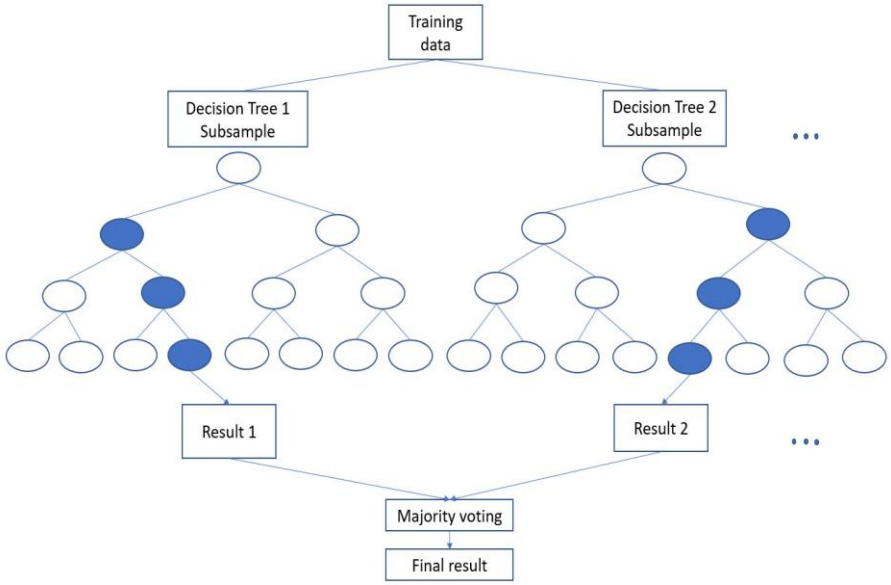


Figure 1 Random Forest classifier

RF works well with nonlinear data and has low risk of overfitting. It has also achieved good accuracy. RF is quite slow to train but it is fast when making predictions [7].

2.2.4.1.3 DNN architectures

In a fully connected DNN, the network consists of fully connected layers: an input layer, many hidden layers, and an output layer. Each successive layer takes the output of the previous layer and feeds the result to the next layer. The result is calculated as a dot product of the input values of the neurons of the layer and the weights that have been calculated to the neurons [12]. Each layer extracts features from the previous layer gradually increasing the abstraction level of the features. The network optimizes the result by iteratively calculating the error of the predictions and recalculating the weights of the neurons with an error backpropagation algorithm [14]. A fully connected DNN is illustrated in Figure 2.

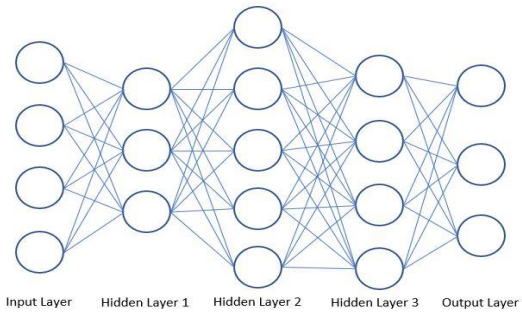


Figure 2 A fully connected neural network with tree hidden layers

A convolutional neural network (CNN) processes a volume of activations rather than vectors and produces feature maps. The activations of the neurons use convolution operations that extract features to the next layer. In a convolution operation, a convolution unit is shifted step by step across the input values using a weight vector (or a filter) resulting in inputs to the units of the next layer [14]. The CNN has also subsampling layers (or maxpooling layers) that reduce the size of the feature maps. The CNNs can model temporal dependencies in the data when gradually extracting more high-level features from the previous layers to the next ones [12].

A temporal convolutional network (TCN) is a CNN developed for sequential data. TCNs use dilated convolutions that can only use present and past inputs like convolutions in

CNNs but can take a sequence of any length in the previous layer and map it to an output sequence of the same length. In this way, an output can represent a wider range of inputs and TCNs can have long effective history sizes [44].

Recurrent neural networks (RNN) can include circles unlike DNNs and CNNs that are feedforward networks. In RNNs, the output depends on both present and past inputs. They can create and process memories of the temporal sequences of the data and mix both sequential and parallel information [14]. The RNN architectures with long short-term memory units (LSTM) or gated recurrent units (GRU) can keep track of internal states that represent the memory of the network. They improve the learning of long time-scale temporal dependences of the sequences and help the system to model more complex patterns [1].

Bi-directional RNNs can be used when both past and future content of the sequences of the data are known in advance. The bi-directional RNN processes the sequences from start to end and from end to start and makes predictions from their combined outputs. The RNNs can also be stacked to create deep RNNs [14].

Attention models have been developed to alleviate RNNs difficulties to learn from long input sequences. They can selectively access the most important parts of the input sequences based on the current contexts instead of accessing the input sequences through fixed size vectors [55].

DNN architectures can learn complex nonlinear functions and have outperformed traditional machine learning methods in accuracy. However, they have significant computational complexity and require large volumes of data for not overfitting when training the models.

2.2.4.2 Unsupervised machine learning algorithms

In unsupervised machine learning, there are no true labels associated with the training data. The aim is to draw inferences from the data and to model the underlying structure and the distribution [12]. It is assumed that certain patterns occur more often than others related to the output values to be predicted [13]. When hidden patterns are found in the groups of the training data, groups of similar physical activities may have been identified [32].

2.2.4.2.1 K-means clustering

Centroid-based K-means clustering aims to identify clusters of similar data instances. The number of clusters must be defined with a parameter k. The centers of the clusters are first randomly initialized and each data instance in the data is pointed to the cluster, the center of which is closest to it. Then new centers of clusters are computed as a mean vector of the assigned data instances. These two steps are repeated until the centers of the clusters do not change anymore. Like with KNN, different distance measures can be used, most commonly the Euclidian distance [9].

K-means clustering is fast, and it has achieved good accuracy in many domains. However, K-means clustering is sensitive to the initial positions of the centers of the clusters, and it may fail if they are badly initialized. Also, the number of clusters has to be pre-specified which may be challenging [9].

2.2.4.2.2 DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm. The idea is that high data density corresponds to clusters. The given parameters are the initial size of the neighborhood area and the number of data instances that should be in the area. The DBSCAN starts with finding an area according to the parameters. The neighborhood area is expanded as long as the density criteria is satisfied. The area forms a cluster that is removed from the data set. These two steps are repeated until suitable areas cannot be found any more [9].

In DBSCAN the number of clusters is not needed. DBSCAN is efficient, and it has also shown good accuracy. It can find clusters of arbitrary shape, but it is not effective when clusters have varying densities [9].

2.2.4.2.3 Hierarchical agglomerative clustering

Hierarchical agglomerative clustering (HIER) builds clusters hierarchically first considering each data instance as a separate cluster. The two clusters that are closest to each other are joined together. This step is repeated until a suitable number of clusters given with a parameter k are formed. Similarity of data clusters can be calculated for example with Euclidean distance between the centroids or mean value vectors of the clusters [9].

In HIER the number of clusters is not needed. The output of HIER is a dendrogram where the hierarchical relationship of the clusters can be visualized. It is possible to choose suitable clusters also merging subclusters [9].

2.2.4.2.4 Gaussian mixture model

A Gaussian mixture model (GMM) is a probabilistic clustering algorithm. GMM optimizes the fit between data and a parametric distribution like a Gaussian or Poisson distribution for each cluster. The data is modeled by a mixture of the distributions. The optimal values for the parameters: a mean, a variance, and a prior probability of the distribution are calculated for each distribution maximizing the likelihood of the data with regards to the model parameters. GMM is a soft clustering method where data instances are not associated only to one cluster, but probabilities of belonging to different clusters are calculated for each data instance [38].

2.2.4.2.5 Principal Component Analysis

In Principal Component Analysis (PCA) dimensionality of data is reduced while trying to retain most of the variation in the data. PCA identifies orthogonal directions called principal components, that maximize the variation of the components. It projects features of data instances to these principal components forming new features that are linear combinations of the original ones. The original features of the data are compressed to fewer features preserving as much variance as possible [9].

PCA is a linear method that is suitable for reducing the number of features and for visualizing data in two or three dimensions [9].

2.2.4.2.6 Deep learning autoencoders

Autoencoders (AE) are an unsupervised technique of neural networks (NN) that can learn compressed knowledge representations of input data. They are a nonlinear generalization of PCA. The task of the AEs is to reconstruct the input data by minimizing the reconstruction error to find structure in the data. First, an encoder encodes the input data

to a latent state representation of the data and a decoder reconstructs the representation back to the input data through the network. The aim is to learn a generalizable way to encode and decode data, not just to memorize the input values [31] .

In a bottleneck AE architecture, hidden layers have fewer nodes than the input layer forming a bottleneck that forces the network to learn compressed latent state representations of the data [31]. The AE with a bottleneck architecture is illustrated in Figure 3.

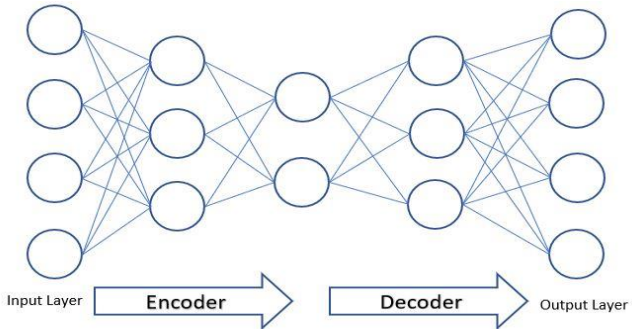


Figure 3 **An autoencoder with a bottleneck architecture**

In a sparse AE architecture, the number of nodes in the hidden layers is not reduced, but only a small number of nodes are activated to learn compressed latent state representations. This is done with a loss function that penalizes activations within hidden layers. Because the activations depend on the input data different input values activate different nodes through the network [31]. The sparse AE architecture is illustrated in Figure 4.

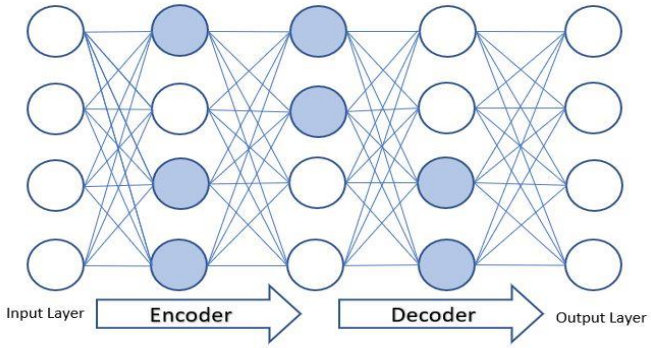


Figure 4 **A sparse autoencoder with a restricted number of nodes activated**

Denoising and contractive AEs aim to learn representations that are robust against noise. In the denoising AE, the input data is slightly corrupted, and the target output is maintained as the original input data. In the contractive AE, a loss function penalizes large derivatives of hidden layer activations with respect to the input data. In this way small changes in the input data maintain similar encoded values and contract a neighborhood of the input values into a smaller neighborhood in the output values [31].

Variational AEs use a probabilistic way to describe values in latent state representations. Instead of giving single values to the attributes in the representation vector, the variational encoder describes a probability distribution for each latent attribute. The encoder builds two output vectors, one describing the mean and the other the variance of the latent state distributions. A vector for the decoder is generated randomly sampling from each latent state distribution. A loss function penalizes the reconstruction error and encourages learning distributions like the true distribution simultaneously. The result is a smooth latent space representation where the outputs are ranges of possible values instead of single values [31].

2.2.4.3 Semi-supervised machine learning methods

In semi-supervised techniques a large amount of unannotated data is used on top of limited annotated data. The idea in semi-supervised learning is that useful information in the unannotated data can be leveraged to learn more effectively from a small set of annotated data [4].

2.2.4.3.1 Self-learning method

Self-learning iteratively uses a supervised machine learning method. A supervised classifier is first trained on a small amount of annotated data, and the classifier is then

used to predict pseudo-labels to some or all the unannotated data. Typically, pseudo-labels are given to the most confident predictions. The data with pseudo-labels can then be used together with the annotated data to retrain the classifier and the self-learning procedure is repeated [41]. The challenge in this approach is that the initial model trained with limited annotated data needs to be good [4].

2.2.4.3.2 Co-learning method

Co-learning follows the procedure of self-learning also simultaneously augmenting the training process with an additional source of information. For example, two separately trained classifiers can teach one another by augmenting each other's training sets with the most confident predictions. The classifiers are retrained, and the process is repeated. In this method, it is assumed that the two separate training sets are sufficient to train the classifiers to make reliable predictions. Also, one classifier's high confidence data instances need to be independent and identically distributed for the other classifier [41].

2.2.4.3.3 En-Co-Training and democratic co-learning methods

En-Co-Training is like self-learning, but consensus of classifiers determines the confidence of the predictions. Confident predictions are added to a common training set and classifiers are retrained on it. En-Co-Training uses majority voting to make the predictions. In democratic co-learning majority voting is used to make predictions and then for example the most confident labeled samples are added to the separate training sets of the classifiers that disagreed with the majority. In En-Co-Training and democratic co-learning the classifiers can be trained on the same data unlike in co-learning. They rely on the difference between the classifiers instead of different feature sets [35].

2.2.4.3.4 Deep semi-supervised methods

Another approach in semi-supervised machine learning is to try to learn class boundaries that are smooth for example with consistency-based methods like denoising AEs. The intuition is that the data should be in the right representation exhibiting clustering, where the classes correspond to the clusters. Because consistency-based methods encourage smooth class boundaries they may not promote clustering that would be needed with very few available labels, though [4].

A ladder network simultaneously trains an AE on unlabeled data and an NN with labeled data. The ladder network consists of a noisy feed forward path (an encoder), a decoder, and a clean feed forward path. The noisy feed forward path and the clean feed forward path share the same mapping function, and the decoder has cost functions on each layer minimizing the difference between the mappings of the noisy and the clean feed forward paths. The output of the noisy feed forward path is also trained with labeled data [47].

Semi-supervised approaches that incorporate pairwise similarity information about different data instances may be used to more explicitly separate classes. For example, Siamese NNs and Triplet networks learn representations from similar/dissimilar pairs [4]. Siamese NNs include dual branches and shared weights between pairs of data instances. They process input pairs and learn pairs of representations, the distances of which can be used to describe the semantic similarity of the pairs [1].

2.2.5 Evaluation metrics

Metrics that are used when evaluating the performance of the solution of this thesis and metrics often used as evaluation metrics of the solutions in HAR are described in this chapter.

2.2.5.1 Metrics used in supervised machine learning

2.2.5.1.1 Accuracy

Accuracy tells the fraction of correct predictions out of all the predictions of the model. It can also be defined with the terms true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) with the equation 1 below.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Accuracy is an often-used metric, but very sensitive to class imbalance [9].

2.2.5.1.2 F1-score

F1-score is a balanced combination of precision and recall and can be calculated with the equation 2.

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \text{ when} \quad (2)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{and} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision indicates the proportion of true predictions among the data instances that have been predicted to belong to the category. Recall that is also called true positive rate or sensitivity defines the proportion of true predictions among the data instances that belong to the category. It shows how well correct categories have been found.

The values of F1-score may vary between 0 and 1. Values close to 1 indicate particularly good precision and recall. F1-score is more robust to class imbalance than accuracy [9].

2.2.5.1.3 Sensitivity

Sensitivity is also called a true positive rate. It shows the fraction of correct positive predictions out of the data instances that belong to the predicted category. It is also called recall and can be calculated with the equation of recall shown above.

2.2.5.1.4 Specificity

Specificity is also called a true negative rate. It tells the fraction of correct negative predictions out of the data instances that do not belong to the predicted category. It can be calculated with the equation 3 [36].

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (3)$$

2.2.5.1.5 Concordance index

Concordance index (C-index) calculates how many times the order of the predictions of pairs were correct out of all possible pairs. C-index is a suitable metric to measure the performance of the model on the data where the labels can be interpreted as an ordinal scale of increasing activity levels. The value 0,5 represents a random prediction and value 1 corresponds to the best model prediction. C-index can be calculated with the equation 4 below [34].

$$\text{C-index} = \frac{\sum_{i,j} 1_{T_j<T_i} \cdot 1_{\eta_j>\eta_i} \cdot \delta_j}{\sum_{i,j} 1_{T_j<T_i} \cdot \delta_j} \quad (4)$$

, where

η_i is the risk score of a unit i

$1_{T_j<T_i} = 1$ if $T_j < T_i$ else 0

$1_{\eta_j>\eta_i} = 1$ if $\eta_j > \eta_i$ else 0

2.2.5.2 Metrics used in unsupervised machine learning

Some of the metrics that are used in unsupervised machine learning can be calculated without access to true labels and the ground truth of the true clusters such as the Silhouette coefficient. However, many of them require true labels making them useless on data that has no labels. For example, to calculate clustering accuracy, the Adjusted Randomizing Index (ARI), or Normalized mutual information (NMI) at least some labels are needed to present the ground truth of the true clusters.

2.2.5.2.1 Silhouette coefficient

The quality of the clustering can be measured for example with the Silhouette coefficient calculated with the following equation 5.

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \text{ where} \quad (5)$$

a(i) is an average distance between i:th data instance and instances in the same cluster and b(i) is an average distance between i:th data instance and instances in the other clusters. In the clusters formed well the data instances are close to the instances in the same cluster and far from those of other clusters. The values of Silhouette coefficient may vary between -1 and 1, values close to 1 meaning particularly good clusters [9].

2.2.5.2.2 Clustering accuracy

Clustering accuracy is a classification accuracy for unsupervised learning. It uses a mapping function to find the best mapping between clusters found by the clustering algorithm and true clusters. This is needed because the algorithm may use different labels from the true labels to represent the same cluster. The clustering algorithm is calculated with equation 6 [39].

$$ACC = max_m \frac{\sum_{i=1}^{n} 1\{y_i = m(c_i)\}}{n}$$, where (6)

m is a mapping function, y is a true cluster, and c is a cluster found by the clustering algorithm.

2.2.5.2.3 Adjusted Randomizing Index

ARI is computed to evaluate similarity between the clusters found by the clustering algorithm and the true clusters given in the annotation. ARI computes the similarity measure between the clusters by considering all pairs of data instances and counting pairs that are assigned in the same or different clusters. It can be calculated with the following equation 7.

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}$$, where (7)

n_{ij} is the number of instances in cluster i formed by the clustering algorithm and the true cluster j, n_i is the number of instances in the cluster i formed by the clustering algorithm, and n_j is the number of instances in the true cluster j. A value close to 0 means random labeling and a value 1, that the clusters are identical [1].

2.2.5.2.4 Normalized mutual information

NMI measures the mutual information between the cluster assignments and the true clusters, and it is normalized by the average of entropy in them. It can be calculated with the equation 8 below.

$$\text{NMI} = \frac{\sum_i \sum_j n_{ij} \log(\frac{n \cdot n_{ij}}{n_i \cdot n_j})}{\sqrt{\sum_i n_i \log \frac{n_i}{n} \sum_j n_j \log \frac{n_j}{n}}}$$, where (8)

n_{ij} is the number of data instances in cluster i formed by the clustering algorithm and the true cluster j, n is the number of instances, n_i is the number of instances in the cluster i

formed by the clustering algorithm and n_j is the number of instances in the true cluster j. A value 0 indicates that the clusters found by the clustering algorithm and the true clusters are totally different, and value 1, that they are similar [1].

3 Related work

3.1 Machine learning approaches in HAR

Works that have successfully used machine learning methods in supervised, unsupervised, or semi-supervised approaches in sensor based HAR are introduced in the following chapters. The aim is to study the state of the research, especially related to the task of recognizing physical activities from unannotated acceleration data collected with the Axivity accelerometer positioned on a thigh. Possibilities to use a small, annotated dataset of one person are also studied. The summaries of supervised, unsupervised, and semi-supervised approaches in HAR with information about used sensors, types of activities to be recognized, applied methods, and used metrics are shown in Tables 1-3 in the end of each corresponding section.

3.1.1 Supervised machine learning in HAR

Traditional supervised machine learning methods that use manual feature extraction have been successful in recognizing human activities from sensor data of wearable devices [15]. However, because of superior performance compared to traditional machine learning there has been a shift towards deep machine learning methods in HAR. CNNs, RNNs and a combination of CNNs and RNNs have been effective in modelling temporal dependencies inherent in sequences captured with sensors of wearable devices [3]. Also, an attention-based framework has been proposed for HAR recently [55].

3.1.1.1 Supervised traditional machine learning approaches

A work [50] investigated decision tables, an instance-based learning method IBL and nearest neighbor, C4.5 decision tree, and Naïve Bayes on datasets annotated by 20 persons to recognize 20 daily activities in real-life situations. The persons were wearing five wire-free bi-axial accelerometers and were asked to perform given tasks outside the laboratory setting. Mean, energy, frequency-domain entropy, and correlation features were extracted from segments of 6,7 seconds with 50 % overlap. The C4.5 decision tree showed the best accuracy (84 %) and nearest neighbor the second-best accuracy (83%). When using only two accelerometers, on thigh and wrist or on hip and wrist, the accuracy decreased only slightly. The accelerometer placed on a thigh was the most powerful in recognizing the activities. It was shown to be possible to recognize daily activities with pre-trained classifiers in real-life situations. Some activities appeared to require user-specific data to be recognized accurately.

In a work [51] the effectiveness of decision tables, C4.5 decision tree, KNN, Support Vector Machine (SVM), and Naïve Bayes as well as meta-level methods boosting, bagging, plurality voting, and stacking was studied on data collected with an accelerometer near the pelvic region from two persons to recognize standing, walking, running, climbing up the stairs, climbing down the stairs, sit-ups, vacuuming, and brushing teeth. The activities were annotated with the help of a stopwatch. Mean, standard deviation, frequency-domain energy, and correlation were extracted from segments of 5,12 seconds with 50 % overlap. Plurality voting turned out to outperform the other classifiers with accuracy (> 90 %).

C4.5 decision tree, KNN, Naïve Bayes, and Bayes Net were compared with accuracy and computational complexity to build an online system to recognize sitting, standing, walking, ascending stairs, descending stairs, and running in a study [49]. Bi-axial acceleration data and light data were collected from six persons, who were wearing a sport watch on various body positions and performing given tasks for 45-50 minutes.

Time domain features: empirical mean Y axis, root mean square, standard deviation, variance, mean absolute deviation, cumulative histogram, n'th percentile, interquartile range, zero crossing rate, mean crossing rate, and squared length of X,Y were extracted from segments of 4 seconds. The C4.5 decision tree was chosen, because it achieved the best balance with accuracy (87 %) and computational complexity. Climbing stairs was difficult to distinguish from walking.

A work [48] compared decision tree methods CART and ID3, an adaptive neuro-fuzzy inference system ANFIS, Nearest Neighbor, KNN, and Naïve Bayes to recognize daily activities lying, standing, jogging, walking, climbing upstairs, and climbing downstairs on acceleration data collected from twenty-eight healthy adults for one hour. Step count, frequency z axis, frequency x axis, mean of maxima x, angle z, RMS of derivative x, energy y, entropy z, entropy x, and area z were the most frequently selected features from segments of 4 seconds. A Java application was used to annotate the data with markers, descriptions, and timestamps. KNN had the best accuracy (> 96 %) on individual datasets and the CART decision tree showed the best accuracy (> 85 %) on group datasets. The sensitivity of climbing stairs was the lowest with all the methods.

In a study [52] SVM, NN, and C4.5 decision tree as well as a model combining them with majority voting were trained on laboratory data and evaluated on data collected in free-living conditions. 52 individuals were wearing a tri-axial accelerometer at the lower back and other accelerometers on various body positions to gather reference information considered as the ground truth, both in a laboratory setting and without supervision. Activities were also annotated in diaries. Mean, standard deviation, kurtosis, skewness, range, cross-axis correlation, accelerometer angle, spectral energy, spectral entropy, peak frequencies, and cross-spectral densities were extracted from the segments of 6,4 seconds. All the models showed good accuracy (> 92 %) on laboratory data but a significant decrease in accuracy (> 72 %) in free-living conditions. Majority voting had the best accuracy (95 % on laboratory data and 75 % in free-living conditions). It was concluded that daily-life data is essential when training and testing classification models in HAR.

A study [28] created the publicly available PAMAP2 dataset from sensor data collected with three inertial measurement units containing two tree-axial accelerometers, a gyroscope, and a magnetometer. They were placed on the chest, the dominant arm, and the dominant ankle. In addition, a heart rate monitor was positioned on the chest. 9 people were performing 12 daily and six optional activities following a protocol. In addition, C4.5 decision tree, boosted C4.5 decision tree, bagging C4.5 decision tree, Naïve Bayes, and KNN were compared on the data. Time and feature domain features were extracted from the acceleration data and mean and gradient from the heart rate data in segments of 5,12 seconds with a shift of 1 second. The boosted C4.5 decision tree and KNN reached the best accuracy and F1-score (both > 99 %).

A work [53] compared KNN, SVM, GMM, and RF to recognize daily activities with accuracy, F1-score, recall, precision, and specificity. Sensor data was collected with accelerometers worn on the chest, right thigh, and left ankle by six persons who were asked to perform 12 daily activities that were annotated by an observer for 30 minutes. Mean, variance, median, interquartile range, skewness, kurtosis, root mean square, zero crossing, peak to peak, crest factor, range, DC component in FFT spectrum, energy spectrum, entropy spectrum, sum of the wavelet coefficients, squared sum of the wavelet coefficients and energy of the wavelet coefficients, the correlation coefficients of mean, and variance of the norm of each acceleration were extracted from the segments of 1 second with 80 % overlap. A wrapper approach based on RF feature selection had been used to select the features. The KNN and RF reached the best performance with all the used metrics F-score, recall, precision, specificity, and accuracy (near 99 %).

3.1.1.2 Supervised deep learning approaches

A generic deep framework based on CNN and RNN was proposed for enhancing recognition accuracy and recognizing increasingly complex physical activities in [43]. The features were automatically extracted from raw sensor data by CNN and temporal dynamics of feature activations were modeled by RNN. Multimodal sensor data could also be fused. The framework was evaluated with the task of recognizing standing,

walking, sitting, and lying down and right-hand gestures in the OPPORTUNITY dataset [54] collected in a sensory-rich environment and 10 different hand gestures in the Skoda dataset [26] collected from assembly-line workers in a car production environment. The framework outperformed the previously published results, also CNN approaches, on the OPPORTUNITY and Skoda datasets with F1-score (between 89 % and 95 %).

A new study [55] suggested the first purely attention-based deep learning framework for HAR. In addition, a personalization framework was proposed to adapt the model to a specific user acquiring data and labels from the user over time. The framework was evaluated on the HHAR [33], PAMAP2 [28], and USC-HAD [56] datasets with F1-score (70 – 84 %) outperforming RF and the previously published deep learning approaches. Personalization increased the F1-scores (74 – 88 %). It was concluded that purely attention-based models are highly capable of extracting temporal dependencies in sensor based HAR.

Table 1 Related work with a supervised machine learning approach

Reference	Sensors	Activities	Methods	Metrics
Ling Bao et al., 2004 [50]	Accelerometers	20 daily activities	C4.5 decision tree, decision tables, instance-based methods IBL and nearest neighbor, Naïve Bayes	Accuracy with C4.5 decision tree 84 %, with nearest neighbor 83 %
Nishkam Ravi et al., 2005 [51]	Accelerometer	lying, standing, jogging, walking, climbing up the stairs, climbing down the stairs	boosting, bagging, plurality voting, and stacking with decision tables, C4.5 decision tree, KNN, SVM, and Naïve Bayes	Accuracy with plurality voting > 90 %
Uwe Maurer et al., 2006 [49]	Accelerometers and light sensors of sport watches	sitting, standing, walking, ascending stairs, descending stairs, running	C4.5 decision tree, KNN, Naïve Bayes, Bayes Net	Accuracy with C4.5 decision tree 87 %

Luciana C. Jatoba et al., 2008 [48]	Accelerometers	lying, standing, jogging, walking, climbing upstairs, climbing downstairs	Decision tree methods CART and ID3, ANFIS, Nearest Neighbor, KNN, Naïve Bayes	Accuracy with CART decision tree 86 %, sensitivity
Illapha Cuba Gyllensten et al., 2011 [52]	Accelerometer	lying down, sitting / standing, dynamic / transitions, walking, running, cycling	SVM, NN, and C4.5 decision tree, majority voting	Accuracy with majority voting 95 % (lab data) / 75 % (free-living data)
Attila Reiss et al., 2012 [28]	Accelerometers, gyroscopes, magnetometers	12 daily activities (PAMAP2)	C4.5 decision tree, boosted C4.5, bagging C4.5, Naïve Bayes, KNN	Accuracy and F1-score with boosted C4.5 decision tree and KNN > 99 %
Attal Ferhat et al., 2015 [53]	Accelerometers	12 daily activities	KNN, SVM, GMM, RF	Accuracy with KNN and RF near 99 %, F1-score, recall, precision, specificity
Francisco Javier Ordóñez et al., 2016 [43]	Accelerometers, gyroscopes, magnetometers	4 locomotion activities and 17 hand gestures in the OPPORTUNITY dataset, 10 hand gestures in the Skoda dataset	Combination of CNN and RNN	F1-score 89 % - 95 %
Davide Buffelli et al., 2020 [55]	Accelerometers, gyroscopes, magnetometers	Activities of the HHAR (6), PAMAP2 (12), and USC-HAD (12) datasets	Attention model	F1-score 70 – 84 %, with personalization 74 – 88 %

3.1.2 Unsupervised machine learning in HAR

Unsupervised methods do not need labeled data to train the model, but they have not been used as much as supervised machine learning methods in HAR. The performance of unsupervised methods has usually been inferior to supervised methods [2]. Research of unsupervised learning in HAR has mostly been conducted in clustering of handcrafted features, in weight initialization in pre-training, and in unsupervised feature learning prior to supervised fine tuning. Some works have been suggested to recognize human activities in an unsupervised manner [3].

DNNs have been used to create clustering-friendly representations and cluster assignments simultaneously for still image data and impressive results have been achieved with unsupervised deep clustering frameworks for computer vision applications. However, they have not been able to exploit the sequential nature of sensor data and learn representations of human activities from raw sensor data of wearable devices [3].

3.1.2.1 Unsupervised traditional approaches

A study [8] investigated DBSCAN, HIER, GMM, and K-means clustering that were applied on means and standard deviations extracted from sensor data of accelerometers and gyroscopes of smartphones. Volunteers were asked to perform five activities common in daily living: walking, running, sitting, standing, and lying down for ten minutes. When the number of clusters was known, GMM showed 100 % accuracy. When the number of clusters was unknown DBSCAN and HIER reached over 90 % clustering accuracy. The Calinski-Harabasz index was used to find an optimal number of clusters to the HIER algorithm.

In addition to supervised machine learning methods, the unsupervised methods K-means clustering, GMM and Hidden Markov Model were compared in [53]. The Hidden Markov Model showed the best performance with F1-score, recall, precision, specificity, and clustering accuracy (near 84 %).

A study [36] suggested a protein interaction model MCODE to recognize human activities. MCODE, GMM, HIER, centroid-based clustering methods K-means++ and K-medoids, and a graph-based Spectral clustering were compared. They were applied on mean, standard deviation, variance, skewness, kurtosis, correlation, and signal magnitude area features that were extracted from segments of 180 seconds with 75 % overlap of acceleration data obtained with smartphones. To evaluate the results two datasets were collected, one from basketball playing and another from race-walking activities. Video was recorded and used to manually annotate the activities. MCODE was shown to outperform the other models with ARI, FM-index, accuracy (74% – 88 %), recall, precision, specificity, and F1-score on the daily living activities collected by WISDM Lab [37] and the two own datasets.

In [57], centroid-based clustering methods K-means, K-mode and CLARANS clustering, a hierarchical BIRCH clustering, and DBSCAN clustering were applied on sensor data from the UCI HAR [25] dataset collected with accelerometers and gyroscopes of smartphones. Features of the time and frequency domain had been extracted from segments of 2,56 seconds with 50 % overlap. K-means and DBSCAN clustering reached the highest clustering accuracy (95 %) also when the number of features was reduced.

3.1.2.2 Unsupervised deep learning approaches

A work [32] proposed a deep learning variational AE model for learning representations of human activities. Relative changes of position and orientation were calculated from sensor data of accelerometers and gyroscopes of wristbands as input to a variational AE consisting of bi-directional LSTMs. The model was evaluated on data collected and

annotated in laboratory-based sessions with 10 persons and the epileptic patients' daily activities of the public HHAR dataset [33]. The supervised classifiers, a decision tree classifier C4.5, KNN and RF, were applied on the embedded mean vector of the variational AE. They outperformed those applied on hand-crafted features with F1-score. The unsupervised model reached a clustering accuracy higher than 87 %.

Recently, the first unsupervised, standalone, end-to-end deep clustering method Deep Sensory Clustering [3] was suggested to recognize human activities straight from raw sensor data of wearable devices. A recurrent AE with bi-directional GRUs and with reconstruction and future prediction objectives, and centroid-based Cluster assignment hardening were jointly used to learn clustering-friendly representations and to generate soft cluster assignments. The approach was compared with K-means clustering, HIER, and end-to-end deep clustering for still images on the public datasets UCI HAR [25], Skoda [26] and MHEALTH [27]. They showed consistent improvement of performance with metrics of clustering accuracy (53 % – 75 %) and NMI.

Unsupervised Embedding Learning for HAR [1] using deep learning AE architecture was also recently suggested for unsupervised clustering in HAR. Mean, variance, standard deviation, median value, largest value, smallest value, and interquartile range features were extracted from raw sensor data as input to AE with objectives to minimize reconstruction, temporal coherence, and locality preserving losses. K-means clustering was then applied on the learned representations to find cluster assignments. The approach was compared with PCA and the traditional AE on the public datasets PAMAP2 [28], REALDISP [29] and SBHAR [30] with metrics of clustering accuracy (71 % – 92 %), ARI and NMI showing improved performance.

Table 2 Related work with an unsupervised machine learning approach

Reference	Sensors	Activities	Methods	Metrics
Yongjin Kwon et al., 2014 [8]	Accelerometers and gyroscopes of smartphones	Walking, running, sitting, standing, lying down	DBSCAN, HIER with Calinski–Harabasz index, K-means clustering, GMM	Clustering accuracy with DBSCAN and HIER > 90 %, NMI

Attal Ferhat et al., 2015 [53]	Accelerometers	12 daily activities	Hidden Markov Model, K-means, GMM	Accuracy with Hidden Markov Model near 84 %, F1-score, recall, precision, specificity
Yonggang Lu et al., 2017 [36]	Accelerometers of smartphones	Basketball playing, race walking, daily activities of the WISDM dataset (6)	MCODE, GMM, HIER, K-means++, K-medoids, Spectral clustering	Clustering accuracy with MCODE 74 – 88 %, ARI, FM-index, recall, precision, specificity, F1-score
Jue Wang et al., 2018 [57]	Accelerometers and gyroscopes of smartphones	6 daily activities of the UCI HAR dataset	K-means , K-mode, CLARANS, BIRCH, DBSCAN	Clustering accuracy with K-means and DBSCAN 95 %
Lu Bai et al., 2019 [32]	Accelerometers and gyroscopes of wristbands	9 daily activities, epileptic patient daily activities (6) of the HHAR dataset	Deep learning variational AE with bi-directional LSTMs	Clustering accuracy > 87 %, F1-score
Alireza Abedin et al.,2020 [3]	Wearable devices	Activities of the UCI HAR (6), Skoda (10), and MHEALTH (12) datasets	End-to-end deep learning RNN AE with bi-directional GRUs and Cluster Assignment Hardening	Clustering accuracy 53 – 75 %, NMI
Sheng Taoran, 2020 [1]	Wearable devices	Daily and sport activities of the PAMAP2 (12), REALDISP (33), and SBHAR (6) datasets	Deep learning AE with temporal coherence and locality preserving loss and K-means clustering	Clustering accuracy 71 – 92 %, ARI, NMI

3.1.3 Semi-supervised machine learning in HAR

Relatively little work has been conducted with semi-supervised machine learning in HAR [4]. Semi-supervised approaches that use hand-crafted features have been applied to reduce the required amount of annotated training data [40]. Most research on semi-supervised learning in HAR has used sequential AEs to learn representations from unlabeled sensor data to improve supervised classification [4].

Although impressive classification performance has been achieved with semi-supervised learning in computer vision using denoising AEs with class-preserving augmentations, semi-supervised learning is challenging in HAR. The data segments in the sequential data should map to the clusters, but the boundaries of the segments are not known. In addition, class-preserving augmentations, such as rotation and mirroring with images, are difficult to define in HAR [4].

3.1.3.1 Semi-supervised traditional machine learning approaches

A study [41] explored self-learning and co-learning with a supervised method joint boosting on the sensor data in the PLCouple1 dataset [42]. The data was collected with accelerometers on the dominant wrist, the dominant hip, and the non-dominant thigh and 10 infra-red sensors. The male's daily activities, actively watching tv or movies, dishwashing, eating, grooming, hygiene, meal preparation, reading paper/book/magazine, using computer, and using phone, had been annotated for 15 days with the help of an audio-visual recording system. Mean, variance, energy, spectral entropy, area under curve, pairwise correlation between the three axes, and the first ten FFT coefficients were extracted from segments of 30 seconds with 50 % overlap from the acceleration data. The number of activations of the infra-red sensors were also

calculated as features. Both self-learning and co-learning improved the accuracy of the classifier. Co-learning using two types of sensors reached the best accuracy (40 % when the number of used labels was 2,5 %) compared to self-learning and supervised training.

In [35] semi-supervised methods self-learning, En-Co-Training, and democratic co-learning were compared to find suitable methods to augment a HAR classifier with new unlabeled data after it had been deployed in a mobile device. The mean, variance, and the FFT coefficients between 1 and 10 Hz were extracted from the segments of one second from acceleration and GPS speed data of smartphones worn by 17 participants staying in one place, walking, and running for 90 minutes. It was shown that En-Co-Training and democratic co-learning performed well when the accuracy of the initial classifier was low, between 75 – 80 %. When the initial accuracy was high, 90 %, the methods did not improve the accuracy of the initial classifiers but did not decrease the accuracy either. Self-learning did not significantly improve the accuracy of the initial classifier. Democratic co-learning was nearly as good as active learning, where a user is asked to label the least confident predictions. It was able to improve the initial accuracy from 84 % to 90 %.

3.1.3.2 Semi-supervised deep learning approaches

A work [40] presented two semi-supervised CNN methods, a denoising CNN AE with a supervised CNN and a convolutional ladder network, for recognizing human activities from both labeled and unlabeled raw sensor data split into segments of 1 second with 50 % overlap. Both models outperformed a supervised CNN classifier pretrained with unlabeled data, self-learning with logical regression, and a pseudo-label method on the public ActiTracker [46], the PAMAP2 [28], and MHEALTH [27] datasets with F1-score (> 75 % when the number of the labels was 1 %). It was shown that adjusting low-level features based on unlabeled data in the CNN AE and the convolutional ladder network improved the high-level features.

A new semi-supervised sequence classification approach [4] through change point detection was suggested to learn representations that incorporate pairwise similarity information about data instances in both unlabeled and labeled sensor data. The segments between the change points were classified similarly and adjacent segments on opposite sides of the change points were classified differently. Similar and dissimilar pairs were fed to TCN resulting means of empirical distributions that were used as representations of the data. The learned representations were shown to outperform the representations learned by a denoising AE in a semi-supervised setting using a DNN classifier. The models were tested on simulated and real datasets the HCI [45] and the WISDM [37] with F1-score (65 % when the number of the labels 3 %). Also, the results were close to the results of training a supervised classifier on the learned representations.

A semi-supervised approach using an AE and a Siamese NN [1] was also recently proposed for HAR. Unsupervised temporal and feature consistency criteria were used through the AE, and weakly supervised label consistency criteria with pairwise constraints was used through the Siamese NN on a mean, variance, standard deviation, median, and interquartile range extracted from raw sensor data. K-means clustering was applied on the learned clustering-friendly representations. The model outperformed the unsupervised Embedding Learning for HAR [1] and the supervised methods RNN with LSTM, CNN, DNN, SVM, C4.5 decision tree, and a boosted C 4.5 using 10 % of the labeled data on the PAMAP2 dataset [28] with a clustering accuracy (99 %). When the number of the labels was 5 % the model reached a clustering accuracy 97 %.

Table 3 Related work with a semi-supervised machine learning approach

Reference	Sensors	Activities	Methods	Metrics
Maja Stikic et al., 2008 [41]	Accelerometers, infra-red sensors	9 daily activities of the PLCouple1 dataset	Self-learning and co-learning with joint boosting	Accuracy 40 % when labels 2,5 %
Brent Longstaff et al., 2010 [35]	Accelerometer and GPS speed of smartphones	Staying in one place, walking, running	Self-learning with C4.5 decision tree, En-Co-Training, and democratic co-learning with C4.5	Accuracy 90 % when initial accuracy 84 %

			decision tree, Naïve Bayes and SVM	
Ming Zeng et al., 2018 [40]	Accelerometers, gyroscopes, magnetometers, temperature, heart rate data, ECG data	Daily and sport activities of the ActiTracker (6), PAMAP2 (12), and MHEALTH (12) datasets	Denoising CNN AE with supervised CNN, Convolutional ladder network	F1-score > 75 % when labels 1 %
Nauman Ahad et al., 2020 [4]	Accelerometers, gyroscopes	Gesture recognition of the HCI dataset (5), daily activities of the WISDM (6) dataset	TCN with Change point detection and DNN	F1-score 65 % when labels 3 %
Sheng Taoran, 2020 [1]	Accelerometers, gyroscopes, magnetometers, temperature, heart rate data, ECG data	The PAMAP2 (12) dataset	AE with Siamese NNs with temporal, feature, and label consistency criteria and K-means clustering	Clustering accuracy 97 % when labels 5 % and 99 % when labels 10 %

3.2 Summary of related work

Although various machine learning approaches have been successfully used in sensor based HAR, most of the works have used supervised machine learning methods that require all the training data to be labeled. For example, traditional machine learning methods such as decision trees used in [48-50], a boosted decision tree and KNN used in the work [28], and KNN and RF in [53] achieved good accuracy and outperformed supervised methods like SVM and Naïve Bayes reaching accuracies over 80 % up to 99 %. Supervised deep learning approaches such as the combination of CNN and RNN [43] and the recently proposed attention based NN [55] outperformed the traditional methods and were able to recognize complex activities more accurately. But the supervised deep

learning approaches need even higher volumes of labeled training data and are not feasible methods in this thesis.

Unsupervised methods can find patterns in unlabeled data and promising results have been achieved with traditional unsupervised approaches such as K-means clustering and DBSCAN in a work [57], and DBSCAN and HIER in [8] with over 90 % accuracy. Also, the recent unsupervised deep learning approaches in works [1,3,32] were able to successfully use deep AE frameworks on sequential sensor data of wearable devices with accuracy up to 92 %. However, the performance of unsupervised methods has been inferior to supervised methods. In addition, also with unsupervised machine learning, at least some labeled data is required to present the ground truth to evaluate the performance of the model.

Some works have proposed semi-supervised machine learning methods using both unsupervised methods on a large amount of unlabeled data and a small, labeled data set in HAR. For example, a work [41] improved the accuracy of the initial classifier with self-learning and co-learning and a study [35] improved the initial classifier with En-Co-Training and democratic co-learning from 84 % to 90 %. Deep AEs have been used to learn representations to improve the performance of a supervised classifier. For example, a denoising CNN AE and a supervised CNN classifier, and a convolutional ladder network were studied in [40] achieving 75 % F1-score. The recent work [1] proposed AE with Siamese NN with temporal, feature, and label consistency criteria followed by K-means clustering. It achieved 97 % accuracy when the number of the labels was 5 % and 99 % accuracy when the number of the labels was 10 % of all the training data.

Unlike in most previous works, there are no available labels related to the data that is used in this thesis to present the ground truth. So, there is no direct way to use supervised or semi-supervised machine learning methods or even to evaluate the performance of the unsupervised methods comparing the results with the true clusters. A new dataset of one user is collected and annotated to be able to evaluate the performance of the unsupervised machine learning methods and also to be able to use supervised and semi-supervised methods when recognizing activities from the original unannotated acceleration data.

Another difference is that the data used has been collected with only one sensor, a tri-axial accelerometer positioned on the thigh of the participants. Based on a work [50] where it was shown that a sensor positioned on a thigh was the most powerful to recognize physical activities, it is assumed that it is possible to recognize basic activities like sleeping, sitting, sitting in a car, walking around, taking a walk, and jogging from the data collected with the Axivity accelerometer positioned on a thigh.

3.3 Open questions in HAR

The challenge of annotating sensor data in HAR and a large amount of continuously streaming unlabeled data has increased the interest in methods that help to reduce the need for labeled data. In semi-supervised machine learning a small, labeled dataset is used together with a large amount of unlabeled data, but also other methods have been studied in HAR to train classifiers with less labeled training data. In active learning, a user is only asked to label the training data instances that the classifier has not been able to classify with high confidence. In transfer learning, on the other hand, a pre-trained classifier can be used and only fine-tuned with a small amount of labeled data that has been collected for example from other persons, by other types of sensors or in a different environment [2].

Another challenge in HAR, intra-class variability between people, but also in a data stream of one person, is also a current research area in HAR. The sensor data of different people typically has variations within the same activities, and sensor data of one person does not stay static over time either. Change of existing activities and also emergence of new activities can be expected. How to adapt a model that has been trained on sensor data of a group of people to be able to better recognize activities of other persons and also from the evolving data stream of the same person is actively studied in HAR. The aim is to personalize a user-independent model to increase its accuracy when recognizing activities from an individual data stream and also adapt it with evolving activities [58].

Using mobile devices with limited resources to recognize human activities has also become an active research area in HAR. Sensors can be embedded in mobile devices like smartphones that either transmit the data and receive the results via the backend server, where the HAR model is applied, or the HAR model is implemented directly on the mobile device. The latter has become a feasible option because of the improved computational power of the devices. In a mobile real-time activity recognition both time and accuracy are key criteria for measuring performance of a HAR model. An interesting possibility is also to aggregate recognized activities from users' devices on a high-level platform like the cloud to be used and studied together with other information for example related to a location. In context aware activity recognition, the aim is also to leverage information from the context of the surrounding environment to recognize higher level and more complex activities more accurately [58].

Incremental and active learning has become a new and promising research area in HAR. In this approach, an initial model is trained on a small amount of labeled data and then the model is continuously accumulated with incremental and active learning only asking labels for informative samples in a continuous data stream [58]. In incremental learning, a model is not retrained with new data, but only incrementally updated to adapt the model to new instances in a data stream. Incremental learning without any user interaction has also been suggested in HAR. In this approach only the predicted labels of the model are used when updating the model. However, this kind of totally autonomous learning can lead to concept drift and incorrect predictions [24].

4 Extracting activities from Axivity accelerometer device

First, to answer the research question RQ1: “Can different activity levels be reliably extracted from an accelerometer device with machine learning using only unlabeled acceleration data?” the unsupervised machine learning algorithm K-means clustering is applied on the unlabeled acceleration data because it has shown good performance in the research of HAR [57]. The aim is to find clusters that would correspond to physical activities to be recognized. To be able to evaluate the reliability of these unsupervised methods new acceleration data is recorded with the Axivity device and annotated. K-means clustering is applied on data containing both unannotated and new, annotated data. The assigned clusters of the annotated data can then be compared with the true labels given in the annotation.

Next, to find an answer to the research question RQ2: “Can machine learning models that are trained with new labeled acceleration data of one person be used to annotate unlabeled acceleration data reliably?” the supervised machine learning algorithms, KNN and RF are applied on the new, annotated acceleration data. The KNN and RF have shown competitive performance compared to other traditional supervised methods in HAR [28, 53]. The aim is to train two separate classifiers that can predict physical activities from unannotated acceleration data.

To answer the research question RQ3: “How can both unlabeled and new labeled acceleration data be used together when extracting activities from unlabeled acceleration data?”, the previously trained supervised classifiers are used with the En-Co-Training method in a semi-supervised setting. The En-Co-Training is used like in [35], but together with two classifiers and making separate predictions by the classifiers instead of majority voting. In addition, the cut point analysis of the OMGUI software [18] is performed. The activity levels produced by the cut point analysis are used as reference information to increase confidence of selecting correct pseudo-labels. The aim is to leverage knowledge

from the unannotated data and to improve the classifiers to better generalize on the data collected from other users of the Axivity device.

Finally, the research question RQ4: "How to get information about the performance of the solution without true labels and the ground truth?" is examined. A new metric is proposed. It calculates a fraction of correctly predicted activity levels out of all the predictions also according to the cut point analysis of the OMGUI software. The new metric is used to get reference information about the reliability of the classifiers to predict activities from unannotated acceleration data.

The Jupyter Notebook IDE, Python version 3.6.8, Scikit Learn Library version 0.20.3 and Scipy Library version 1.2.1 are used when implementing the solution and performing experiments with the data.

4.1 Data

The data of this study has been collected with the Axivity accelerometer device from 12 people, who were asked to wear an Axivity accelerometer on a thigh for one week. The individuals were between 27 and 46 of age. The physical activity rate during the week, age, weight, and height were also asked from them. Table 4 shows the background information of the participants.

Table 4 Participants' background information

Characteristics	**Values**
Age (years), mean (SD)	36,8 (5,4)
BMI, mean (SD)	23,0 (2,5)
Physical activity during the week, n (%)	
Rarely	3 (25)
A few times a week	5 (42)
Almost every day	4 (33)

The acceleration data of the Axivity device is first converted from binary files to CSV files in units of g (=9.81 m/s/s) with the OMGUI software [18]. A CSV file is created from each day of a participant. The data consists of timestamps and acceleration values of X, Y and Z axes that have been recorded in the frequency of 100 Hz. Thus, there are 360 000 recordings per hour and about 9 million recordings per day. The acceleration values of X, Y and Z axes of one user for one day is shown in Figure 5.

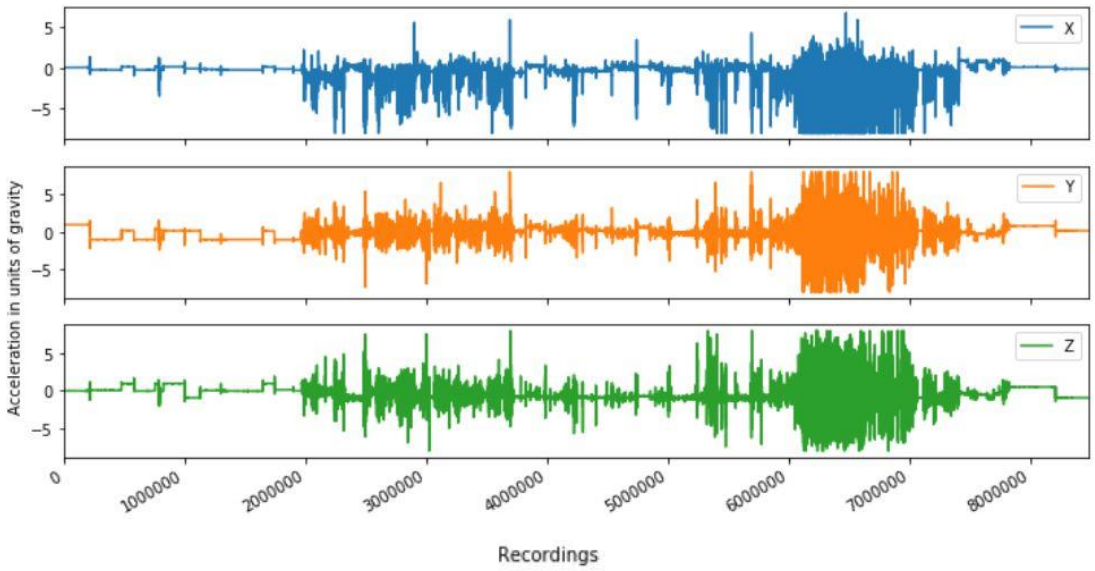


Figure 5 **Sensor data from Axivity accelerometer in X, Y and Z axes for one day**

4.2 New annotated data

To obtain annotated data more acceleration data of one person is recorded and labeled for one week. The true labels are saved in a note application of a mobile phone at a minute level and then converted to Excel files. The aim is to label basic daily activities that could be reliably recognized also with traditional machine learning methods. The activity types should also cover all the activities performed during the week. In addition, it should be easy to compare the activity types with the activity levels later produced by the cut point analysis of the OMGUI software [18].

The activities are annotated using the following labels: 0 = sleeping, 1 = sitting, 2 = sitting in a car, 3 = walking around and doing tasks, 4 = doing workout, 5 = taking a walk, 6 = jogging, 9 = a break in the annotation, 10 = to be automatically annotated that will be used with the unannotated data. The label 4 is combined with the label 3, because the results of both labels seem to be close to each other in the analysis.

The labels can be interpreted as an ordinal scale of increasing activity levels. Sleeping, sitting, or sitting in a car correspond sedentary time or light activity. Walking around and doing tasks can be interpreted as sedentary time, light, or moderate activity. Taking a walk should be light or moderate activity and jogging should be vigorous activity [10]. The activity types and the corresponding activity levels are shown in Table 5.

Table 5 Activity types and corresponding activity levels

Label	Activity type	Activity level
0	Sleeping	Sedentary time / Light activity
1	Sitting	Sedentary time / Light activity
2	Sitting in a car	Sedentary time / Light activity
3	Walking around and doing tasks	Sedentary time / Light activity / Moderate activity
4	Workout (will be combined with the label 3)	Sedentary time / Light activity / Moderate activity
5	Taking a walk	Light activity / Moderate activity
6	Jogging	Vigorous activity
9	A break in the annotation	
10	To be annotated (will be used with the unannotated data)	

4.3 Methodology

The process used in this thesis follows the steps commonly used in the HAR process: 1) data collection 2) preprocessing of sensor data 3) feature extraction and 4) applying

machine learning algorithms. The result is 5) a model that can recognize activities from new sensor data [1]. The HAR process is shown in Figure 6.

Figure 6 Process of human activity recognition

4.3.1 Pre-processing of the acceleration data

4.3.1.1 Segmentation

The acceleration data of X, Y and Z axes, that has been collected with the Axivity device, is split into consecutive segments to separate different activities in the sensor data stream so that each segment can be labeled and recognized as one physical activity. The lengths of the segments of 1, 5 and 10 seconds are tested, and the length is set to 10 seconds. It seems to be a suitable window size for recognizing the previously chosen activity types.

The timestamp of each segment is compared to the timestamp of the annotation data. If the annotation is 9 (= a break in an annotation), the segment is not processed further, but discarded. Otherwise, the segment will be further processed.

4.3.1.2 Butterworth low-pass filtering

The acceleration data of the segments is filtered because the raw sensor data is scattered and noisy. Butterworth low-pass filtering [20] is used to keep the low frequencies that are important to recognize human physical activities and to discard higher frequencies. The order is set to 4 and the cutoff frequency is set to 10 Hz. The order of the Butterworth filtering affects the sharpness of the cutoff. The higher the order is the sharper the cut-off frequencies are.

4.3.2 Feature extraction

4.3.2.1 Time domain features

Features are extracted from each filtered segment of the sensor data because they are more effective for separating different activities than the sensor data. A set of statistical features are first extracted from the segments in the time domain, where the segments are represented with respect to time like in the original sensor data stream. The features, that are extracted from the filtered segments in the time domain, are shown in Table 6.

The following statistical features: mean, median, standard deviation, largest value, smallest value, interquartile range, skewness, kurtosis, and root mean square, are calculated from the filtered acceleration values of each segment from the X, Y and Z axes separately. Also, peak prominences, that measure how much the peaks of the signal stand out from the surrounding baseline, and peak widths in the middle of the peak heights and contours are calculated and summarized from the segments of each axis. Approximate entropy is also calculated to quantify the amount of regularity of fluctuations in the filtered acceleration values of the segments. The smaller the approximate entropy is the more regular the signal is in the segment.

In addition, Pearson correlation coefficients between the axes X and Y, X and Z, and Y and Z are calculated from the segments and signal vector magnitudes are calculated to describe the intensity of the movements from the filtered acceleration values of the X, Y and Z axes from each segment with the equation 9 below.

$$SVM = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (9)$$

Table 6 Features extracted from the segments in the time domain

Axis	Extracted features
X axis	X Mean, X Median, X Standard deviation, X Largest, X Smallest, X Interquartile range, X Skewness, X Kurtosis, X Root Mean Square, X Peak prominences sum, X Peak widths sum, X Approximate entropy
Y axis	Y Mean, Y Median, Y Standard deviation, Y Largest, Y Smallest, Y Interquartile range, Y Skewness, Y Kurtosis, Y Root Mean Square, Y Peak prominences sum, Y Peak widths sum, Y Approximate entropy
Z axis	Z Mean, Z Median, Z Standard deviation, Z Largest, Z Smallest, Z Interquartile range, Z Skewness, Z Kurtosis, X Rot Mean Square, Z Peak prominences sum, Z Peak widths sum, Z Approximate entropy
Several axes	Pearson correlation (X, Y), Pearson correlation (X, Z), Pearson correlation (Y, Z), Signal vector magnitude

4.3.2.2 Frequency domain features

Statistical features are also extracted from the segments in the frequency domain. The filtered acceleration data is transformed from the time domain to the frequency domain to show how much of the signal lies within each given frequency band over a range of frequencies. FFT is used to transform the signal data of the segments, that is represented in respect to time, to the magnitude values of the frequency content of the signal. The features, that are extracted from the segments in the frequency domain, are shown in Table 7.

The following statistical features are calculated from the magnitude values of the frequency content: mean, median, standard deviation, largest value, smallest value, interquartile range, skewness, kurtosis, and root mean square. Power spectral densities, that measure the signal's power content versus frequency, are calculated for the frequencies from 0 to 10 Hz, within frequency bins of 1 Hz, and the dominant power spectral densities are calculated from the segments of the axes. Normalized spectral entropy is calculated to measure the uniformity of the power spectral densities in the segments of the axes. The smaller the normalized spectral entropy is the more uniform the power spectral densities are in the segment.

Table 7 Features extracted from the segments in the frequency domain

Axis	Extracted features in the frequency domain
X axis	X Magnitudes mean, X Magnitudes Median, X Magnitudes Standard deviation, X Magnitudes Largest, X Magnitudes Smallest, X Magnitudes Interquartile range, X Magnitudes Skewness, X Magnitudes Kurtosis, X Magnitudes Root Mean Square, X PSD (Power Spectral Density) 0, X PSD 1, X PSD 2, X PSD 3, X PSD 4, X PSD 5, X PSD 6, X PSD 7, X PSD 8, X PSD 9, X PSD10, X Dominant PSD, X Normalized Spectral entropy
Y axis	Y Magnitudes mean, Y Magnitudes Median, Magnitudes Standard deviation, Y Magnitudes Largest, Y Magnitudes Smallest, Y Magnitudes Interquartile range, Y Magnitudes Skewness, Y Magnitudes Kurtosis, Y Magnitudes Root Mean Square, Y PSD (Power Spectral Density) 0, Y PSD 1, Y PSD 2, Y PSD 3, Y PSD 4, Y PSD 5, Y PSD 6, Y PSD 7, Y PSD 8, Y PSD 9, Y PSD10, Y Dominant PSD, Y Normalized Spectral entropy
Z axis	Z Magnitudes mean, Z Magnitudes Median, Z Magnitudes Standard deviation, Z Magnitudes Largest, Z Magnitudes Smallest, Z Magnitudes Interquartile range, Z Magnitudes Skewness, Z Magnitudes Kurtosis, Z Magnitudes Root Mean Square, Z PSD (Power Spectral Density) 0, Z PSD 1, Z PSD 2, Z PSD 3, Z PSD 4, Z PSD 5, Z PSD 6, Z PSD 7, Z PSD 8, Z PSD 9, Z PSD10, Z Dominant PSD, Z Normalized Spectral entropy

4.3.2.3 Standardization

All the extracted features are standardized with Z-score standardization to change the values of the features to a common scale so that the mean value will be 0 and the standard deviation will be 1 with the equation 10 below. The standardization prevents the features with a larger scale from dominating in machine learning algorithms.

$$z = \frac{x - \mu}{\sigma}$$, where (10)

μ is the mean and σ is the standard deviation.

4.3.3 Applying machine learning algorithms

The unsupervised machine learning algorithm K-means clustering is applied on the time and frequency domain features extracted from the segments of the unannotated acceleration data to study if K-means clustering can find clusters with similar features. The similar features between the segments of the data would suggest that the activity types of the segments could also be the same.

The supervised machine learning methods, KNN and RF, that are suitable to be used on a small amount of data, are applied on the time and frequency domain features extracted from the segments of the new, labeled acceleration data. The aim is to study, if the trained KNN and RF models can be used to reliably predict labels from the original unannotated data collected from the participants of the study.

In addition, the semi-supervised method En-Co-Training is used with the KNN and RF models to leverage knowledge from the unannotated acceleration data and to improve the generalization performance of the models that have only been trained on the labeled data

of one person. The aim is to study if activities can be predicted more reliably from the original unannotated acceleration data using both unlabeled and labeled data in a semi-supervised setting.

4.3.4 Cut point analysis of OMGUI software

The cut point analysis of the OMGUI software [18] is performed to produce activity levels from the unannotated sensor data for reference information. The activity levels of the cut point analysis can be compared to the labels that are predicted by the KNN and RF models, and the comparison can help the human evaluation of the predicted labels without knowing the true labels and the ground truth.

The cut point analysis of the OMGUI software produces the following activity levels: 0 = sedentary time, 1 = light activity, 2 = moderate activity and 3 = vigorous activity based on the approach proposed in [17]. It predicts energy expenditure of a person given in units of a metabolic equivalent of task (MET) based on mean signal vector magnitude values that are extracted from segments of acceleration data. It calculates the signal vector magnitudes also subtracting the gravity 1 m/s/s with the equation 11 below and sets the thresholds between the activity levels to 1,5 MET, 4 MET and 7 MET as suggested in [17]. The activity levels of the cut point analysis are shown in Table 8.

$$SVM = \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \quad (11)$$

Table 8 Activity levels produced by the cut point analysis of OMGUI software

Label	Activity level	Measurement
0	Sedentary time	< 1.5 MET
1	Light activity	>= 1,5 MET, < 4 MET
2	Moderate activity	>= 4 MET, < 7 MET
3	Vigorous activity	>= 7 MET

MET measures the amount of oxygen consumed per kilogram of body weight per minute. 1 MET means that a person consumes approximately 3,5 millilitres of oxygen per kilogram of body weight in a minute, which is roughly equivalent to being at rest. The energy expenditure may differ between persons based on several factors, for example age and fitness level, but thresholds can be set to approximate the difference between different activity levels [10].

In the interface of the cut point analysis tool the predictions are chosen to be made every minute. A fourth-order Butterworth band-pass filtering between 0,5 and 20 Hz is chosen to be used. The position of the device is chosen to be on a hip instead of on a wrist because it better corresponds to the true position on a thigh.

Although the cut point analysis predicts the activity levels of the segments based on a single feature the result of the analysis is still interesting. It is assumed that the predicted activity levels can help to evaluate the reliability of the solution that is implemented in the thesis. The signal vector magnitude is shown to correlate to the intensity of the physical activity or the activity level well [17] and the activity level should relate to the activity types that are predicted by the models [10].

5 Experiments

5.1 Recording and annotating new data

To evaluate the result of the unsupervised method K-means clustering, new acceleration data is recorded with the Axivity device and annotated for one week. The activities are carefully annotated at a minute level, which is the same level as will be used in the cut point analysis of the OMGUI software [18]. The same level of annotation makes it easy to compare the results later.

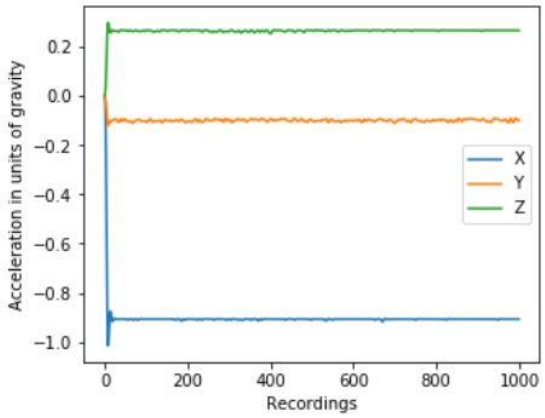
It is first quite challenging to make annotations in practice at a minute level, but a practical way is found with a note application of a mobile phone and a systematic way to annotate the activities. It is sometimes difficult to remember the activities performed every minute, especially for short periods and to recognize, when the activity has changed to another exactly. The most practical way is to use the label 9 (= a break in the annotation) for the time periods, when the annotation has not succeeded for some reason or the device has been taken off for example because of taking a shower.

The true labels are saved in a note application of a mobile phone. The labels are given every time a new activity begins as exactly as possible. No other labels are given to keep the amount of the labels as small as possible. The labels in a note application are then converted to Excel files. The annotations are quality checked comparing them to the corresponding activity levels of the cut point analysis of the OMGUI software to find and correct clear misspellings in the annotation.

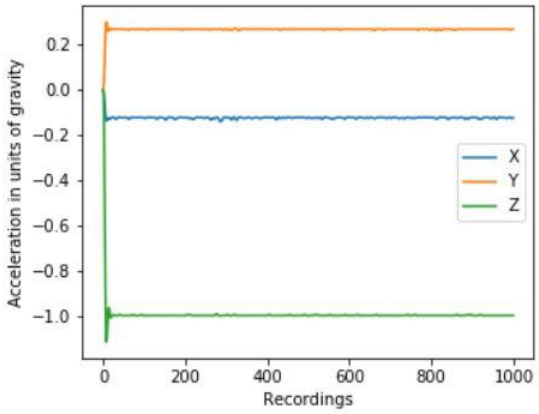
5.2 Analyzing the new annotated data

5.2.1 Visualisation of the segments

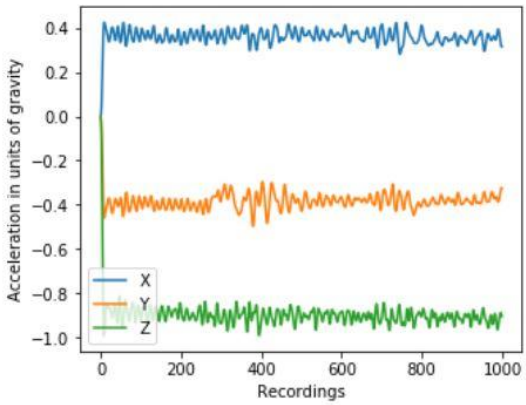
The new acceleration data that has been annotated is first pre-processed with segmenting and filtering, and the features are extracted from the segments of the X, Y and Z axes. Filtered segments of X, Y and Z axes of different activities are first plotted. Also peaks of the filtered segments of the Y axis and the contour heights of the peaks, and magnitude values of the frequency content of Y axis are plotted to visually examine possible differences between the annotated activities. There seem to be clear differences between the segments annotated as different activities. The visualization of each activity type is shown in Figures 7-9.



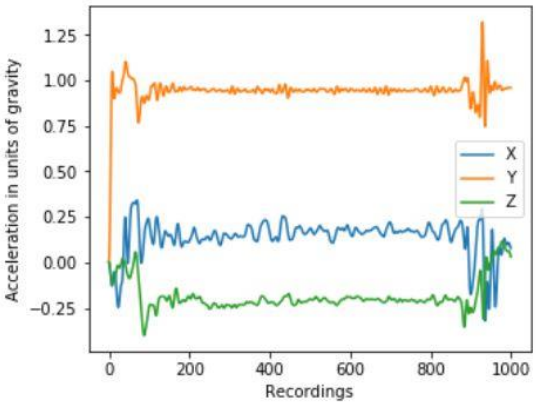
Sleeping



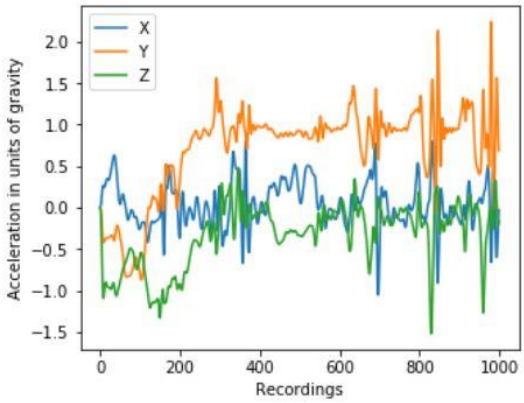
Sitting



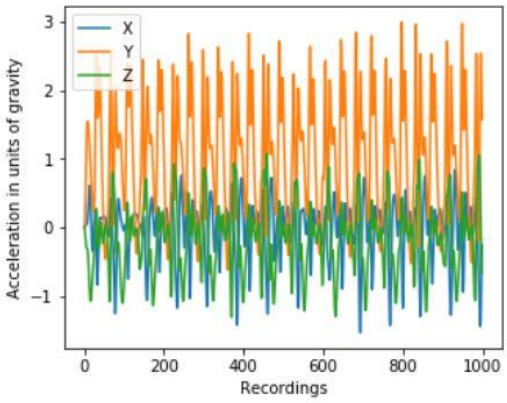
Sitting in a car



Walking around and doing tasks

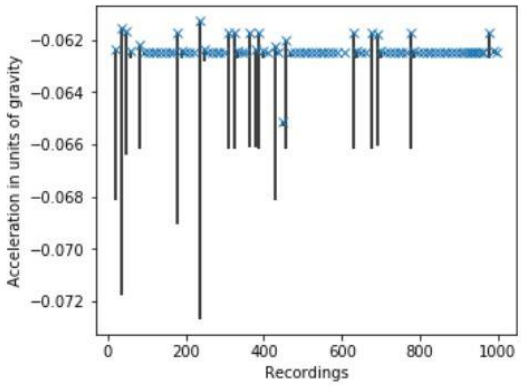


Taking a walk

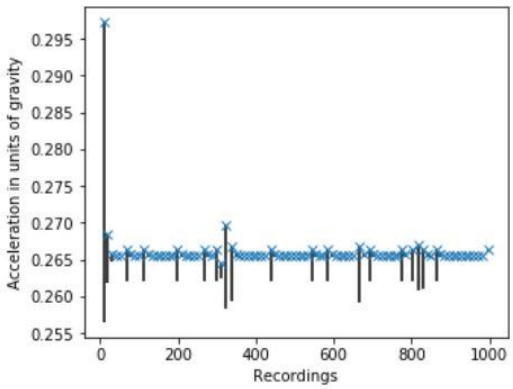


Jogging

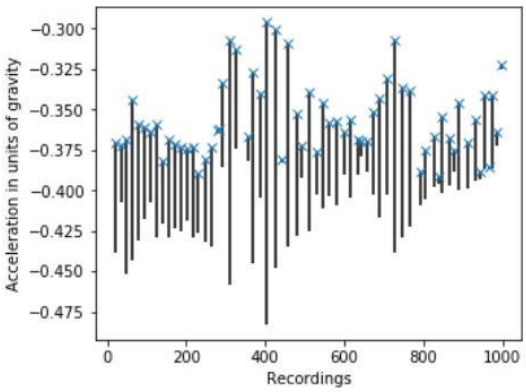
Figure 7 Filtered segments of the X, Y and Z axes



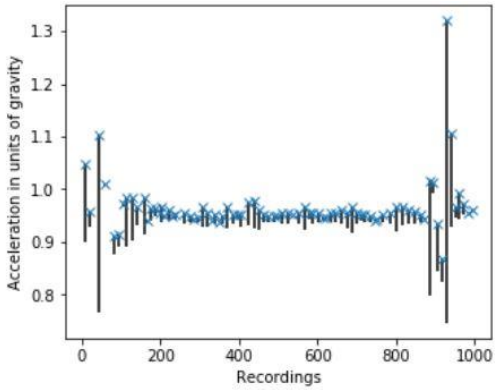
Sleeping



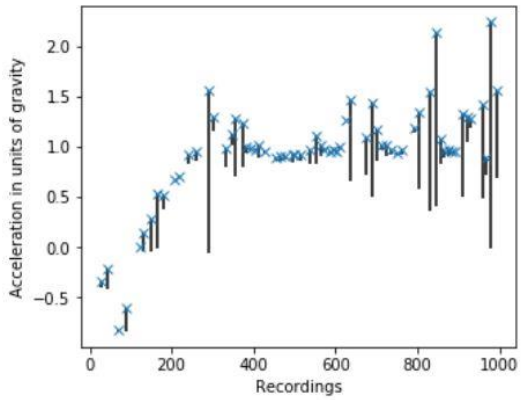
Sitting



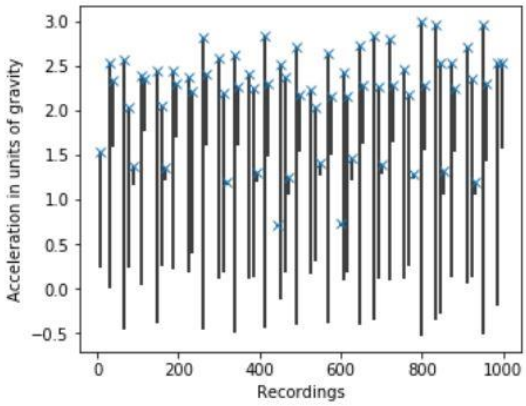
Sitting in a car



Walking around and doing tasks

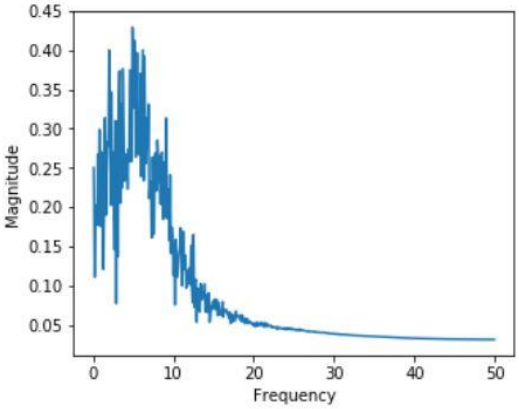


Taking a walk

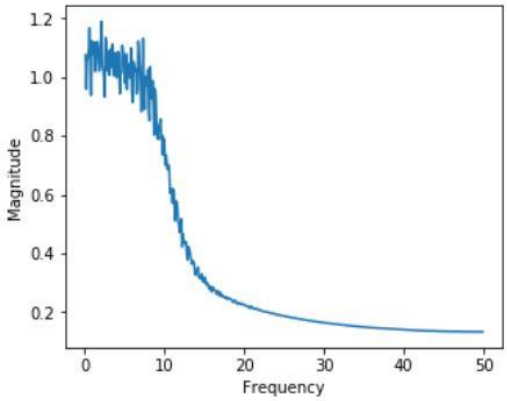


Jogging

Figure 8 Peaks and contour heights of the filtered segments of the Y axis



Sleeping



Sitting

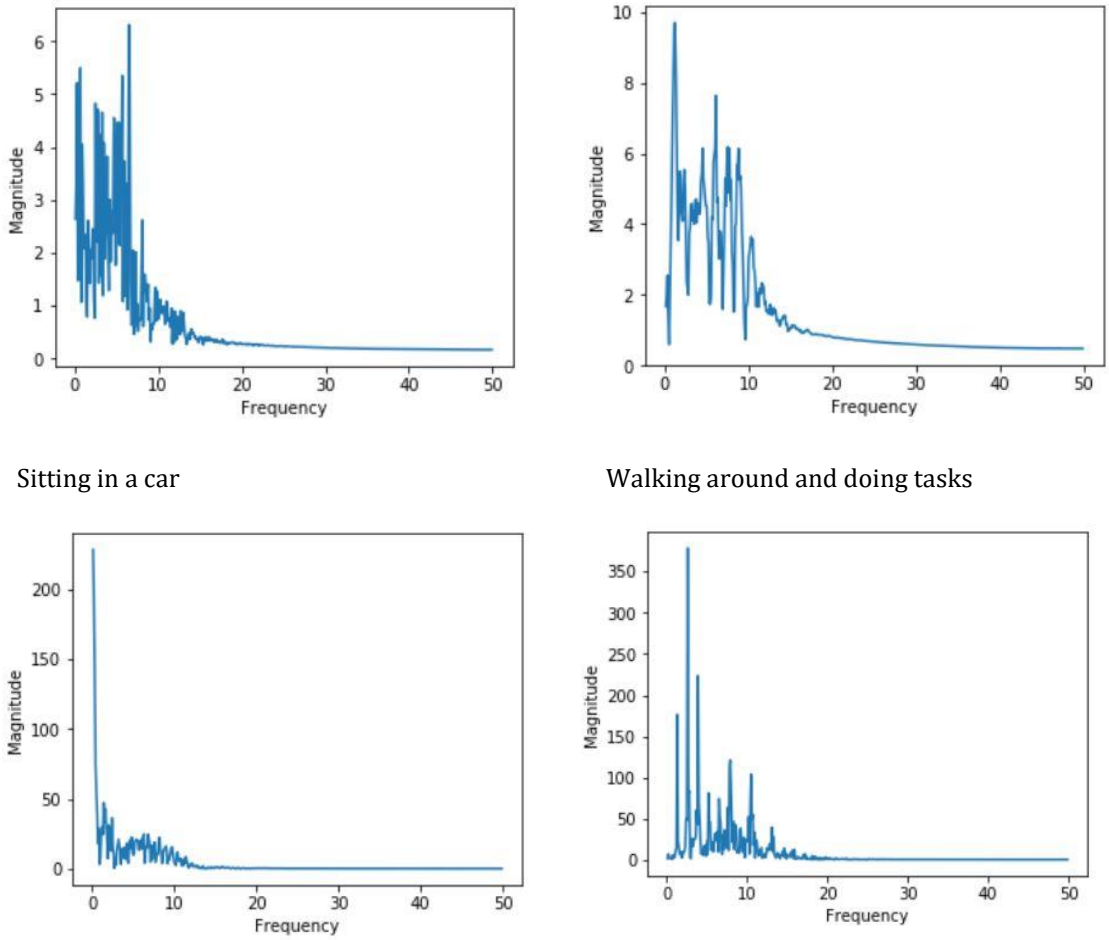


Figure 9 Magnitude values of segments of the Y axis in the frequency domain

5.2.2 Positioning of the Axivity device

The positioning of the Axivity device on the thigh is checked with the median of the acceleration values of the axes in each segment. The axis, the median of which is closest to 1 (or -1) i.e., the gravity (9.81 m/s/s), is the vertical axis and the others are horizontal axes in the segment. For example, in the segments that have been labeled as sitting or sitting in a car, the median of Z axis is close to -1 and the vertical axis is Z. The median of X axis is close to -1 or 1 in the segments labeled as sleeping. In addition, the median of Y axis is close to 1 in the segments labeled as walking and doing tasks, taking a walk, or jogging.

If the vertical axis seems to differ in the segments labeled as the same activity, it should be considered, if the positioning of the Axivity device has changed during recording. Then a conversion of the axes may be needed to keep the acceleration data comparable in the analysis.

5.3 Finding clusters

5.3.1 K-means clustering

The unsupervised machine learning method, K-means clustering, is first used to find clusters in the data including both unannotated acceleration data that has been collected from participants of the study and new annotated acceleration data. The clusters with similar features could correspond to similar activity types performed by the users of the Axivity accelerometer device.

The data of one day from eight users each and four days of the annotated data of one user is selected and pre-processed. The data is split into segments of 10 seconds and filtered with Butterworth low-pass filtering using cut level 10 Hz and order of 4. The features are extracted from the filtered data from each segment of the X, Y and Z axis in the time and frequency domain and standardized. Then, K-means clustering is applied to find clusters with similar features.

The results with different parameters k (the number of the clusters) of K-means clustering are first evaluated using the Silhouette coefficient that measures how far the data instances are from the data instances of the same cluster and other clusters in the scale from -1 to 1. The best parameter value of k is 2 with Silhouette coefficient 0,66. The k value 6 that is the true number of the labels is selected and has the Silhouette coefficient value 0,23.

5.3.2 Visualisation of clusters

Scatter plots of the selected features added with the information of the clusters are plotted to analyze how well the features of the segments have been able to separate the clusters found by K-means clustering. For example, the largest value of Y axis and the mean of the magnitude values of Y axis can separate the 6 clusters assigned by the K-means clustering quite well. The scatter plots of some selected features are shown in Figure 10.

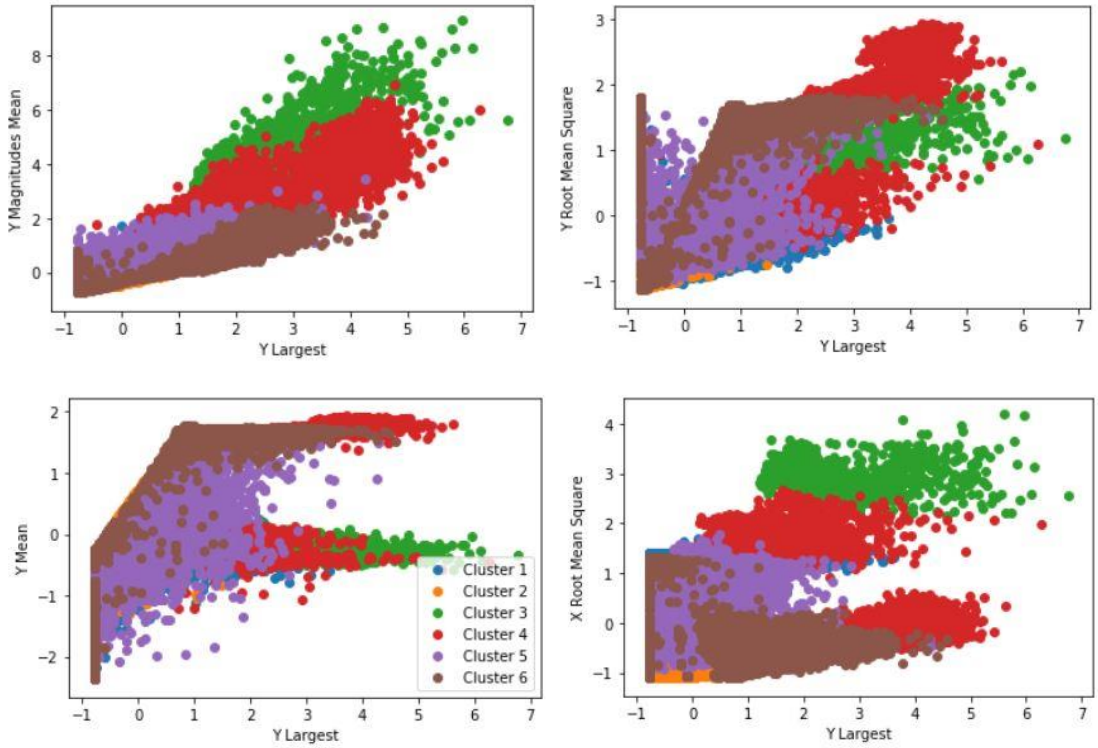


Figure 10 **Scatter plots of the selected features with the information of belonging to the clusters found by K-means clustering**

Next, the clusters found by K-means clustering in all the training data and the true clusters of the annotated data are visualized with PCA with two principal components. Some similarity can be seen between the clusters found by K-means clustering and the true clusters with the two principal components of PCA. It can also be seen that no annotated data is assigned to the cluster 3 found by K-means clustering. The results of the comparison are shown in Figure 11.

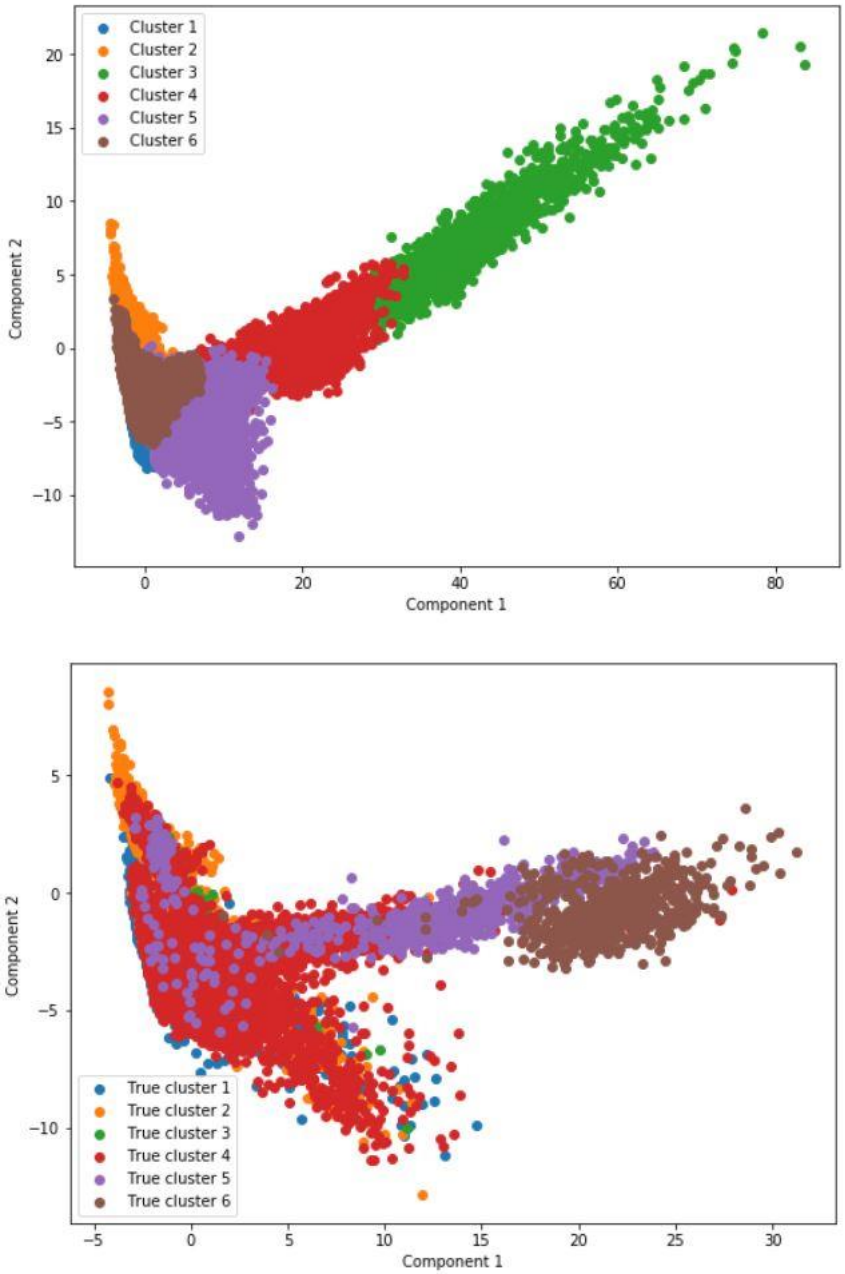


Figure 11 PCA with the clusters found by K-means clustering above and the true clusters of the annotated data below

5.3.3 Performance of K-means clustering

A confusion matrix where the true labels of the annotated data and the labels predicted by K-means clustering are compared in a matrix is computed. Information about how the annotated data has been assigned to the clusters found by K-means clustering helps to evaluate the reliability of K-means clustering to assign all the data including both unannotated and annotated data into clusters.

K-means clustering has been able to identify the actual cluster 6 (jogging) almost perfectly with accuracy near 100 %. Also, the actual cluster 1 (sitting) and 3 (walking around and doing tasks) have been recognized quite well, with 89 % and 84 % accuracies, although the latter has been split into two separate clusters. The actual cluster 0 (sleeping) has been confused with the actual cluster 1 (sitting) and 3 (walking around and doing tasks). The actual cluster 2 (sitting in a car) has been assigned to the same cluster as actual cluster 1 (sitting), and the actual cluster 5 (taking a walk) has been assigned to the same cluster as the actual cluster 6 (jogging). In addition, no data instances of the annotated data have been assigned to one cluster found by K-means clustering. This refers to an activity type that has not been performed when collecting and annotating data for one person. The confusion matrix of K-means clustering is shown in Figure 12.

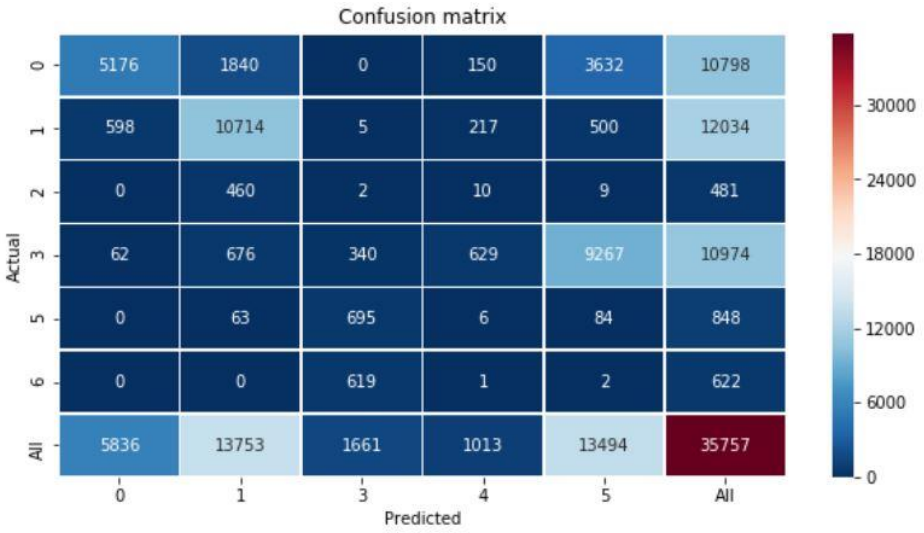


Figure 12 Confusion matrix of K-means clustering

In addition, ARI is computed to evaluate similarity between the clusters found by K-means clustering and the true clusters given in the annotation. A value close to 0 means random labeling and 1, that the clusters are identical. The value of ARI is 0,45 which shows that there is some similarity between the clusters found by K-means clustering compared to the true clusters. The clustering accuracy is 72 %.

It is shown that it is possible to recognize physical activities from unannotated acceleration data of the Axivity accelerometer device positioned on a thigh with K-means clustering with 72 %, accuracy and ARI 0,45.

5.4 Training supervised classifiers

Next, supervised machine learning methods KNN and RF are applied on the standardized features extracted from the filtered segments of the annotated data. The aim is to study, if a KNN model or a RF model trained on labeled data of one person can reliably predict labels and recognize activities from unannotated acceleration data of other persons.

5.4.1 KNN classification

The best parameter value k (the number of the neighbors) is selected for KNN using a separate training set (three days) and test set (a new day) to avoid overfitting of the model because of possible dependencies during the same days. The best k value is 12 resulting in a C-index value 0,95. Other KNN parameters like different distance and weight parameters are also tested, however not improving the best result.

The final model is trained with both the training and the test set of the previous phase and evaluated with a test set of two new days. With the k equals to 12 C-index and accuracy are 0,93 and 88%, respectively.

A confusion matrix, where the actual labels and the labels predicted by KNN are compared in a matrix, is computed. KNN has been able to identify the actual label 1 (sitting), 3 (walking around and doing tasks), and 5 (taking a walk) well with 94 %, 93 % and 91 % accuracies. The actual label 0 (sleeping) has somewhat been confused with the actual label 1 (sitting), the actual label 2 (sitting in a car) with the actual label 1 (sitting), and the actual label 6 (jogging) with the actual label 5 (taking a walk). Sleeping, sitting in a car, and jogging have been recognized with 81 %, 78 % and 87 % accuracy correspondingly. The confusion matrix is shown in Figure 13.

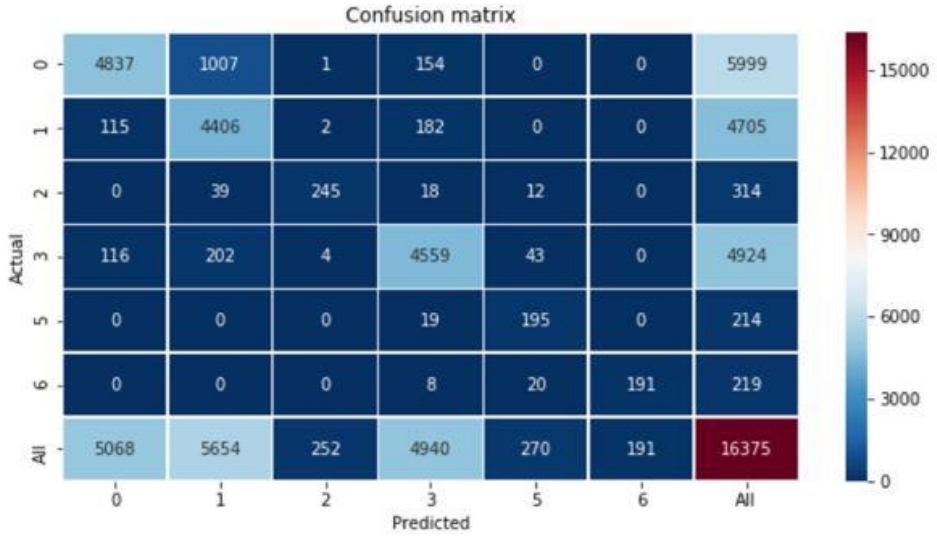


Figure 13 Confusion matrix of KNN model

The results show that the KNN classifier trained on annotated data of one person can predict activities from acceleration data of the same person with C-index value 0,93 and 88 % accuracy.

5.4.2 Random Forest classification

The parameter n_estimators (the number of forests) is set to 500, and RF is applied on the same training and test sets as when evaluating the final KNN model. The result of C-index with the RF model is 0,93 and the accuracy is 88 %, the same as with the KNN model.

A confusion matrix is also computed for RF. Like KNN, RF has been able to identify the actual label 1 (sitting), 3 (walking around and doing tasks), and 5 (taking a walk) well with 95 %, 92 % and 92 % accuracies. Moreover, like with KNN, the actual label 0 (sleeping) has somewhat been confused with the actual label 1 (sitting), the actual label 2 (sitting in a car) with the actual label 1 (sitting), and the actual label 6 (jogging) with the actual label 5 (taking a walk). Sleeping, sitting in a car, and jogging have been recognized with 79 %, 82 % and 87 % accuracy. The confusion matrix is shown in Figure 14.

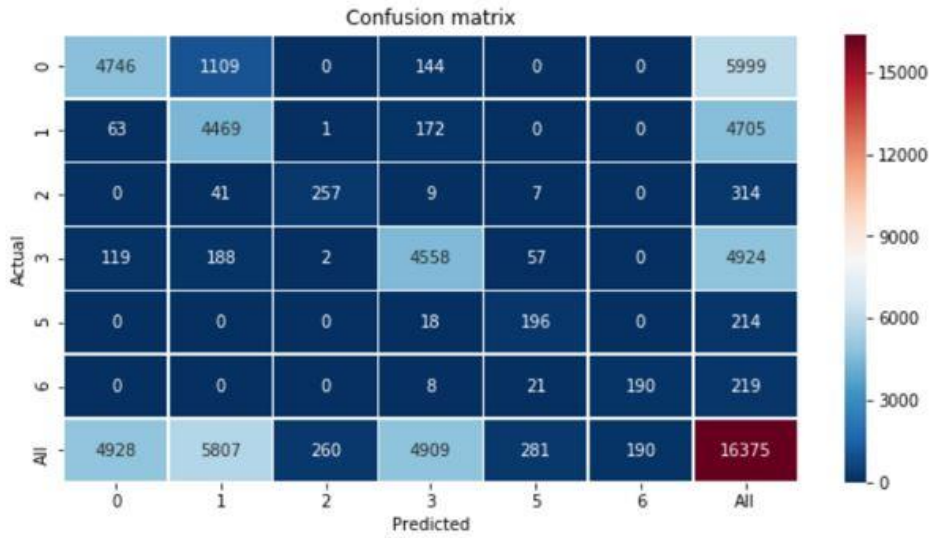


Figure 14 Confusion matrix of RF model

The results show that also the RF classifier trained on annotated data of one person has learned to recognize activities from acceleration data of the same person with the C-index value 0,93 and 88 % accuracy.

5.4.3 The importance of the features

The importance of the features when training the RF model is calculated. This information would be useful in a feature selection phase that could be made to further improve the classification models. The results of the most important and least important features are plotted in Figures 15 and 16.

The 4 most important features have been extracted from the values of the Y axis. They are Y Largest, Y Magnitudes mean, Y Root mean square and Y Mean. The following features are next: X Root mean square, Y median, Y power spectral density of a frequency 7, X Magnitudes median, Y power spectral density of a frequency 6 and Z Median. The least important features are Y Peak widths sum, Y Magnitudes Kurtosis, Y dominant frequency, Z Dominant frequency, and X Dominant frequency.

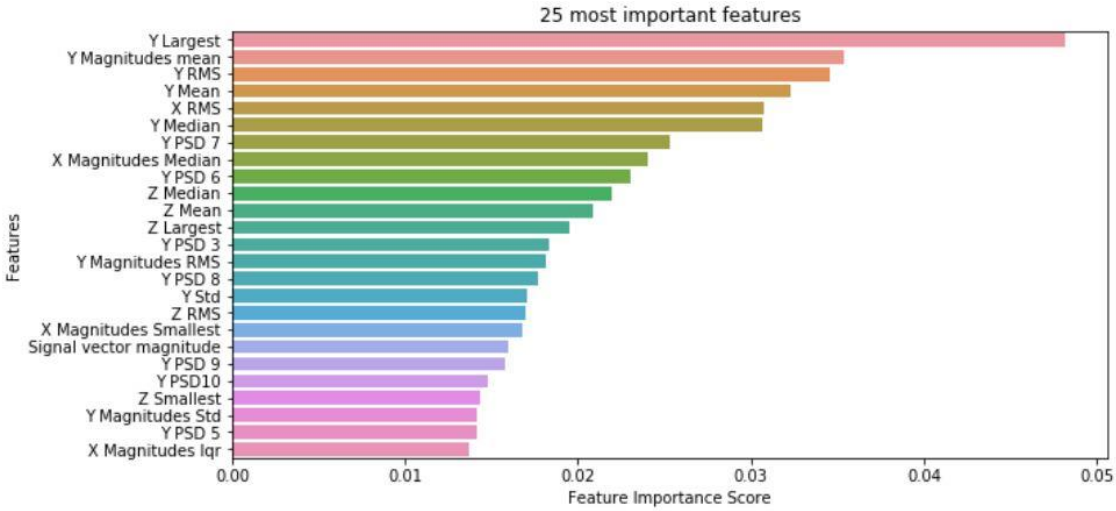


Figure 15 The most important features when training RF model

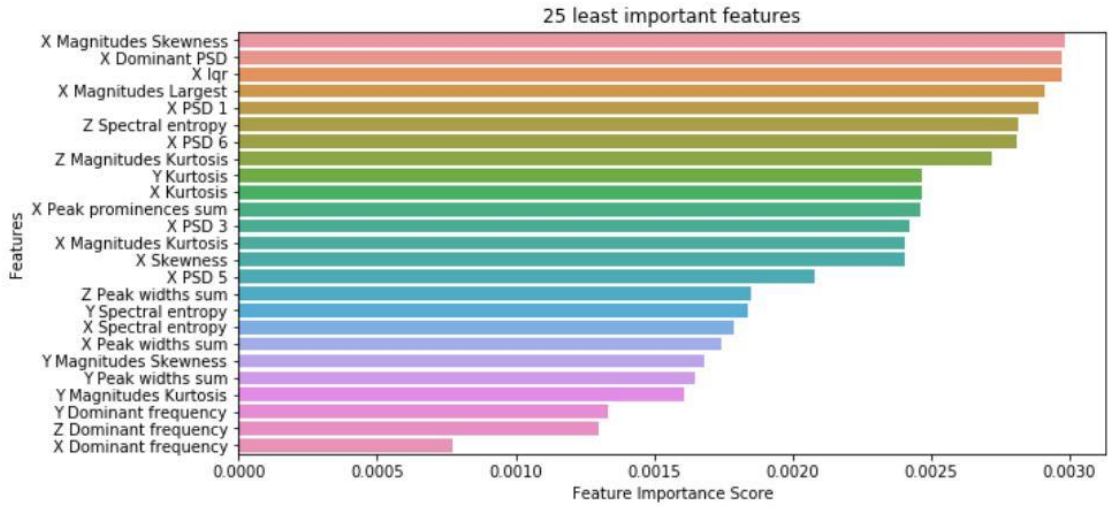


Figure 16 The least important features when training RF model

5.4.4 Reliability of supervised classifiers

The reliability of the previously trained KNN and RF classifiers to recognize activities from unannotated data of other persons is studied next.

5.4.4.1 Using activity levels as reference information

Because there are no true labels and ground truth available, the cut point analysis of OMGUI software [18] is performed to obtain activity levels from the same data. The aim is to compare activities predicted by the classifiers to activity levels produced by the cut point analysis. The reliability of the models to predict physical activities from unannotated acceleration data can be studied with this reference information.

5.4.4.1.1 New metric: fraction of predictions with correct activity levels

A new metric is introduced: a fraction of the labels that correspond to correct activity levels out of all the predicted labels. Different activity types should have activity levels that are shown in Table 5 in the section 4.2. The metric is used to get information about the reliability of the classifiers to predict labels and recognize activities from unannotated acceleration data. If a classifier predicts an activity type that can have an activity level predicted by the cut point analysis, the prediction is correct also according to this reference information.

The new metric is able to highlight the predictions that have a wrong activity type according to the cut point analysis produced from the same data. For example, if sleeping or sitting has been predicted by the classifier, and an activity level predicted by the cut

point analysis is high or vigorous activity the metric interprets the prediction false. The metric cannot differentiate the predictions that share the same activity type. For example, if the prediction is sleeping and the true activity is sitting, the new metric interprets the prediction true because the activity could be sleeping also according to the cut point analysis.

Despite of these limitations, the new metric can help to acquire information about the reliability of recognizing activities from unannotated acceleration data based on this additional source of information when there are no true labels and the ground truth available.

5.4.4.2 Predictions from unannotated data

The previously trained KNN and RF classifiers are run to predict activities from unannotated data collected from 8 users of the Axivity device for one day each. The new metric, a fraction of the predictions that correspond to the correct activity levels in the cut point analysis, is then calculated for the supervised models.

The results are 97 % for the KNN model and 98 % for the RF model. The KNN model predicts activities with the following results of the new metric: sleeping with 97 %, sitting, sitting in a car, and walking around and doing tasks with 100 %, taking a walk 57 %, and jogging with 84 %. The RF model predicts sleeping with 97 %, sitting, and sitting in a car with 100%, walking around and doing tasks 98 %, taking a walk 75 %, and jogging with 95 %.

The results show that the KNN and RF classifiers trained with the small, labeled data of one person can make predictions that are correct also according to the activity levels predicted in the cut point analysis with 97 % and 98 % “accuracy” from unannotated acceleration data of other persons. The predicted activities sleeping, sitting, sitting in a car, and walking around and doing tasks correspond well to activity levels produced by the cut point analysis of the OMGUI software. Taking a walk has been predicted

somewhat differently from the expected activity levels. The RF classifier has predicted jogging corresponding well to the activity levels, while the KNN classifier has predicted jogging partly differently.

Activity levels of the cut point analysis of the OMGUI software and the predictions made by the KNN and RF classifiers are compared in Figure 17.

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 00:26	0	0	0
2019-07-02 00:26	0	0	0
2019-07-02 00:26	0	0	0
2019-07-02 00:26	0	0	0
2019-07-02 00:26	0	0	0
2019-07-02 00:26	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:27	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:28	0	0	0
2019-07-02 00:29	0	0	0
2019-07-02 00:29	0	0	0
2019-07-02 00:29	0	3	0
2019-07-02 00:29	0	1	0
2019-07-02 00:29	0	1	0
2019-07-02 00:29	0	1	0

RF has predicted sleeping (0)

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 00:09	0	1	1
2019-07-02 00:09	0	0	1
2019-07-02 00:09	0	0	1
2019-07-02 00:09	0	0	1
2019-07-02 00:09	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:10	0	0	1
2019-07-02 00:39	0	1	1
2019-07-02 00:39	0	1	1
2019-07-02 00:39	0	1	1
2019-07-02 00:40	0	1	1
2019-07-02 00:40	0	1	1
2019-07-02 00:40	0	1	1
2019-07-02 00:40	0	1	1
2019-07-02 00:40	0	1	1
2019-07-02 00:41	0	1	1
2019-07-02 00:41	0	1	1
2019-07-02 00:41	0	1	1
2019-07-02 00:41	0	1	1
2019-07-02 00:41	0	1	1
2019-07-02 00:41	0	1	1

RF has predicted sitting (1)

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 12:30	0	1	2
2019-07-02 12:30	0	1	2
2019-07-02 13:45	1	2	2
2019-07-02 15:03	1	2	2
2019-07-02 15:13	0	2	2
2019-07-02 15:13	0	2	2
2019-07-02 15:14	1	2	2
2019-07-02 15:14	1	1	2
2019-07-02 15:14	1	1	2
2019-07-02 15:15	0	2	2
2019-07-02 15:15	0	2	2
2019-07-02 15:15	0	1	2
2019-07-02 15:16	1	2	2
2019-07-02 15:16	1	2	2
2019-07-02 15:16	1	2	2
2019-07-02 15:16	1	2	2
2019-07-02 15:16	1	2	2
2019-07-02 15:16	1	2	2
2019-07-02 15:17	1	2	2
2019-07-02 15:17	1	1	2
2019-07-02 15:17	1	1	2
2019-07-02 15:17	1	2	2
2019-07-02 15:17	1	2	2

RF has predicted sitting in a car (2)

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 01:08	0	0	3
2019-07-02 02:25	0	3	3
2019-07-02 02:25	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:26	0	3	3
2019-07-02 02:27	0	3	3
2019-07-02 02:27	0	3	3
2019-07-02 02:38	0	1	3
2019-07-02 03:02	0	3	3
2019-07-02 03:02	0	3	3
2019-07-02 03:03	0	0	3
2019-07-02 03:14	0	3	3
2019-07-02 03:14	0	3	3
2019-07-02 03:15	0	3	3
2019-07-02 03:15	0	3	3
2019-07-02 03:15	0	3	3
2019-07-02 03:15	0	3	3
2019-07-02 03:15	0	3	3
2019-07-02 03:15	0	3	3

RF has predicted walking around and doing tasks (3)

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 10:58	1	5	5
2019-07-02 11:00	2	5	5
2019-07-02 11:00	2	5	5
2019-07-02 11:02	1	6	5
2019-07-02 11:07	2	5	5
2019-07-02 11:09	1	5	5
2019-07-02 12:24	2	5	5
2019-07-02 14:53	2	5	5
2019-07-02 14:53	2	6	5
2019-07-02 14:54	1	5	5
2019-07-02 14:54	1	5	5
2019-07-02 15:12	1	5	5
2019-07-02 15:12	1	5	5
2019-07-02 15:12	1	5	5
2019-07-02 15:23	2	5	5
2019-07-02 15:23	2	5	5
2019-07-02 15:24	1	5	5
2019-07-02 15:33	1	5	5
2019-07-02 16:00	2	5	5
2019-07-02 16:00	2	5	5
2019-07-02 16:00	2	5	5
2019-07-02 16:09	2	5	5
2019-07-02 16:16	2	5	5

RF has predicted taking a walk (5)

Time minutes	Activity level	Predicted KNN	Predicted RF
2019-07-02 17:59	3	5	6
2019-07-02 17:59	3	5	6
2019-07-02 17:59	3	5	6
2019-07-02 17:59	3	5	6
2019-07-02 17:59	3	5	6
2019-07-02 18:00	3	5	6
2019-07-02 18:00	3	5	6
2019-07-02 18:00	3	5	6
2019-07-02 18:00	3	5	6
2019-07-02 18:01	3	6	6
2019-07-02 18:01	3	5	6
2019-07-02 18:01	3	6	6
2019-07-02 18:01	3	6	6
2019-07-02 18:01	3	6	6
2019-07-02 18:02	3	6	6
2019-07-02 18:02	3	6	6
2019-07-02 18:02	3	5	6
2019-07-02 18:02	3	5	6
2019-07-02 18:02	3	6	6
2019-07-02 18:02	3	6	6

RF has predicted jogging (6)

Figure 17 Comparing activity levels of the cut point analysis and predictions of the KNN and RF classifiers

5.4.4.3 Predictions from annotated data

The same metric is also computed for the supervised KNN and RF models that have predicted activities from the test data including only the labeled data of one person. The aim is to compare the results of the new metric to the results when predicting activities from unannotated data of other persons.

The results are near 100 % for both the KNN and the RF model. The KNN model predicts activities with the following results: sleeping, sitting, sitting in a car, and walking around and doing tasks with 100 %, taking a walk with 91 %, and jogging with 95 % “accuracy”. The RF model predicts sleeping, sitting, sitting in a car, and walking around and doing tasks with 100 %, taking a walk with 94 %, and jogging with 96 %.

The results show that all the predicted activities correspond well to activity levels of the cut point analysis of the OMGUI software when predictions have been made from the data of the same person whose data has been used in training. It is also shown that the predictions better correspond to the activity levels, compared to the results when making predictions from unannotated data of other persons.

Activity levels of the cut point analysis of the OMGUI software, true labels, and the predictions made by the KNN and RF classifiers are compared in Figure 18.

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-08-01 00:36	0	0	3	3
2020-08-01 00:36	0	0	3	3
2020-08-01 00:36	0	0	2	1
2020-08-01 00:36	0	0	1	1
2020-08-01 00:36	0	0	1	1
2020-08-01 00:36	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:37	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:38	0	0	0	0
2020-08-01 00:39	0	0	0	0
2020-08-01 00:39	0	0	0	0
2020-08-01 00:39	0	0	0	0
2020-08-01 00:39	0	0	0	0
2020-08-01 00:39	0	0	0	0
2020-08-01 00:39	0	0	0	0

True label sleeping (0)

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-07-31 20:31	0	1	1	1
2020-07-31 20:31	0	1	1	1
2020-07-31 20:31	0	1	1	1
2020-07-31 20:31	0	1	1	1
2020-07-31 20:31	0	1	1	1
2020-07-31 20:31	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:32	0	1	1	1
2020-07-31 20:33	0	1	1	1
2020-07-31 20:33	0	1	1	1
2020-07-31 20:33	0	1	1	1
2020-07-31 20:33	0	1	1	1
2020-07-31 20:33	0	1	1	1
2020-07-31 20:34	0	1	1	1
2020-07-31 20:34	0	1	1	1
2020-07-31 20:34	0	1	1	1
2020-07-31 20:34	0	1	1	1
2020-07-31 20:34	0	1	1	1
2020-07-31 20:34	0	1	1	1

True label sitting (1)

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-08-01 11:58	0	2	2	2
2020-08-01 11:58	0	2	2	2
2020-08-01 11:58	0	2	2	2
2020-08-01 11:58	0	2	2	2
2020-08-01 11:58	0	2	2	2
2020-08-01 11:58	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 11:59	0	2	2	2
2020-08-01 12:00	0	2	2	2
2020-08-01 12:00	0	2	1	2
2020-08-01 12:00	0	2	2	2
2020-08-01 12:00	0	2	1	1
2020-08-01 12:00	0	2	2	2
2020-08-01 12:00	0	2	1	1
2020-08-01 14:54	1	2	3	3
2020-08-01 14:54	1	2	3	3
2020-08-01 14:54	1	2	3	3
2020-08-01 14:54	1	2	3	3
2020-08-01 14:54	1	2	2	2
2020-08-01 14:54	1	2	3	1

True label sitting in a car (2)

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-07-31 20:39	1	3	1	1
2020-07-31 20:39	1	3	3	3
2020-07-31 20:39	1	3	3	3
2020-07-31 20:39	1	3	3	3
2020-07-31 20:39	1	3	3	3
2020-07-31 20:39	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:40	1	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:41	0	3	3	3
2020-07-31 20:42	0	3	3	3
2020-07-31 20:42	0	3	3	3
2020-07-31 20:42	0	3	3	3
2020-07-31 20:42	0	3	3	3
2020-07-31 20:42	0	3	3	3
2020-07-31 20:42	0	3	3	3

True label walking around and doing tasks (3)

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-08-13 18:35	1	5	3	3
2020-08-13 18:35	1	5	3	3
2020-08-13 18:35	1	5	3	3
2020-08-13 18:35	1	5	3	3
2020-08-13 18:35	1	5	3	3
2020-08-13 18:35	1	5	3	3
2020-08-13 18:36	2	5	3	3
2020-08-13 18:36	2	5	3	3
2020-08-13 18:36	2	5	3	3
2020-08-13 18:36	2	5	3	3
2020-08-13 18:36	2	5	5	5
2020-08-13 18:36	2	5	5	5
2020-08-13 18:37	1	5	3	5
2020-08-13 18:37	1	5	5	5
2020-08-13 18:37	1	5	3	3
2020-08-13 18:37	1	5	3	3
2020-08-13 18:37	1	5	3	3
2020-08-13 18:37	1	5	3	3
2020-08-13 18:38	2	5	3	3
2020-08-13 18:38	2	5	5	5
2020-08-13 18:38	2	5	5	5
2020-08-13 18:38	2	5	5	5
2020-08-13 18:38	2	5	5	5
2020-08-13 18:38	2	5	5	5

True label taking a walk (5)

Time minutes	Activity level	True label	Predicted KNN	Predicted RF
2020-08-01 18:34	2	6	3	3
2020-08-01 18:34	2	6	3	3
2020-08-01 18:34	2	6	5	5
2020-08-01 18:34	2	6	3	3
2020-08-01 18:34	2	6	3	3
2020-08-01 18:34	2	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:35	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:36	3	6	6	6
2020-08-01 18:37	3	6	6	6
2020-08-01 18:37	3	6	6	6
2020-08-01 18:37	3	6	6	6
2020-08-01 18:37	3	6	6	6
2020-08-01 18:37	3	6	6	6
2020-08-01 18:37	3	6	6	6

True label jogging (6)

Figure 18 **Comparing activity levels of the cut point analysis, true labels, and predictions of the KNN and RF classifiers**

5.5 Improving classifiers in semi-supervised setting

Next, the En-Co-Training method is used with the supervised KNN and RF classifiers that have been trained with the labeled data of one person to leverage knowledge from unannotated acceleration data of other users of the Axivity device. It is studied if the classifiers can be improved to better generalize on unannotated data of other persons. It is investigated if the KNN and RF classifiers retrained in a semi-supervised setting can recognize activities more reliably than the initial supervised classifiers.

The initial KNN and RF classifiers first make predictions from the training data of 2 persons. The predictions that both the classifiers have consensus about and have a right corresponding activity level are accepted. In addition, if one of the models has predicted jogging and the activity level is vigorous activity, the prediction is considered to be confident enough and is accepted because jogging should clearly correspond to this one activity level. The accepted predictions are added to the set of the true labels as pseudo-labels and the corresponding data instances are added to the common training set of the

models. The models are retrained, and new predictions are made from the training data of 2 new persons. This is iterated until predictions have been made from all the training data of 8 persons.

Instead of using majority voting of three classifiers like in the work [35] the KNN and RF classifiers are used separately to make predictions. In addition, unlike in [35] also the activity levels produced by the cut point analysis of the OMGUI software are considered as explained previously to increase the confidence of the accepted predictions. This way both information can be leveraged from unannotated acceleration data, and activity levels can be used as additional source of information to increase the confidence of selected pseudo-labels in the semi-supervised setting.

5.5.1 Reliability of classifiers in semi-supervised setting

5.5.1.1 Predictions from unannotated data

The semi-supervised training of the initial KNN and RF classifiers is performed and evaluated three times on separate training and test sets of unannotated acceleration data. In each test round one-day unannotated acceleration data collected from 8 individuals are used as the training data, and one-day unannotated acceleration data of 4 individuals as the test data.

Similar to the previous evaluation, the new metric, a fraction of the predictions that correspond to the correct activity levels in the cut point analysis of the OMGUI software, is calculated to evaluate the reliability of the models. In addition, the number of predictions that correspond to correct activity levels is calculated. The results of both the initial and the retrained classifiers are shown in Table 9. If the number of correct predictions shown in parenthesis has improved compared to the initial classifier, the result

is bolded. In the test round 3 no jogging has been performed by the persons during the selected days. Therefore, the last line in Table 9 is excluded from the comparison of the results.

Table 9 A fraction and the number of predictions that correspond to correct activity levels of the cut point analysis

Predicted Activity	**Initial KNN classifier**	**Retrained KNN classifier**	**Initial RF classifier**	**Retrained RF classifier**
Test 1 all activities	97 % (32264)	**98 % (32597)**	98 % (32597)	**99 % (32929)**
Sleeping	99 % (12123)	**99 % (13612)**	98 % (11914)	**99 % (12868)**
Sitting	100 % (13060)	100 % (11751)	100 % (12503)	100 % (12189)
Sitting in a car	100 % (183)	**100 % (224)**	100 % (180)	100 % (177)
Walking around and doing tasks	100 % (5839)	**100 % (5883)**	98 % (6842)	100 % (6232)
Taking a walk	58 % (905)	67 % (648)	81 % (510)	**77 % (627)**
Jogging	96 % (360)	**97 % (668)**	95 % (770)	**95 % (810)**
Test 2 all activities	96 % (33074)	**99 % (34108)**	97 % (33419)	**98 % (33764)**
Sleeping	95 % (12179)	**98 % (13215)**	96 % (13157)	95 % (12981)
Sitting	100 % (13350)	100 % (11861)	100 % (10854)	100 % (11677)
Sitting in a car	100 % (1632)	**100 % (1760)**	100 % (2429)	100 % (2040)
Walking around and doing tasks	100 % (5081)	**100 % (6324)**	97 % (6564)	100 % (5890)
Taking a walk	55 % (836)	77 % (358)	63 % (219)	**81 % (516)**
Jogging	0 % (0)	**98 % (548)**	95 % (334)	**98 % (535)**
Test 3 all activities	98 % (33043)	98 % (33043)	98 % (33043)	98 % (33043)
Sleeping	98 % (12520)	98 % (12114)	97 % (12540)	**98 % (12814)**
Sitting	100 % (11809)	100 % (11566)	100 % (11998)	100 % (11285)
Sitting in a car	100 % (804)	**100 % (839)**	100 % (1013)	100 % (886)
Walking around and doing tasks	100 % (7326)	**100 % (8341)**	99 % (7324)	**100 % (7771)**
Taking a walk	81 % (799)	73 % (428)	79 % (291)	**75 % (496)**
Jogging	*0 % (0)*	*8 % (2)*	*23 % (3)*	*5 % (2)*

The overall results are 96-98 % for the initial classifiers and 98-99 % for the retrained classifiers. Sleeping, sitting, sitting in a car, and walking around and doing tasks have been predicted with over 95 % by all the initial and retrained classifiers. The results of taking a walk predicted by the KNN classifiers have changed from 55-81 % to 67-77 % and jogging from 0-96 % to 97-98 %. The results of the RF classifiers when predicting taking a walk and jogging have changed from 63-81 % to 77-81 %, and from 95 % to 95-98 % respectively. The number of correctly predicted activities taking a walk and jogging have either stayed the same or improved by all the retrained RF classifiers. The number of correctly predicted jogging has always improved by the retrained KNN classifiers.

It can be concluded that the semi-supervised setting using the En-Co-Training method and the KNN and RF classifiers trained with only small, annotated data of one person can improve the initial supervised KNN and RF classifiers. In addition, the retrained classifiers can predict activities taking a walk and jogging more reliably from acceleration data of other persons.

5.5.1.2 Predictions from annotated data

To evaluate the reliability of the KNN and RF classifiers retrained in the 3 test rounds in the semi-supervised setting, they are also tested on the same annotated test data of one person that has been used when evaluating the initial supervised classifiers. The new metric, a fraction of the predictions that correspond to the correct activity levels in the cut point analysis, is first calculated. Like with the initial supervised classifiers, the overall results are near 100 % for both the KNN and the RF models. Also, all the results of all the activities are between 90 % and 100 % like with the initial KNN and RF classifiers tested earlier.

The C-index of the retrained KNN and RF classifiers is also calculated. The C-index values are 0,94, 0,93 and 0,93 in the test round 1, 2 and 3, respectively. The C-index value has either stayed the same or improved compared to the C-index value 0,93 of the initial classifiers. The accuracies have also stayed the same in all the test rounds: i.e., 88 % for both the classifiers.

The confusion matrices are plotted from the results of the retrained KNN and RF classifiers in the test round 1 in Figures 19 and 20. The results are very similar to the results of the initial supervised classifiers tested on the same data.

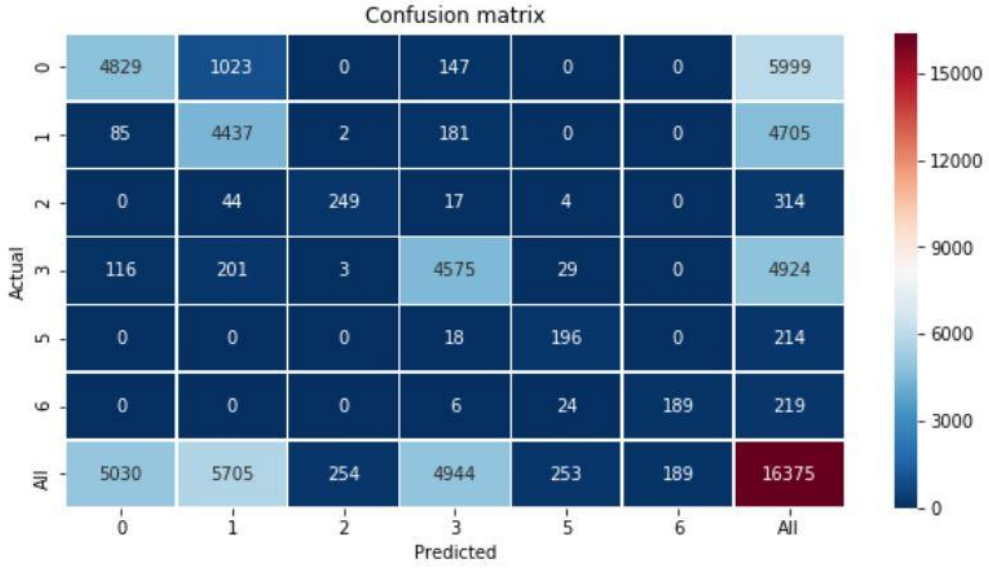


Figure 19 Confusion matrix of retrained KNN model

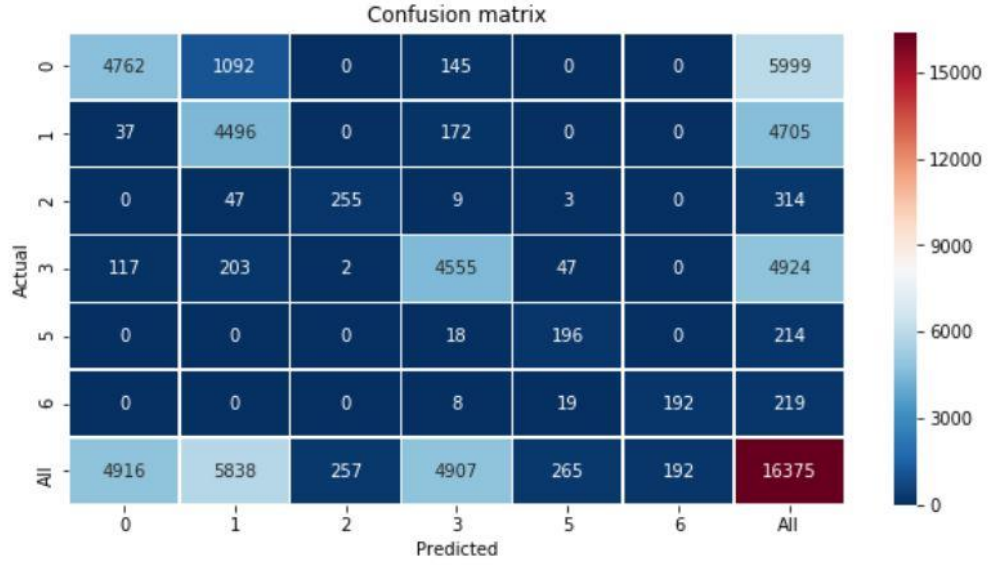


Figure 20 Confusion matrix of retrained RF model

The results show that the KNN and RF classifiers retrained in a semi-supervised setting can recognize activities as well as the initial KNN and RF classifiers when measuring them both with the new metric and with the C-index value and accuracy. Retraining the KNN and RF classifiers in a semi-supervised setting has not decreased their performance on the annotated test data.

6 Discussion

This thesis introduces a solution to extract physical activities from unannotated acceleration data collected with an Axivity device positioned on a thigh using traditional unsupervised, supervised, and semi-supervised machine learning methods. It is shown to be beneficial to collect and label new acceleration data although for only one person and to use the labeled data to develop supervised KNN and RF classifiers to retrain them in a semi-supervised setting using the En-Co-Training method. A new metric is proposed: a fraction of the labels that correspond to correct activity levels out of all the predicted labels according to the cut-point analysis of the OMGUI software [18]. The reliability of the classifiers is shown to consistently improve when comparing the retrained KNN and RF classifiers to the initial ones with the new metric.

Although deep learning methods have outperformed traditional machine learning methods in HAR, most of them are supervised methods that require an extensive amount of labeled training data and are not feasible solutions in this thesis when only unlabeled acceleration of 12 people is available. Deep unsupervised and semi-supervised methods are not suitable either because of the small amount of available data. The traditional KNN and RF methods are easy to implement, and they have been successful in HAR outperforming methods like SVM and Naïve Bayes [28,53]. Also, the En-Co-Training method has performed well in HAR [35].

Furthermore, traditional machine learning methods are more competitive with deep learning methods when an objective is to recognize basic physical activities such as sleeping, sitting, sitting in a car, walking around and doing tasks, taking a walk and jogging that are used as activity types in this thesis. Physical activity of people can be interpreted with these basic activities common in daily life. They can also easily be compared with activity levels, reference information that can be obtained with the cut point analysis of the OMGUI software. However, if very different activities have been performed by other persons the classifiers might be inaccurate. Performing and labeling

more different activity types and adding them to the training data could improve the reliability of the supervised classifiers.

The classifiers retrained in the semi-supervised setting could be further improved by adding more iterations and more unannotated acceleration data to the training data of the En-Co-Training method. In that way, more examples of activities performed by other people would be added. Adding more unannotated data would also increase the risk of choosing wrong predictions as pseudo-labels, and it might decrease the performance compared to the initial supervised classifiers. To mitigate this risk, the reliability of the retrained classifiers should be compared to the supervised classifiers after retraining.

Also, the reliability of the classifiers could be improved by using data collected from another device, for example a smartwatch positioned on a wrist in addition to the Axivity accelerometer positioned on a thigh. It would be possible to better separate stationary activities like sleeping and sitting where the position of a thigh may be quite identical or taking a walk from walking around and doing tasks. It would require collecting and annotating new acceleration data using both the devices, training supervised classifiers on new training data, and retraining them in the semi-supervised setting.

7 Conclusion

In this thesis, the objective was to develop a machine learning solution that can recognize physical activities from unannotated acceleration data collected with an Axivity accelerometer positioned on a thigh. The solution was tested on real-life acceleration data collected from 12 people. It is a challenge in HAR to annotate acceleration data, and the existing approaches in HAR mostly use supervised machine learning methods that require true labels. It was studied if different activities can reliably be extracted from unannotated acceleration data only using unsupervised machine learning methods. Furthermore, it was examined if small, labeled data collected from one person can be utilized with supervised and semi-supervised machine learning methods so that they can recognize activities reliably. In addition, it was studied how to get information about the reliability of the used machine learning methods without knowing true labels and the ground truth.

After a brief introduction to HAR using wearable devices and machine learning, characteristics, and challenges in HAR as well as machine learning methods and evaluation metrics that are commonly used in HAR were presented in Chapter 2. In Chapter 3 the current state of research in sensor based HAR was studied, and works that have successfully used supervised, unsupervised, and semi-supervised machine learning methods were introduced. The works were summarized with used sensors, methods, evaluation metrics, and physical activities that had been recognized. Also, open questions in HAR and new promising research areas that aim at utilizing continuously streaming unlabeled acceleration data were introduced.

Machine learning solutions were developed for recognizing physical activities from unlabeled acceleration data collected with an Axivity accelerometer, and they were described in Chapter 4. First, new acceleration data was collected with the Axivity device positioned on a thigh and annotated for one person. The unsupervised machine learning method K-means clustering was then applied on the preprocessed data including both one-day unannotated data of 8 individuals and new, labeled acceleration data of one person collected for 4 days. The reliability of the K-means clustering to find correct

clusters related to performed physical activities was evaluated studying if the model had been able to assign the true labels to correct clusters.

Second, supervised machine learning classifiers were trained on the labeled acceleration data of one person collected for 4 days. The KNN and RF classifiers were first used to predict activities from labeled data collected from the same person for 3 separate days to evaluate their performance with the known true labels. Then, the classifiers were used to automatically annotate one-day unlabeled acceleration data of 8 individuals to study if the supervised classifiers which were trained on the labeled data of one person could reliably recognize activities from unlabeled acceleration data.

Third, the En-Co-Training method was used to retrain the supervised KNN and RF classifiers in a semi-supervised setting with the training data of 8 persons and the test data of 4 persons collected for one day each in tree test rounds. The activity levels produced by the cut point analysis of the OMGUI software [18] were also used as additional information when choosing confident pseudo-labels. The retrained classifiers were used to automatically annotate unlabeled acceleration data to study if the semi-supervised setting helped the classifiers to predict physical activities more reliably.

In the experiments in Chapter 5 it was shown that the unsupervised K-means clustering could recognize physical activities from data including both unannotated and annotated acceleration data with the ARI value 0,45 and 72 % clustering accuracy. Although the K-means clustering recognized jogging almost perfectly, the stationary activities sleeping and sitting were confused, sitting in a car was assigned to the same cluster as sitting, and taking a walk was assigned to the same cluster as jogging. Also, there seemed to be an activity type that had not been performed when collecting annotated data.

Both the supervised KNN and RF classifiers trained on the labeled data of one person could recognize activities from the data of the same person with the C-index 0,93 and 88 % accuracy. The activities sitting, walking around and doing tasks, and taking a walk were recognized well, but the stationary activities sleeping and sitting in a car had

somewhat been confused with sitting, and jogging was partly misclassified as taking a walk.

However, it was more important to evaluate how reliably the supervised KNN and RF classifiers were able to recognize activities from unlabeled data of other persons. A new metric was proposed: i.e., a fraction of predictions that have a correct activity level according to the cut point analysis of the OMGUI software run on the same unlabeled acceleration data out of all the predictions. This metric was only able to highlight if a classifier predicted an activity that should have a different activity level than the cut point analysis had predicted. However, it was valuable information when no true labels and the ground truth were available.

The new metric was calculated both for the initial supervised KNN and RF classifiers and the classifiers retrained in the semi-supervised setting. The overall results of the initial supervised classifiers to recognize activities from unlabeled data of other users were 96-98 %. The results were 95-100 % for all other activities but 55-81 % for taking a walk and 0-95 % for jogging. The overall results of the classifiers retrained in the semi-supervised setting were 98-99 %. The results of activity types were 95-100 % for all other activities, but 67-81 % for taking a walk, and 95-98 % for jogging. In addition, the number of correctly predicted activities taking a walk and jogging had either stayed the same or improved by the retrained RF classifier in all the test rounds. It was shown that the semi-supervised setting improved the reliability of the classifiers to predict activities that have a correct activity level also according to the cut-point analysis of the OMGUI software.

It could be concluded that when only using unlabeled acceleration data and the unsupervised K-means clustering method the reliability of recognizing activities remained quite modest. The model had challenges to separate stationary activities, and it could not differentiate sitting in a car from sitting and taking a walk from jogging. It was beneficial to collect and label new acceleration data although for only a single person. The labeled data could be used to train supervised KNN or RF classifiers to recognize activities from unannotated acceleration data of other users. Furthermore, the reliability of the KNN and RF classifiers could consistently be improved when they were retrained

in a semi-supervised setting using the En-Co-Training method with the initial supervised KNN and RF classifiers leveraging knowledge from unannotated data. In addition, reference information of the cut point analysis of the OMGUI software could be used to further reduce the risk of choosing wrong pseudo-labels.

It was possible to get information about the reliability of the supervised and semi-supervised classifiers with the new metric, a fraction of predictions that correspond to the activity levels also predicted by the cut point analysis of the OMGUI software out of all the predictions. The metric could not separate activity types that share the same activity level, but with the new metric, it was possible to evaluate how reliably a classifier had predicted activities with correct activity levels also according to the cut point analysis run on the same data.

References

[1] Sheng Taoran: Learning Embeddings for Wearable-based Human Activity Analysis. University of Texas Arlington Theses and Dissertations (library), 2020.

[2] Sreenivasan Ramasamy Ramamurthy, Nirmalya Roy: Recent Trends in Machine Learning for Human Activity Recognition - A Survey. WIREs Data Mining and Knowledge Discovery 8(4), 2018.

[3] Alireza Abedin, Farbod Motlagh, Qinfeng Shi, Damith Rezatofighi, Chinthana Ranasinghe: Towards deep clustering of human activities from wearables. ISWC '20: Proceedings of the 2020 International Symposium on Wearable Computers:1-6, 2020.

[4] Nauman Ahad, Mark A. Davenport: Semi-supervised sequence classification through change point detection. ArXiv Computer Science, Machine Learning, 2020.

[5] Mohammad Sabik Irbaz Abir Azad, Tanjila Alam Sathi, Lutfun Nahar Lota: Nurse Care Activity Recognition Based on Machine Learning Techniques Using Accelerometer Data. UbiComp-ISWC '20: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers:402-407, 2020.

[6] D. Jakhar, I. Kaur: Artificial intelligence, machine learning and deep learning: definitions and differences. Clinical and Experimental Dermatology 45(1):131-132, 2020.

[7] Sunita Kumari Chaurasia, S.R.N Reddy: AI Assisted Human Activity Recognition (HAR). International Journal of Engineering and Advanced Technology (IJEAT) ISSN 8(6):2249-8958, 2019.

[8] Yongjin Kwon, Kyuchang Kang, Changseok Bae: Unsupervised learning for human activity recognition using smartphone sensors. Expert Systems with Applications 41(14):6067-6074, 2014.

[9] M.R. Berthold, C. Borgelt, F. Höppner, F. Klawonn: Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data. Springer, London, 2010.

[10] Marcio de Almeida Mendes, Inacio da Silva, Virgilio Ramires, Felipe Reichert, Rafaela Martins, Rodrigo Ferreira, Elaine Tomasi: Metabolic equivalent of task (METs) thresholds as an indicator of physical activity intensity. Plos One, 2013.

[11] https://axivity.com/downloads/ax3

[12] Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, Mike Stanley: A brief survey of machine learning methods and their sensor and IoT applications. 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), 2018.

[13] Fei Hu, Qi Hao: Intelligent Sensor Networks, The Integration of Sensor Networks, Signal Processing and Machine Learning. CRC Press, 2012.

[14] Jürgen Schmidhuber: Deep learning in neural networks: An overview. Neural Networks 61:85-117, 2015.

[15] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, Lisha Hu: Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters 119:3-11, 2019.
[16] Oscar D. Lara, Miguel A. Labrador: A Survey on Human Activity Recognition using Wearable Sensors. IEEE Communications Surveys & Tutorials 15(3):1192-1209, 2013.
[17] Dale Esliger, Alex Rowlands, Tina Hurst, Michael Catt, Peter Murray, Roger Eston: Validation of the GENEA accelerometer. Medicine and Science in Sports and Exercise, 43(6):1085-1093, 2010.
[18] AX3 GUI · digitalinteraction/openmovement Wiki · GitHub
[19] Oresti Banos, Juan-Manuel Galvez, Miguel DamasOrcID, Hector Pomares, Ignacio Rojas: Window Size Impact in Human Activity Recognition. MDPI Open Access Journals, Sensors 2014 14(4):6474-6499, 2014.
[20] Prajoy Podder, Mehedi Hasan, Rafiqul Islam, Mursalin Sayeed: Design and Implementation of Butterworth, Chebyshev-I and Elliptic Filter for Speech Signal Analysis. International Journal of Computer Applications 98(7):12-18, 2014.
[21] E. O. Brigham, R. E. Morrow: The fast Fourier transform. IEEE Spectrum 4(12):63-70, 1967.
[22] Zhenyu He, Lianwen Jin: Activity Recognition from acceleration data Based on Discrete Consine Transform and SVM. 2009 IEEE International Conference on Systems, Man and Cybernetics, 2009.
[23] Zhenyu He: Activity Recognition from Accelerometer Signals Based on Wavelet-AR Model. 2010 IEEE International Conference on Progress in Informatics and Computing, 2010.
[24] Pekka Siirtola, Juha Röning: Incremental Learning to Personalize Human Activity Recognition Models: The Importance of Human AI Collaboration. Sensors 2019 19(23), 2019.
[25] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz: A public domain dataset for human activity recognition using smartphones. ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2013.
[26] Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, Gerhard Tröster: Wearable activity tracking in car manufacturing. IEEE Pervasive Computing 7(2):42-50, 2008.
[27] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, Claudia Villalonga: A novel Framework for Agile Development of Mobile Health Applications. Lecture Notes in Computer Science 8868:91-98, 2014.
[28] Attila Reiss, Didier Stricker: Introducing a new benchmarked dataset for activity Monitoring. 2012 16th International Symposium on Wearable Computers:108-109, 2012.
[29] Oresti Baños, Miguel Damas, Ignacio Rojas, Máté Attila Tóth, Oliver Amft: A benchmark dataset to evaluate sensor displacement in activity recognition. UbiComp '12: Proceedings of the 2012 ACM Conference on Ubiquitous Computing:1026-1035, 2012.
[30] D. Anguita, A. Ghio, L. Oneto, X. Parra, Jorge Luis Reyes-Ortiz: A public domain dataset for human activity recognition using smartphones. ESANN 2013

proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence, and machine learning, 2013.
[31] Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep learning. MIT Press, 2016.
[32] Lu Bai, Chris Yeung, Christos Efstratiou, Moyra Chikomo: Motion2vector: unsupervised learning in human activity recognition using wrist-sensing data. UbiComp/ISWC '19 Adjunct: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers:537-542, 2019.
[33] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, Mads Møller Jensen: 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:127-140, 2015.
[34] Matthias Schmid, Marvin N. Wright, Andreas Ziegler: On the use of Harrell's C for clinical risk prediction via random survival forests. Expert Systems with Applications (63):450-459, 2016.
[35] Brent Longstaff, Sasank Reddy and Deborah Estrin: Improving activity classification for health applications on mobile devices using active and semi-supervised learning. 2010 4th International Conference on Pervasive Computing Technologies for Healthcare:1-7, 2010.
[36] Yonggang Lu, Ye Wei, Li Liu, Jun Zhong, Letian Sun, Ye Liu: Towards unsupervised physical activity recognition using smartphone accelerometers. Multimedia Tools and Applications 76(8):10701-10719, 2017.
[37] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore: Activity Recognition using Cell Phone Accelerometers. ACM SigKDD Explorations Newsletter 12(2):74-82, 2011.
[38] Nabil Alshurafa, Wenyao Xu, Jason J. Liu, Ming-Chun HuangBobak Mortazavi, Christian K. Roberts, Majid Sarrafzadeh: Designing a Robust Activity Recognition Framework for Health and Exergaming Using Wearable Sensors. IEEE Journal of Biomedical and Health Informatics 18(5), 2014.
[39] Feng Siwei: Sparsity in Machine Learning: An Information Selecting Perspective. Doctoral Dissertations, 2019.
[40] Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, Ian Lane: Semi-supervised convolutional neural networks for human activity recognition. 2017 IEEE International Conference on Big Data (Big Data), 2017.
[41] Maja Stikic, Kristof Van Laerhoven, Bernt Schiele: Exploring semi-supervised and active learning for activity recognition. 2008 12th IEEE International Symposium on Wearable Computers:81-88, 2008.
[42] Beth Logan, Jennifer Healey, Mattahai Philipose, Emmanuel Mungia Tapia, Stephen Intille: A Long-Term Evaluation of Sensing Modalities for Activity Recognition. Ubicomp 2007: Ubiquitous Computing:483-500, 2007.
[43] Francisco Javier Ordóñez and Daniel Roggen: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors 16(1):115, 2016.

[44] Shaojie Bai, J. Zico Kolter, Vladlen Koltun: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. Cornell University Computer Science Machine Learning, 2018.
[45] Kilian Forster, Daniel Roggen, Gerhard Troster: Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain? 2009 International Symposium on Wearable Computers:77-84, 2009.
[46] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore: Activity recognition using cell phone accelerometers. ACM SIGKDD Explorations Newsletter 12(2):74-82, 2011.
[47] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, Tapani Raiko: Semi-supervised learning with ladder networks. arXiv, Computer Science, Neural and Evolutionary Computing, 2015.
[48] Luciana C. Jatoba, Ulrich Grossmann, Chistophe Kunze, Jorg Ottenbacher, Wilhelm Stork: Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society:5250-5253, 2008.
[49] Uwe Maurer, Asim Smailagic, Daniel P. Siewiorek, Michael Deisher: Activity recognition and monitoring using multiple sensors on different body positions. International Workshop on Wearable and Implantable Body Sensor Networks:4-116, 2006.
[50] Ling Bao, Stephen S. Intille: Activity recognition from user-annotated acceleration data. Pervasive Computing:1-17, 2004.
[51] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, Michael L. Littman: Activity recognition from accelerometer data. IAAI'05: Proceedings of the 17th conference on Innovative applications of artificial intelligence 3:1541-1546, 2005.
[52] Illapha Cuba Gyllensten, Alberto G. Bonomi: Identifying Types of Physical Activity with a Single Accelerometer: Evaluating Laboratory-trained Algorithms in Daily Life. IEEE Transactions on Biomedical Engineering 58(9):2656-2663, 2011.
[53] Attal Ferhat, Mohammed Samer, Dedabrishvili Mariam, Chamroukhi Faicel, Oukhellou Latifa, Amirat Yacine: Physical Human Activity Recognition Using Wearable Sensors. Sensors 2015 15(12):31314-31338, 2015.
[54] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Forster, Gerhard Troster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, Jose del R. Millan: Collecting complex activity data sets in highly rich networked sensor environments. 2010 Seventh International Conference on Networked Sensing Systems (INSS):233-240, 2010.
[55] Davide Buffelli, Fabio Vandin: Attention-Based Deep Learning Framework for Human Activity Recognition with User Adaptation. arXiv, Computer Science, Machine Learning, 2020.
[56] Mi Zhang, Alexander A. Sawchuk: Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors. Proceedings of the 2012 ACM Conference on Ubiquitous Computing:1036-1042, 2012.

[57] Jue Wang, Zhibin Huang, Huanyuan Xu, Zilu Kang: Clustering Analysis of Human Behavior Based on Mobile Phone Sensor Data. Proceedings of the 2018 10th International Conference on Machine Learning and Computing:64-68, 2018.
[58] Zahraa Said Abdallah, Mohamed Medhat Gaber profile, Bala Srinivasan, Shonali Priyadarsini Krishnaswamy: Activity Recognition with Evolving Data Streams: A Review. ACM Computing Surveys 51(4), 2018.