Learning Algorithms for Physical Activity Monitoring

Scientific Seminar

submitted by
Dominic Angerer

NEUROSCIENTIFIC SYSTEM THEORY
Technische Universität München

Prof. Dr. Jörg Conradt

Supervisor: Dr. Cristian Axenie, M.Sc.
Final Submission: 06.07.2016

In your final hardback copy, replace this page with the signed exercise sheet.

Abstract

This report gives an introduction to the research field of physical activity monitoring based on sensor data from smartphones. Therefore, an overview of different approaches based on a number of current publications for the different stages of the process is given: Signal creation, preprocessing and feature extraction, classification, and their achieved performance. Further the C4.5 algorithm is briefly presented as a commonly used classifier. In the comparison based on the F_1 measure calculated from the results of the publications, the Ameva discretization and classification algorithm achieved best results. The approach behind the Ameva algorithm as a top down splitting algorithm and further as classification algorithm is briefly illustrated. The report then addresses the main challenges in the conclusion: the difficulty of comparing the various approaches solely based on their publications and the challenging investigation of feature extraction because of the lack of data for comparison. The most important part of the activity recognition process is concluded as the feature extraction which despite being desired to be done automatically, is still a manual step in many of the reviewed publications.

Contents

Chapter 1

Physical activity monitoring with mobile phones

1.1 Phones as simple daily monitoring tools

In the past years, the connection between movement and health has gained a lot of attention, both from concerned people as well as from researchers. However physical activity is often still measured by surveys in which the patients or clients have to enter their habits. Most of the physical exercise however is done unconsciously and thus filling out such a survey is quite a challenge to many people.
This could change significantly when everybody knows how much exercise he or she is doing on a daily basis. Further, it could become possible to correct bad work out practices or to help intervene habits causing typical health problems like an aching back from sitting too long. The I-Watch from Apple is already doing that and yet there are far more possibilities of these kind of recommendations.
Just imagine a personal health and movement counselor which gives you advice when and if you like it. This is the vision where the research in physical activity monitoring is aiming for. The topic has gained increasing attention in the past years and thus this report will give an overview about the current state of the field.

Essential for any activity recognition tool is that the user actually carries it around, the more the better for the overall monitoring. This makes the usage of smart phones beneficial. The phone accompanies the users everywhere. Many even take it with them in their beds when they go to sleep in order to have their alarm clock nearby. Even if users place their phones on a table when sitting at the desk or during a meal, when they move, the probability is high that their phone is with them. Taking that into account a phone is an ideal long term monitoring tool and researchers like Pärkka et al. have presented first results showing new insights even on long-term-stress by using data gathered through smart-phones (by [Pär11]).
Not only is it convenient to use smart-phones for activity recognition because of their availability but also in terms of how well they are equipped with sensors. This

report will give an overview over recent publications covering the topic of activity recognition. Thereby the technical aspects like the used sensors, feature extraction methods, and classification methods will be analyzed. Later a closer look at what aspects are essential for user-adoption and finally a conclusion is given where to start when conducting further research in this field.

1.2 The task and the approach

The original task of this report had been to compare two classical and two machine learning approaches which are promising or currently applied in the field. After an initial investigation of the relevant publications it became clear that the classical, non-machine learning approach is of no significance to the research field.
Additionally the goal of the report had been to give an introduction to the field and provide recommendations on how to enter the field with new approaches. Due to that throughout this report there will be many quantitative comparisons between the various publications. The report was defined as being only a literature review, without any own experiments conducted. That is why the comparisons are not evaluated by own experiments.
The report starts by entering the field with giving an overview of the different activities being monitored. Then the available and potentially available sensors in smartphones are presented as well as how there signals are preprocessed until feature vectors are gained. The main part is concluded with both an overview of common used classifiers and the description of the C4.5 and the Ameva algorithm.
In the end the main challenges as well as recommendations for further investigation of the field of physical activity recognition is given.

Chapter 2

Common classification process

In order to detect the activities correctly, the classification process is essential for activity recognition. The whole process can be broken down into the following major steps:

1. Select activities to be monitored
2. Capture data through sensors
3. Extract features sets from data stream
4. Classify the sets as the respective activity

Morillo et. al. has considered additionally to store the classified data on a server for additional analysis and verification(by [MGARdL15]). However this does not affect the main classification process as such and therefore only the first three steps are discussed in the sections to come.

2.1 Typically monitored activities

There are a number of different monitored activities of interest to a user. Activity monitoring is often done under the aspect of estimating whether the user is active enough and thus it is related with health. In figure 2.1 there is an overview given about most common activities being monitored. Walking, Running, Cycling and ascending or descending stairs are leading. However there are also more specific excercise movements like jumping.
Among the activities there are also some which seem to be difficult to distinguish like lying, standing, or sitting. Considering a phone tracking mostly relative movement data, the patterns created by those can be assumed to be similar.
Also the bare number of activities varies from five to eight or more in some special cases[1]. With a higher number of activities, the chances rise that there are very similar ones among them. That then again raises higher demands towards the

[1]In the appendix A.14 an overview of the variation in the number of activies is given

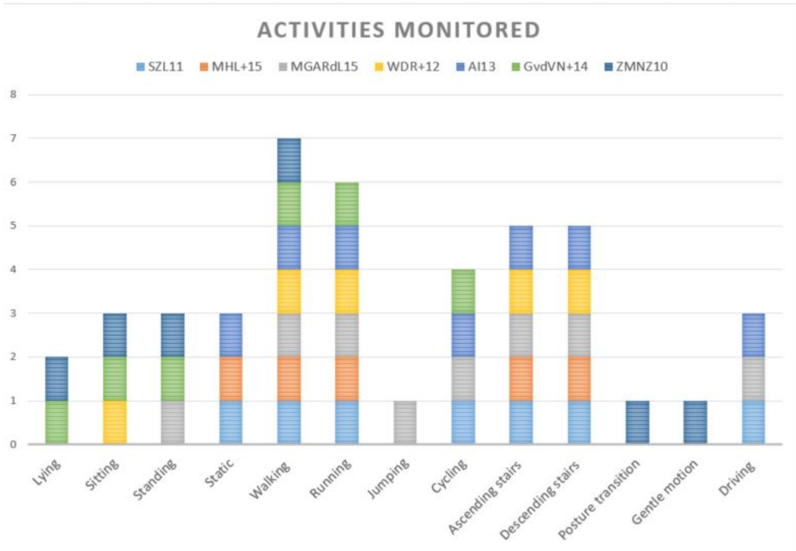


Figure 2.1: Overview of the activities monitored in the reviewed literature

classfication process.
Not only activities differ, but also the phones and their capabilities for measuring and recording data as will be seen in the next section.

2.2 Available and used sensors

The first major step is generating data from the various sensors. The compared publications offer a wide range in number and available types of sensors. Even if as of today many of those sensors are not yet available in common smart-phones, there is a tendency towards more sensors in phones and smart wearables.
Figure 2.2 gives a rough overview of the range of sensors used to create the data for the further analysis. As can be seen, Pärkka et. al covers a wide range of sensors in his study. Among those, there are many sensors not typically found in smart-phones like pulse– or respiratory sensors. However, in this study it is concluded that the accelerometer, magnetometer, and the gyroscope are the most useful ones for the task of activity recognition.(by [Pär11])
Additionally, there are some earlier studies like the one from Györbíró et. al using the wifi or bluetooth module of smart-phones as additional sensor input.
Especially, the recent works of Wu et. al[2] try to focus on sensor data generated only by the accelerometer(see [WDR+12]). This is mainly done because of the significantly lower power consumption of in the background running accelerometer sensor versus a magnetometer, GPS or other sensors.

[2]A comparison of classifcation results achived based on accelerometer signal only and added gyroscope signal was done by Wu et. al and can be found in the appendix A.13

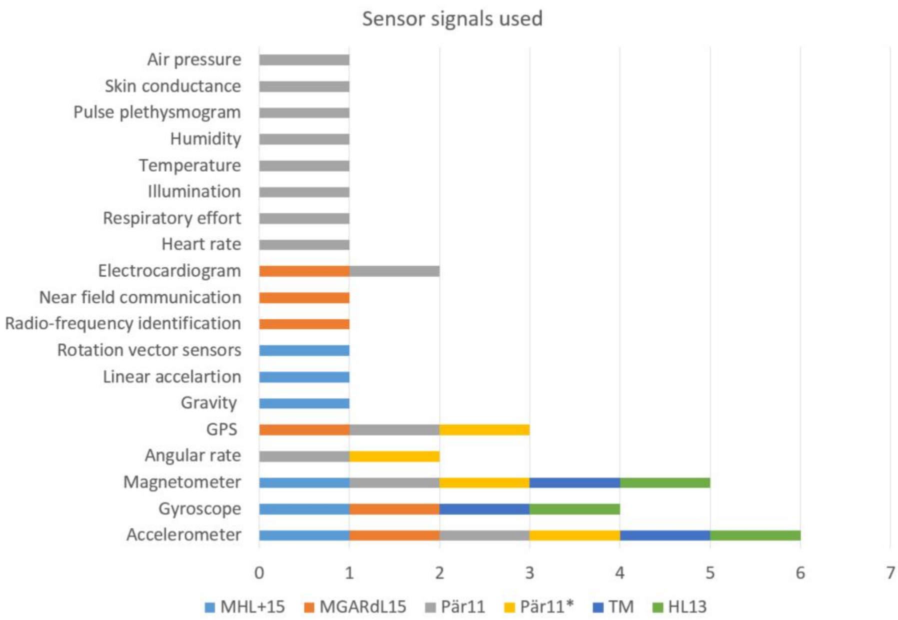


Figure 2.2: Overview of exemplary sensor signals used in reviewed literature. Pärkka et al. has used many different sensor signals in his experiments. Pär11* shows the ones he considered most useful. [Pär11]

Wu et. al have shown that adding the sensor input of the rotation rates from the gyroscope to classification system which previously only was fed by an acceleration signal resulted in performance increases of up to 13 % depending on the observed activity[3]. However, the gyroscope has a higher battery consumptions as it is an active sensor versus the accelerometer being a passive one, thus its usage depends on the requirements. The signals created by the various possible sensors however are subject to constant noise and the data-stream coming directly from the sensors is too dense to be used for classification. Therefore, the significant information has to be extracted first. This is done in the step of feature selection or preprocessing in general.

2.3 Feature selection and preprocessing

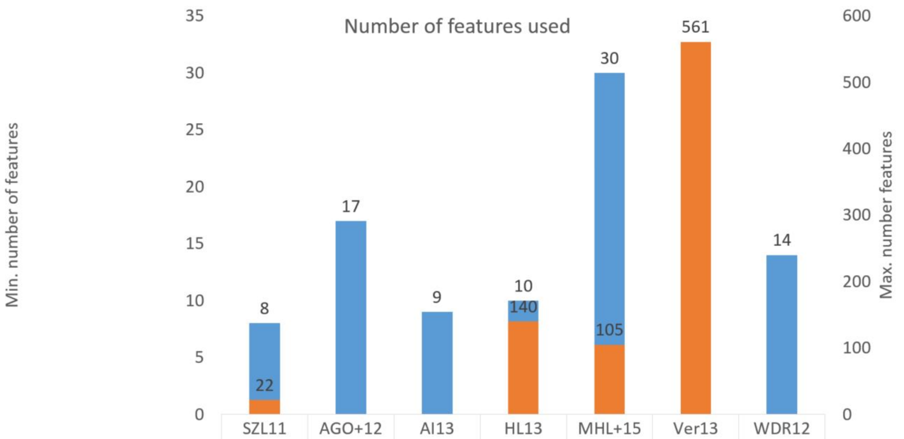
The feature selection is the most crucial step in the whole activity recognition process. On the one hand, the more features have to be calculated the more processing power is needed on the phone and the less likely is it to have the whole recognition running on the phone itself. On the other hand, more calculated features provide more information and are more likely to provide the crucial information for distinguishing between very similar activities like standing or sitting.
Also the number of features is important. The course of dimensionality(see [Wik16c])

[3]The extracted table of results is include in appendix A.13

says that in higher dimensional spaces it is more difficult to distinguish patterns. Otherwise there are a minimum number of features required to distinguish between the different activities.

2.3.1 Number of features

The number of calculated features varies from eight by Sun et. al ([SZL11]) to more than 500 by Verleysen et. al ([Ver13]) as can be seen in figure 2.3. It has to be mentioned here that in some publications the number of features used to detect certain activities vary from the numbers used for other activities



	SZL11	AGO+12	AI13	HL13	MHL+15	Ver13	WDR12
min number of features	8	17	9	10	30		14
max number of features	22			140	105	561	

Figure 2.3: Overview over different amounts of features used. Only numbers specifically mentioned by the authors are shown.

2.3.2 Window length and preprocessing

A single time stamp of the original dataset does not provide enough information to be used for classification. The reason for this is mostly that the recorded data is relative and gets its context only as a timeseries. A common way to deal with this is to cut the timeseries data in a number of windows specified by their length and the overlapping part between two following windows. In figure 2.4 the different published window lengths can be seen. The overlap of the windows ranges from 25 to 50 %[4]

[4]An overview is given in the table in appendix A.1

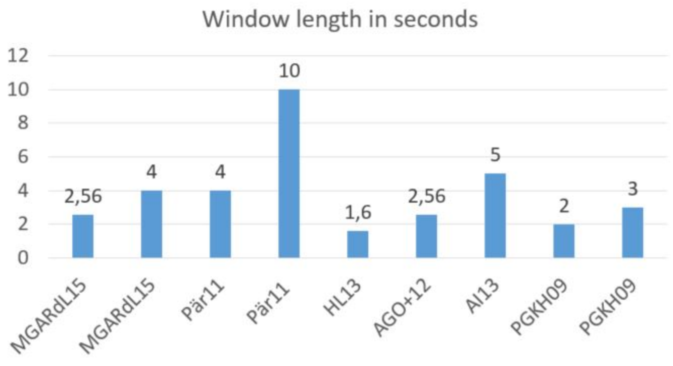


Figure 2.4: Histogram showing the different options for window length appearing in the publications

The window length is also an important factor when it comes to calculating the delay between the occurrence of an activity and the detection of it. This has two major implications. First, the cycles within one activity should be complete within one window to get a comparable pattern. This means single steps for walking for example. Second, the duration of the total activity blocks has to be equal or longer than the sampling window length.(by [MGARdL15])
Zhang et. al suggests additionally an early classification in preprocessing which determines whether movement occures or the phone is static and adapts both the sensor recording intervals as well as the preprocessing calculations carried. The main achievement of this addition is a reduced energy consumption and as such a longer lasting battery. (by [ZMNZ10])
The transformations of the data series to the frequency space like the Fouriertransformation are often also considered preprocessing methods. These will be discussed together with the feature calculation in the next section.
Tundo et. al uses a quarternion-based rotation matrice in order to match the smartphone accelerometer components to a given 3-D reference to get the acceleration vectors aligned with the human movement (in [TM]).
Sun et. al ([SZL11]) have investigated the impact of window length on the achieved accuracy for different classification methods. The result from there is that longer windows– up to 40 seconds – achieve better results[5]. However as described above the window length also determines the minimum duration of an activity and its cycle time and therefore a compromise has to be met in practical applications.

2.3.3 Different features used

The features which are computed for the later use of classification can be grouped into four different sets (in [TM]):

[5]An excerpt of the chart is included in the Appendix A.12

Timeseries domain and heuristic features

These features are directly calculated from the time samples created through cutting windows of the sensor data. Popular features include the mean, median, skewness, root mean square, and more. A histogram of the appreance of the various calculated features in the reviewed publications can be seen in figure 2.5.

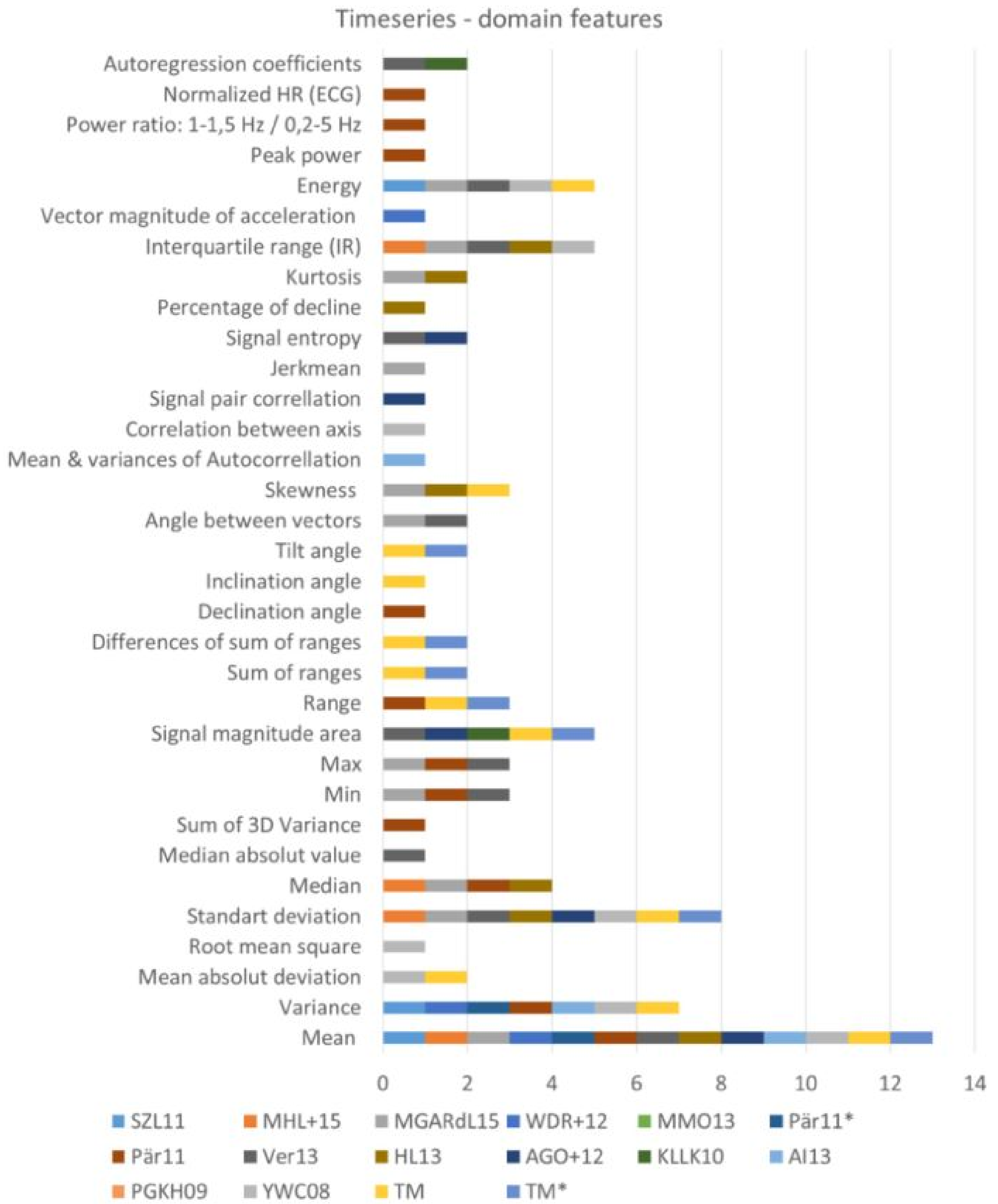


Figure 2.5: Features calculated based on timeseries domain and named as they were reported in the respective publications.

Frequency domain features

The timeseries is transformed using Fourier transform or Fast Fourier transform (in [WDR+12]) and then the features are derived from the resulted data. Figure 2.6 gives a brief overview of the calculated features based on the frequency domain in this reports' considered publications.

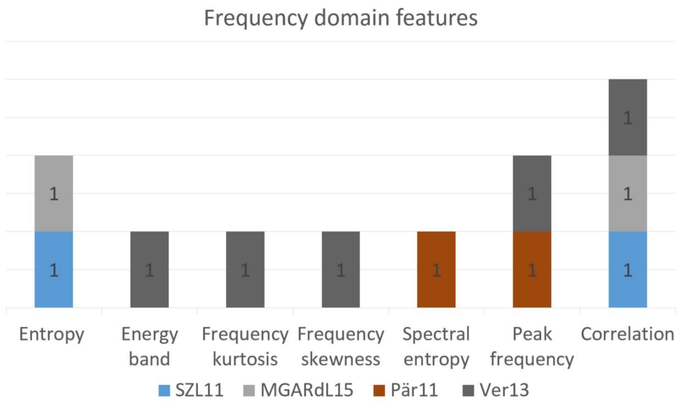


Figure 2.6: Features calculated based on previous transformation to the frequency domain.

Discrete wavelet features (DWT)

The wavelet analysis is a hybrid version of time and frequency analysis. This mainly means it captures both frequency and location information. The method is used by Mitchell et. al (in [MMO13]) and by Preece et. al (in [PGKH09]).
The most common version is the Haar discrete Wavelet transformation. It pairs the input values up while storing the difference and passing the sum. This is repeated recursively until a final sum and $2^n - 1$ differences are achieved. The main advantage is a result of frequencies of the input at different scales as well as the temporal content by the cost of O(n) operations. The DWT is not time-invariant which makes it very sensitive to the alignment of the signal in time and can be called its main disadvantage.(by [Wik16e])
Mitchell et. al additional states the lack of widely accepted method of picking the most suitable mother wavelet for a given application and concludes the Daubechies as the mother wavelet achieving the best result among many others (in [MMO13]). Preece et. al used the Daubechies DWT and showed that features derived from the time or frequency domain outperform the ones resulted from the DWT[6]. However, they also mention that this might be due to the chosen activities. (by [PGKH09])

[6]The extracted table showing a comparison of specificity and sensitivity of both Wavelet and Timeseries feature set can be found in Appendix A.6

Additional features and factors

All those previously mentioned feature can be applied to the various sensor signals discussed in section 2.2. However in most of the publications special attention is given to the acceleration signal and extended to gyroscope. That results in the feature calculation being basically applied to those two signal types.
Additionally to the already mentioned features, Tundo et. al, proposes to use two additional variables covering the predicted state and the change of state. This results in providing more context to the feature vector and can be an explanation for the possibility to have a one second sampling window as shown in section 2.3.2. (by [TM])
Both the Frequency domain analysis and the Discrete wavelet analysis require a transformation to the frequency space. This needs additional computing power and requires bigger sampling windows. As Anjum et. al stated, in order to minimize the number of necessary computations on the mobile device, it is preferred to use features which can be directly computed from the timeseries (by [AI13]).

2.3.4 Feature reduction methods

He et. al have used a combined algorithm of Fisher's discriminant ratio (FDR) criterion and the so called J3 criterion in order to choose 10 out of 140 calculated features which promised to maximize the distinguishablility of the different activities. They used this approach for different classifiers and came up with different features for different classifiers.(by [HL13])
Additionally Yang et. al have used the Constraint Principal Component Analysis to improve the usability of the features for the classification (in [YWC08]).
For this purpose Khan et. al go a step further and apply Linear– and in their final algorithm Kernel Discriminant Analysis (in [KLLK10]). Anguita et. al instead uses Ambient Intelligence (AmI) to choose which features trigger a classification (in [AGO+12]).

2.3.5 Most often calculated features table with formula

Although the variety of features being used by the various authors is considerably large as can be seen in figure 2.5, the features most often calculated can be distinguished. The mean of a given window as well as the variance or standard deviation are almost used by every quoted author. Following are the Energy, the Interquartile range, and the Signal Magnitude Area and all of them are listed in table 2.1. the example below a fixed column width is established.

Feature	Formula	Publication or source
Mean	$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y$	[TM]
Variance	$\sigma_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$	[TM]
Energy	$E = \frac{1}{n}\sum_{i=1}^{n} y^2$	[Ver13]
Interquartile range	$IR = Q_3 - Q_1$, where Q_3, Q_1 is the $75th$ and the $25th$ percentiles over the window, respectively.	[HL13]
Signal Magnitude Area	$SMA = \frac{1}{t}\Big(\int_0^t \lvert A_x(\tau)\rvert d\tau + \int_0^t \lvert A_y(\tau)\rvert d\tau + \int_0^t \lvert A_z(\tau)\rvert d\tau\Big)$	[TM]

Table 2.1: Calculation formulas used to derive the features.

Figure 2.5, from the previous section, also illustrates that apart from those few features which are used in multiple publications many of the authors came up with their own feature calculations. This results in many feature calculations appearing only in one or very few publications.

2.3.6 Further notes on feature calculation and selection

It also has to be mentioned that according to Preece et. al the importance of processing and selecting the features increases if only one sensor is used(by [PGKH09]). They also suggest to include a biomechanical model to differentiate better between the different postures and posture transitions to compensate for this.
When it comes to saving computational costs for calculating the feature sets, Anguita et. al suggest to use fix point arithmetic (in [AGO$^+$12]).

2.4 Classification methods

After a distinctive set of features has been selected, a classification method is chosen to learn the patterns of the various activities. This method then also detects the activities.
Figure 2.7 gives an overview of the most frequently used methods. In general, the classification methods are not explained in detail but most often only referred to by their names. Especially the artificial neural network classifiers are not specified more in detail in the publications.
The histogram in figure 2.7 shows the significance of decision tree algorithms. Naive-Bayes, J48 or C4.5 as well as hybrid-decision-tree and decision-tree itself, together they appear most often of all the other methods in publications. Additional to

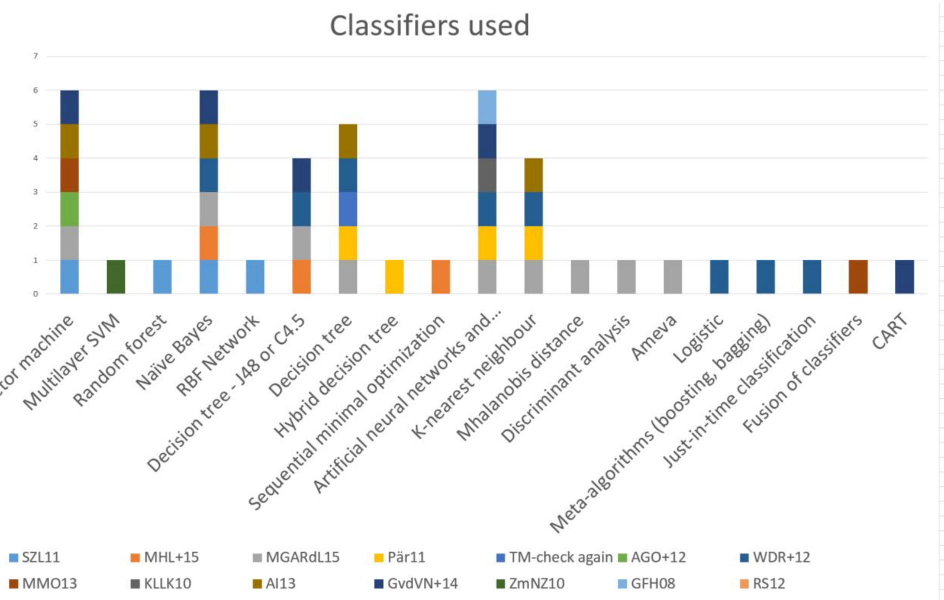


Figure 2.7: Overview of classifiers used in reviewed literature. Artificial networks contain also Multi-Layer-Perceptron and Feedforward implementations.

decision-trees, Support Vector Machines play an important roles as well. They are followed by artificial neural networks and K-Nearest-Neighbor methods.

2.5 Reported results

Despite the defined test datasets which are available, many authors still use their own test data created by specific experiments. Additional the reviewed publications include various different performance criteria. That makes a comparison difficult. Some authors have shown the confusion matrices resulting from their experiments. Based on those, the F_1-measure has been calculated according to formula 2.2. TP refers to the term True positives, FP false positives, and TN and FN to True negatives and False negatives respecting.[Wik16f][Wik16b]

$$F_1 = 2\,\frac{precision * recall}{precision + recall} \tag{2.1}$$

$$F_1 = \frac{2\,TP}{2\,TP + FP + FN} \tag{2.2}$$

The average F1 measure together with the range of one standard deviation for an activity in the respecting publication can be seen in figure 2.8. There are significant differences between the same classification methods carried out by different authors. Not only the mean of the F_1 differs (as shown in the different SVM results) but also the variance(J48 and C4.5)

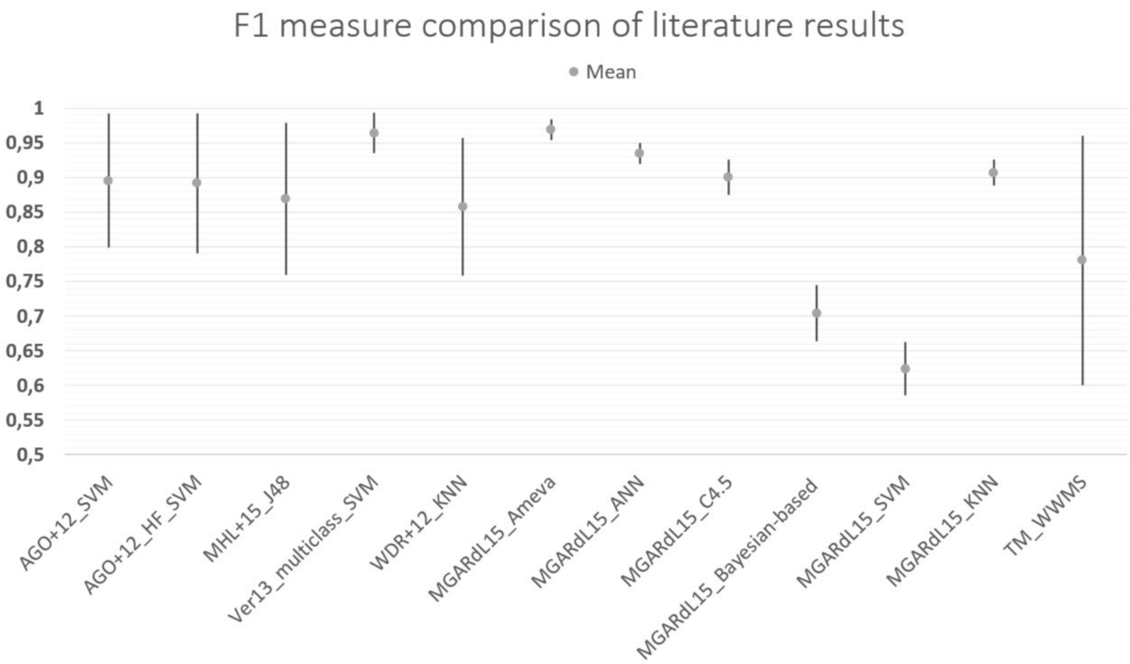


Figure 2.8: F_1-measure calculated based on reported confusion matrix. The confusion matrices can be found in the Appendix under section A. The first part of the label corresponds to the publication, the second to the classification algorithm being used.

Also Pärkkä et. al concludes that the selection of the feature vector is more important than the selection of classifiers. Thus it is not possible to conclude one classifier which outperforms all others based on literature results. How the feature vector was constructed and how the classifiers are implemented in detail is incomplete in most publications.

2.5.1 Most often used approaches briefly explained

Since most of the classifiers belong to one of these three groups, they are briefly explained:

- Decision-tree:
  In a decision-tree, the leaves correspond to the labels of the different classes and new samples are assigned to a leave by taking features and their probability connected to the labels into account.(by [Wik16d])
- Support Vector Machine (SVM):
  SVMs are trying to find a function that separates the different classes in the features space. The separation is done one class against the rest since the space is always divided into two subspaces.

- Artificial Neural Networks (ANN):
  A network of nodes with processing or activation functions and weights which can be adjusted to learn to distinguish between patterns.

As the ANNs can be implemented in a multitude of different realizations and these approaches lack a detailed description of how they are done, they are not further discussed.
Among the decision-tree algorithms the C4.5 get special attention (by [MHL+15]) and the Ameva algorithm used by Morillo et. al is said to have a low computational cost by combining feature extraction and classification.(by [MGARdL15])

C4.5 or J48

The C4.5 has been developed by Quilan et. al based on the ID3 algorithm. It creates a decsion tree which then can be used for statistical classification. As splitting criteria the normalized information gain is used which corresponds to the difference in entropy. The J48[7] is an open source Java implementation and available in the Weka data mining tool.[Wik16a][WKQ+08]
The C4.5 was further developed into the C5.0 which among other advantages supports boosting, produces smaller decision trees, and is significantly faster by using less memory. However this algorithm has not been used in the reviewed activity recognition publications and it promises further perfomance improvements.[Wik16a]
Miao et. al have that a 10-fold J48 outperforms a naive Bayes as well as Sequential minimal optimization. Additionally they emphasized the fast computing time and pointed up that the process of building the model, although slower than the naive Bayes, is capable for real time applications.[MHL+15][8]
In the comparison done by Morillo et. al ([MGARdL15]) the C4.5 however only achieves average classification results among the compared classifiers. Although the C4.5 ranked second when it came to battery consumption, their suggested Ameva algorithm outperfomed the C4.5 in all measured scales(Accuracy, Recall, Specificity, Precision, F_1 Measure, Battery lifetime).
Also Wu et. al ([WDR+12] claims that the performance of the J48 ranks behind k-nearest neighbor or multilayer perceptrons, e.g. artificial neural networks. Wu et. al also carried out comparison with boosting and bagging algorithms and could show that they perform better as well. So one could assume that the c5.0 should perform better.

Ameva

The Ameva algorithm[9] was first published by Gonzalez-Abril et. al([GACVO09]) as a discretization algorithm for working with supervised learning algorithms. It is

[7] A brief pseudo code can be found in appendix B.1
[8] Their results can be found in the appendix A.4
[9] A brief pseudocode can be found in appendix **??**

based on Chi-square statistics and produces a minimal number of discrete intervals. Its special features include that the number of intervals is automatically detected and does not have to be defined by the user. The Ameva algorithm is a splitting algorigthm. It starts with a number of bins and divides the one with the worst Ameva criterion by splitting it in two. The Ameva crition is thereby given as in equation 2.3 with the training data set $X = \{x_1, \dots, x_N\}$ of continuous attribute $\mathscr{X}$ and each x_i only belonging to one class l denoted by $\mathscr{C} = \{C_1, \dots, C_l\}$.

$$Ameva(k) = \frac{\mathscr{X}^2(k)}{k(l-1)};\ for\ k, l \geq\ 2 \tag{2.3}$$

$$with\ \mathscr{X}^2(k) =\ N\left(-1 + \sum_{i=1}^{l}\sum_{j=1}^{k}\frac{n_{ij}^2}{n_{i.}n_{j.}}\right) \tag{2.4}$$

In their publication Morillo et. al ([MGARdL15]) have adapted the original Ameva algorithm and turned it to an classifier. In contrast to more traditional approaches the classification is learned therefore on discrete variables instead of continuous ones.

1. Original Ameva is applied in order to discreticize the variables like presented above.
2. The generated intervals of the training set are used to create statistics and then associated with the activity labels.
3. The activity-interval matrix is obtained. It consists of relative probabilities, is three dimensional, and can be seen as the likelihood that a statistic based on a given value corresponds to a certain activity.
4. The new data is classified based on a majority voting system based on the activity-interval matrix.

This reduces the computational cost of the classification process. Additionally Morillo et. al included a dynamic sample rate to get below the 50 Hz for activities which do not require high sample rates, like sitting and lying. For walking and running the sample rate is adjusted to 50 Hz automatically. This reduces additionally the computational load and thus decreases battery consumption.
All in all they concluded that their developed algorithm called Ameva achieved best results in classification by at the same time very low computational demands.

2.6 Important attributes for user adoption

Regardless of the various option for technical realization of activity recognition, it is important that the developed tool is used by customers. In the reviewed publications there had been a number of statements which are encouraged to be taken into account when further developing tools for activity recognition.

Sun et. al ([SZL11]) points out that the tools as to comply with the people's usage habits. That means people store their phones in their pockets but they do not want to spend special attention to the exact positioning or orientation of the phone in the pocket. As conclusion the tool has to be able to work regardless of its exact position. They also mention the necessity of only monitoring the activities which are actually wanted in order to save battery power.
Morillo et. al ([MGARdL15]) focus on the battery consumption, as the time until the battery has to be recharged is one of the strongest arguments for buying a phone. They present dynamic sampling rates from 32 to $50Hz$ depending on preestimation of what activity is monitored at the moment.
Zhang et. al ([ZMNZ10]) go a step further and emphasis the power save potential of offline classification in contrast to classification completely or partly on a server. Both the necessary computation as well as the kind of sensors used for it determine the needed energy. Morillo et. al ([MGARdL15]) could show that their algorithm called Ameva outperforms other classification methods by more than 20 % in terms of battery consumption.[10].

[10]The test results of Morillo et. al are included in appendix A.10

Chapter 3

Summary and conclusion

3.1 Summary

This report started out by giving an overview of commonly monitored activities(in section 2.1) like: standing, walking, running, cycling, and walking stairs up and down. There it became visible that especially the sport and health related activities: walking, running and cycling got attention. In addition, activities like sitting or standing are very similar in their signal patterns and thus they are often ignored.
The report went further to discuss the available sensors(in section 2.2) in smartphones and also sensors which could be found in smartphones or smartgear in general in the future. There it became clear that it is of advantage to use passive sensors like the accelerometer in order to save battery and many publications are trying to follow that direction. However the additional usage of the gyroscope adds up to 13 % in accuracy and thus cannot easy be neglected.
In section 2.3 an overview about calculated features based on the sensor signals was given. There it became clear that many of the publications do not offer clear enough information to reproduce and evaluate their experiments and suggested methods directly. The feature selection is the most important and the most critical step in order to get good classification results.
The feature extraction and pre-processing description was followed by the classification itself(in section 2.4). The charts show a number of different classifiers being used in the reviewed publications. This section also shows the results the different approaches achieved comparing them with the F_1 measure. The best results are achieved by the Ameva algorithm published by Morillo et. al and thus this algorithm together with the commonly known C4.5 algorithm are described more in detail.
In the end there is a short summary of important aspects for the success of an activity monitoring system in which the significance of its battery consumption is pointed out.

3.2 Conclusion

There were two main aspects for the process of activity monitoring showing up during the research. The first one was the importance of the feature selection. This is often done manually and requires detailed knowledge about the patterns but even more a lot of trial and error. The more recent publications have shown a raising interest in getting this feature vector composed, selected and reduced automatically. Some went so far to reduce the feature vector differently depending on each observed activity. So far this seems impractical to me but it clearly shows that the selection of the feature set is currently in the focus of research.
Morillo et. al showed additionally that even the feature selection and the sampling already have potential to save energy when done in an addaptive, intelligent way.
The second observation was the lack of transparency in the conducted research leading to incomparable results throughout the various publications. Only relatively few published a confusion matrices, thus evaluating their results becomes difficult. Additionally the methodical approach from feature selection to the configuration of the classifiers have been described only sparse. In my opinion a comparison of the suggested approaches based on few benchmark datasets would be necessary to be able to create a proper ranking of the different approaches.

List of Figures

Bibliography

[AGO+12] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, José Bravo, Ramón Hervás, and Marcela Rodríguez, editors, *Ambient Assisted Living and Home Care*, volume 7657 of *Lecture Notes in Computer Science*, pages 216–223. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[AI13] A. Anjum and M. U. Ilyas. Activity recognition using smartphone sensors. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 914–919, Jan 2013.

[BRGPR+14] Judit Bort-Roig, Nicholas D. Gilson, Anna Puig-Ribera, Ruth S. Contreras, and Stewart G. Trost. Measuring and influencing physical activity with smartphone technology: A systematic review. *Sports Medicine*, 44(5):671–686, 2014.

[GACVO09] L Gonzalez-Abril, Francisco Javier Cuberos, Francisco Velasco, and Juan Antonio Ortega. Ameva: An autonomous discretization algorithm. *Expert Systems with Applications*, 36(3):5327–5332, 2009.

[GFH08] Norbert Győrbíró, Ákos Fábián, and Gergely Hományi. An activity recognition system for mobile phones. *Mobile Networks and Applications*, 14(1):82–91, 2008.

[GvdVN+14] John J. Guiry, Pepijn van de Ven, John Nelson, Lisanne Warmerdam, and Heleen Riper. Activity recognition with smartphone support. *Medical Engineering & Physics*, 36(6):670 – 675, 2014.

[HKK+12] David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz,

C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, José Bravo, Ramón Hervás, and Marcela Rodríguez, editors. *Ambient Assisted Living and Home Care*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[HL13] Yi He and Ye Li. Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. *International Journal of Distributed Sensor Networks*, 2013(3):1–10, 2013.

[KDVM12] Morwenna Kirwan, Mitch J. Duncan, Corneel Vandelanotte, and W. Kerry Mummery. Using smartphone technology to monitor physical activity in the 10,000 steps program: A matched case–control trial. *Journal of Medical Internet Research*, 14(2):e55, 2012.

[KLLK10] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *2010 5th International Conference on Future Information Technology*, pages 1–6, May 2010.

[LML+10] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, Sept 2010.

[MCLC04] Merryn J Mathie, Adelle C F Coster, Nigel H Lovell, and Branko G Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological Measurement*, 25(2):R1, 2004.

[MGARdL15] Luis Morillo, Luis Gonzalez-Abril, Juan Ramirez, and Miguel de La Concepcion. Low energy physical activity recognition system on smartphones. *Sensors*, 15(3):5163–5196, 2015.

[MHL+15] Fen Miao, Yi He, Jinlei Liu, Ye Li, and Idowu Ayoola. Identifying typical physical activity on smartphone with varying positions and orientations. *BioMedical Engineering OnLine*, 14(1):1–14, 2015.

[MMO13] Edmond Mitchell, David Monaghan, and Noel O'Connor. Classification of sporting activities using smartphone accelerometers. *Sensors*, 13(4):5317–5337, 2013.

[Pär11] Juha Pärkkä. *Analysis of personal health monitoring data for physical activity recognition and assessment of energy expenditure, mental load and stress*, volume 765 of *VTT publications, 1235-0621*. VTT, Espoo, 2011.

[PGKH09] Stephen J. Preece, John Yannis Goulermas, Laurence P. J. Kenney, and David Howard. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE transactions on bio-medical engineering*, 56(3):871–879, 2009.

[RS12] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '12, pages 40:1–40:8, New York, NY, USA, 2012. ACM.

[SZL11] Lin Sun, Daqing Zhang, and Nan Li. Physical activity monitoring with mobile phones. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Bessam Abdulrazak, Sylvain Giroux, Bruno Bouchard, Hélène Pigot, and Mounir Mokhtari, editors, *Toward Useful Services for Elderly and People with Disabilities*, volume 6719 of *Lecture Notes in Computer Science*, pages 104–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[TM] Tundo and Marco. Development of a human activity recognition system using inertial measurement unit sensors on a smartphone.

[Ver13] Michel Verleysen. *ESANN 2013: 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning : Bruges, Belgium, April 24-25-26, 2013 : proceedings.* Ciaco - i6doc.com, Louvain-la-Neuve, 2013.

[WDR+12] Wanmin Wu, Sanjoy Dasgupta, Ernesto E. Ramirez, Carlyn Peterson, and Gregory J. Norman. Classification accuracies of physical activities using smartphone motion sensors. *Journal of Medical Internet Research*, 14(5):e130, 2012.

[Wik16a] Wikipedia. C4.5 and j48 algorithms - Wikipedia, the free encyclopedia, 2016. [Online; accessed 15-Juni-2016]. URL: `https://en.wikipedia.org/wiki/C4.5_algorithm`.

[Wik16b] Wikipedia. Confusion matrix - Wikipedia, the free encyclopedia, 2016. [Online; accessed 10-May-2016]. URL: `https://en.wikipedia.org/wiki/Confusion_matrix`.

[Wik16c] Wikipedia. Course-of-dimensions - Wikipedia, the free encyclopedia, 2016. [Online; accessed 5-July-2016]. URL: `https://en.wikipedia.org/wiki/Curse_of_dimensionality`.

[Wik16d] Wikipedia. Decision tree - Wikipedia, the free encyclopedia, 2016. [Online; accessed 19-Juni-2016]. URL: https://en.wikipedia.org/wiki/Decision_tree_learning.

[Wik16e] Wikipedia. Discrete wavelet transform - Wikipedia, the free encyclopedia, 2016. [Online; accessed 19-Juni-2016]. URL: https://en.wikipedia.org/wiki/Discrete_wavelet_transform.

[Wik16f] Wikipedia. F_1 - score - Wikipedia, the free encyclopedia, 2016. [Online; accessed 10-May-2016]. URL: https://en.wikipedia.org/wiki/F1_score.

[WKQ+08] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[YWC08] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognition Letters*, 29(16):2213–2220, 2008.

[ZMNZ10] S. Zhang, P. McCullagh, C. Nugent, and H. Zheng. Activity monitoring using a smart phone's accelerometer with hierarchical classification. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 158–163, July 2010.

Appendices

Appendix A

Confusion matrices and other evalutation data from publications

Method	MC-SVM							MC-HF-SVM $k = 8$ bits						
Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	*Recall %*	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	*Recall %*
Walking	**109**	0	5	0	0	0	*95.6*	**109**	2	3	0	0	0	*95.6*
Upstairs	1	**95**	40	0	0	0	*69.8*	1	**98**	37	0	0	0	*72.1*
Downstairs	15	9	**119**	0	0	0	*83.2*	15	14	**114**	0	0	0	*79.7*
Standing	0	5	0	**132**	5	0	*93.0*	0	5	0	**131**	6	0	*92.2*
Sitting	0	0	0	4	**108**	0	*96.4*	0	1	0	3	**108**	0	*96.4*
Laying	0	0	0	0	0	**142**	*100*	0	0	0	0	0	**142**	*100*
Precision %	*87.2*	*87.2*	*72.6*	*97.1*	*95.6*	*100*	***89.3***	*87.2*	*81.7*	*74.0*	*97.8*	*94.7*	*100*	***89.0***

Figure A.1: Confusion matrices gained by Anguita et. al. The MC-SVM is calculated with floating point SVM. The MC-HF-SVM is working based on 8 bit integers. [AGO+12]

Table B1. Confusion matrix and performance values for the Ameva-based classification system.

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-Measure
Walk	7200	10	10	61	80	40	37	89	7527	95.88%
Jump	10	7304	0	152	0	0	21	8	7495	96.89%
Immobile	0	0	7256	0	0	0	0	142	7398	98.86%
Run	16	128	0	7298	0	7	40	8	7497	97.01%
Up	89	70	0	8	7273	195	2	2	7639	95.02%
Down	96	0	0	23	316	7252	4	5	7696	95.48%
Cycle	0	0	0	0	0	0	7302	1	7303	99.22%
Drive	80	70	16	7	0	0	10	7215	7398	97.05%
Accumulated	7491	7582	7282	7549	7669	7494	7416	7470	59,953	96.93%

Table B2. Confusion matrix and performance values for the ANN-based classification system.

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-Measure
Walk	6930	46	41	117	107	79	86	121	7527	91.93%
Jump	76	7124	16	187	13	8	46	25	7495	93.,83%
Immobile	47	9	6970	4	63	46	13	246	7398	95.37%
Run	44	221	3	7060	39	43	59	28	7497	94.15%
Up	126	89	17	25	7091	256	19	16	7639	91.75%
Down	103	4	7	27	415	7103	16	21	7696	92.84%
Cycle	36	32	43	35	75	56	6928	98	7303	95.59%
Drive	187	165	121	45	15	15	25	6825	7398	92.37%
Accumulated	7549	7690	7218	7500	7818	7606	7192	7380	59,953	93.48%

Table B3. Confusion matrix and performance values for the C4.5-based classification system.

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-Measure
Walk	6210	78	98	318	298	96	92	337	7527	85.11%
Jump	113	6932	54	245	36	17	67	31	7495	90.97%
Immobile	59	15	6790	23	83	72	59	297	7398	92.57%
Run	54	294	23	6896	64	57	71	38	7497	90.74%
Up	160	113	51	34	6943	298	21	19	7639	88.88%
Down	187	18	15	53	427	6917	43	36	7696	90.60%
Cycle	64	79	54	66	96	74	6716	154	7303	93.10%
Drive	219	216	187	68	38	43	56	6571	7398	88.31%
Accumulated	7066	7745	7272	7703	7985	7574	7125	7483	59,953	90.03%

Table B4. Confusion matrix and performance values for the Bayesian-based classification system

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-measure
Walk	5067	187	218	573	421	329	198	534	7527	66.15%
Jump	741	5180	113	1023	57	124	216	41	7495	65.18%
Immobile	115	87	5014	288	356	298	243	997	7398	74.31%
Run	78	1520	46	5290	155	96	245	67	7497	69.62%
Up	613	509	79	77	5521	708	56	76	7639	69.03%
Down	470	27	43	76	1309	5641	76	54	7696	73.88%
Cycle	105	127	145	227	451	300	5216	732	7303	76.38%
Drive	603	762	438	145	87	79	105	5179	7398	68.70%
Accumulated	7792	8399	6096	7699	8357	7575	6355	7680	59,953	70.41%

Table B5. Confusion matrix and performance values for the SVM-based classification system.

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-Measure
Walk	4562	210	289	658	513	401	201	693	7527	57.88%
Jump	819	4716	214	1142	64	176	275	89	7495	58.77%
Immobile	156	115	4663	342	398	341	296	1087	7398	68.58%
Run	89	1598	85	4982	174	158	335	76	7497	64.62%
Up	769	654	97	85	4728	1127	82	97	7639	59.83%
Down	636	53	68	84	1614	5092	87	62	7696	65.53%
Cycle	354	297	241	361	580	467	4189	814	7303	64.34%
Drive	852	910	543	268	96	83	254	4392	7398	59.72%
Accumulated	8237	8553	6200	7922	8167	7845	5719	7310	59,953	62.41%

Table B6. Confusion matrix and performance values for the KNN-based classification system.

Test/Real	Walk	Jump	Immobile	Run	Up	Down	Cycle	Drive	Instances	F_1-Measure
Walk	6726	65	52	143	137	87	121	196	7527	88.56%
Jump	161	6879	26	265	36	27	62	39	7495	91.33%
Immobile	65	26	6719	21	95	64	26	382	7398	92.81%
Run	86	244	36	6854	65	79	84	49	7497	91.55%
Up	149	98	32	41	6881	375	36	27	7639	88.69%
Down	187	21	16	31	504	6870	26	41	7696	89.80%
Cycle	65	51	58	47	124	78	6748	132	7303	93.42%
Drive	223	185	142	74	36	25	41	6672	7398	89.34%
Accumulated	7662	7569	7081	7476	7878	7605	7144	7538	59,953	90.69%

Figure A.2: Confusion matrices showing the detailed results of experiments conducted by Morillo et. al. The first table shows the results of the Ameva algorithm, table B2 the ones of their ANN-based approach, table B3 showes the results from the C4.5 classifier, B4 the bayesian-based classifier, B5 the SVM, and B6 the results of the KNN based approach. Additionally the original tables showed evaluation measures calculated like the F_1, the specificity, recall, and precision values whereas this extracted version solely shows the F_1 measure.[MGARdL15]

Table 7. Performance comparison in % by using measures of evaluation.

Method	Accuracy	Recall	Specificity	Precision	F_1-Measure
Ameva	99.23	96.93	99.56	96.94	96.93
ANN	98.36	93.50	99.07	93.47	93.48
C4.5	97.51	90.08	98.58	90.04	90.03
Bayesian	92.56	70.96	95.75	70.22	70.41
KNN	97.66	90.74	98.66	90.66	90.69
SVM	90.56	63.09	94.61	62.22	62.41
Mahalanobis distance	93.48	87.41	93.87	91.74	91.67
Discriminant analysis	99.15	96.14	97.75	96.37	96.28

Figure A.3: In this table the results Morillo et. al has achieved are given in an overview.[MGARdL15]

Table 5 Classification results

Methods	Accuracy (%)	Root mean squared error	Time taken to build model (s)
J48	89.6	0.1804	0.65
Naive Bayes	75.3	0.283	0.12
SMO	81.1	0.332	1.74

Figure A.4: Classification results quoted from Miao et. al. The achieved accuracies as well as the differences in needed time for computation are shown for J48, Naive Bayes, and SMO [MHL+15]

Table 4 Confusion matrix of J48 decision tree

Model actual	Walking upstairs	Walking downstairs	Walking	Running	Static
Walking upstairs	*915*	18	160	21	0
Walking downstairs	22	*454*	158	41	0
Walking	127	85	*2268*	51	1
Running	23	51	68	*1938*	0
Static	0	0	16	0	*1681*

Figure A.5: Confusion matrix quoted from Miao et. al and resulted from using a J48 decision tree classifier. [MHL+15]

	TIME/FREQUENCY FEATURES: FFT MAGNITUDE Sensitivity - Specificity	WAVELET FEATURES: ABSOLUTE SUM Sensitivity - Specificity
Walking	99 – 99	85 – 92
Upstairs	94 -99	67 – 95
Downstairs	96 - 98	88 – 96
Jog	91 -98	78 – 97
Run	91 – 99	87 – 98
Hop(L leg)	83 – 99	77 – 99
Hop(R leg)	74 - 98	69 – 98
Jump	64 – 99	43 - 97

Figure A.6: Comparison between the results achieved based on time/frequency features and on wavelet features done by Preece et. al. The time/frequency feature based classification outperforms the wavelet based.[PGKH09]

	WK	WU	WD	ST	SD	LD	**Recall**
Walking	**492**	1	3	0	0	0	99%
W. Upstairs	18	**451**	2	0	0	0	96%
W. Downstairs	4	6	**410**	0	0	0	98%
Sitting	0	2	0	**432**	57	0	88%
Standing	0	0	0	14	**518**	0	97%
Laying Down	0	0	0	0	0	**537**	100%
Precision	96%	98%	99%	97%	90%	100%	**96%**

Table 4: Confusion Matrix of the classification results on the test data using the multiclass SVM. Rows represent the actual class and columns the predicted class. Activity names on top are abbreviated.

Figure A.7: Confusion matrix showing the results achieved by Verleysen et. al. A multiclass SVM was used to classify the activities. [Ver13]

Results of WMMS Tool v0.9.9:									
Raw Data: *bb10_sample_sub2.txt*									
Gold Data: *gold_sub2_new.xlsx*									
Options		*Value*		*Info*			*Value*	*Stn.Dv.*	
Feature Window (s):		1		Avg window time (s):			1.00819	0.002183	
End of Shake:		17		Avg samples per window:			45.88571	1.088874	
Tolerance COS (windows):		2		Feature windows/second:			0.99094		
Tolerance CAT (windows):		2		Feature windows:			630		
Sliding Window (%):		0		Time (s):			635.76		
Rotation Matrix:		OFF		Total # of videos:			28		
Detail Level:		3		Total # of changes of state:			42		
WMMS Version:		3		Total gold changes of state:			36		
	TP	*FN*	*TN*	*FP*	*Est Total*	*Act Total*	*Sensitivity*	*Specificity*	*F1-Score*
Stand	86	38	297	10	138	156	0.693548	0.967427	0.781818
Sit	21	0	410	0	34	31	1	1	1
Lie	13	0	416	2	24	18	1	0.995215	0.928571
Walk	190	9	214	18	274	268	0.954774	0.922414	0.933661
Stairs	12	17	402	0	15	44	0.413793	1	0.585366
Small Movement	42	3	349	37	115	83	0.933333	0.904145	0.677419
Change of State	23	20	256	17	40	36	0.534884	0.937729	0.554217
During Stand			84	4					
During Sit			29	0					
During Lie			0	0					
During Walk			120	9					
During Stairs			1	2					
During Brush Teeth			12	0					
During Comb Hair			0	0					
During Wash Hands			0	0					
During Dry Hands			0	0					
During Move Dishes			3	0					
During Fill Kettle			3	1					
During Toast Bread			1	0					
During Wash Dishes			3	1					
Stand to Walk	3	2							
Walk to Stand	5	3							
Walk to Sit	1	0							
Sit to Walk	2	1							
Walk to Lie	1	1							
Lie to Walk	0	1							
Walk to Small Move	0	0							
Small Move to Walk	0	0							
Stand to Small Move	0	0							
Small Move to Stand	0	0							
Walk to Stairs	0	2							
Stairs to Walk	0	2							
Other	11	8							

Figure A.8: The confusion matrix resulted from experiments by Tundo et. al who used a custom based tool called WWMS to achieve the results. The tool is a combination of different classifiers. The relevant confusion is visible in red and has the calculated F_1 score at the end of its rows.[TM]

Table 4. Confusion matrix (k-nearest neighbor classifier with accelerometer and gyroscope features).

Activity	Classified as...								
	C1	C2	C3	C4	C5	C6	C7	C8	C9
C1 = Slow walking	572	30	5	0	0	0	0	0	0
C2 = Normal walking	29	602	13	0	0	4	5	0	1
C3 = Brisk walking	7	17	475	25	0	0	1	0	2
C4 = Jogging	0	1	32	389	0	0	0	2	0
C5 = Sitting	0	0	0	0	266	0	0	0	0
C6 = Normal upstairs	8	15	2	0	0	67	4	0	0
C7 = Normal downstairs	6	7	3	0	0	4	77	0	0
C8 = Brisk upstairs	1	8	10	1	0	0	0	50	1
C9 = Brisk downstairs	0	16	14	0	0	1	0	0	34

Figure A.9: Confusion matrix showing the results gained by Wu et. al who used a k-nearest neighbor approach based on signals from the accelerometer and gyroscope.[WDR+12]

Table 8. Battery lifetime in minutes for the execution with different classifiers.

Algorithm	Test				Mean
	1	2	3	4	
Ameva	1130	1150	1070	1205	1138
KNN	750	730	760	700	731
Binary decision tree (C4.5)	915	925	898	907	905
SVM	820	870	780	830	822
Neural Networks	515	597	578	510	548
Naive Bayes	780	750	820	870	814

Figure A.10: This table shows the results Morillo et. al as gathered by comparing the used classifiers in terms of their effect on the battery lifetime in minutes.[MGARdL15]

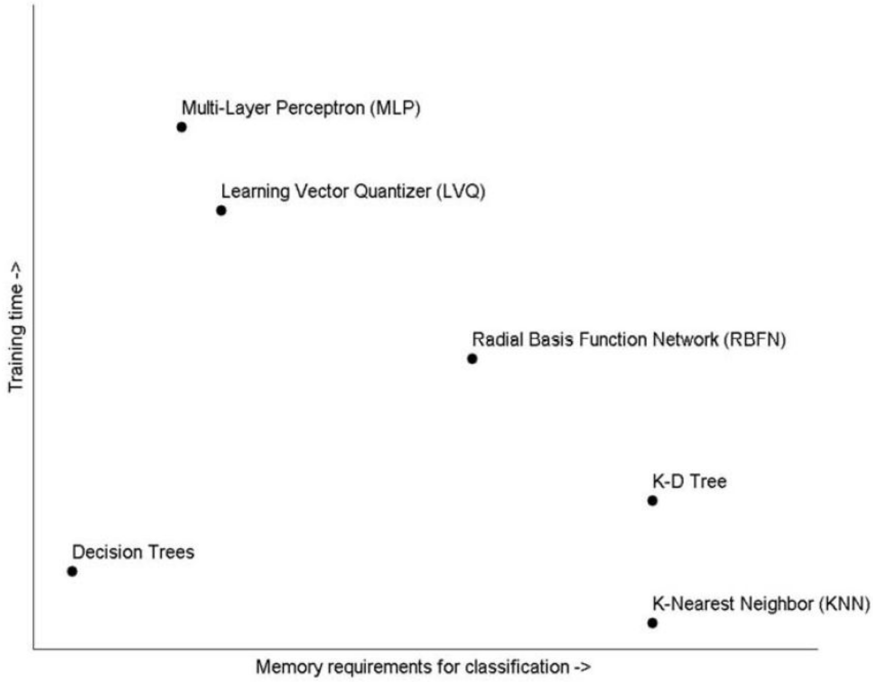


Figure A.11: Pärkkä et. al has compared different approaches based on the memory requirements for the classification and the training time needed. To be implemented on a smartphone and to be able to run offline low memory requirements are important. This chart shows that decision trees are the best option for an offline implementation on a phone[Pär11]

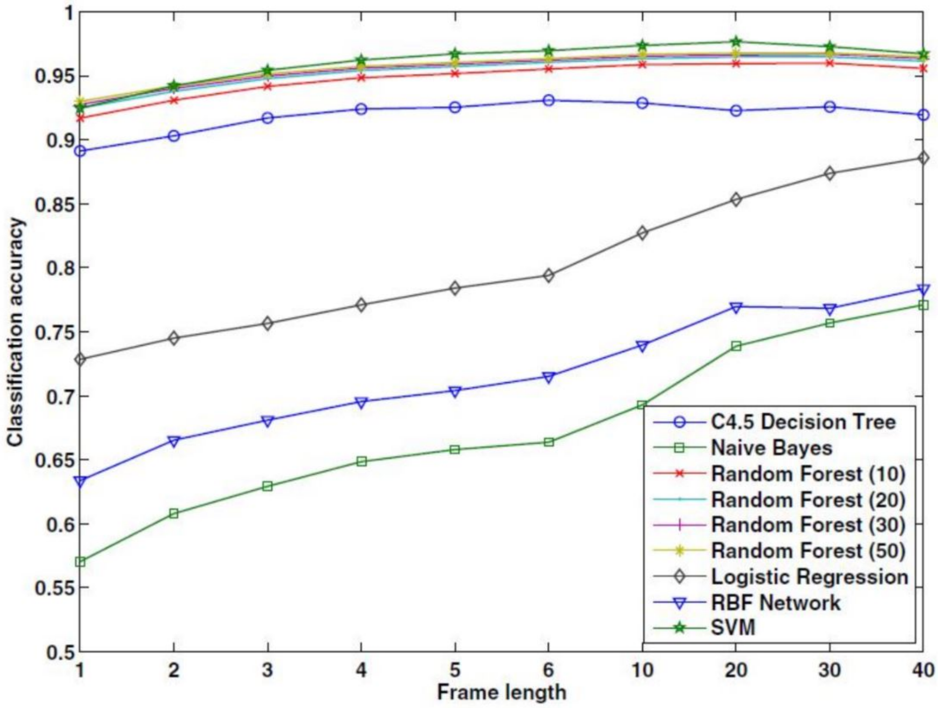


Figure A.12: Sun et. al has conducted experiments for finding an ideal window length and came to the conclusion that the accuracy results improve for longer windows. [SZL11]

Table 5. A comparison of classification accuracies using acceleration features only versus using both acceleration and rotation rate features (k-nearest neighbor classifier).

Activity	Acceleration	Acceleration+ rotation rate	Difference
C1. Slow walking	89.6%	94.1%	+4.5%
C2. Normal walking	85.8%	92%	+6.2%
C3. Brisk walking	78%	90.1%	+12.1%
C4. Jogging	85.4%	91.7%	+6.3%
C5. Sitting	100%	100%	0%
C6. Normal upstairs	65.6%	69.8%	+4.2%
C7. Normal downstairs	66%	79.4%	+13.4%
C8. Brisk upstairs	64.8%	70.4%	+5.6%
C9. Brisk downstairs	49.2%	52.3%	+3.1%
Weighted average	83.7%	90.2%	+6.5%

Figure A.13: Wu et. al has impact of the gyroscope sensor on the achieved accuracy results with a KNN approach. The first column shows the results for only using accelerometer signals and the second shows the accuracy results for both using the signals of accelerometer and gyroscope.[WDR$^+$12]

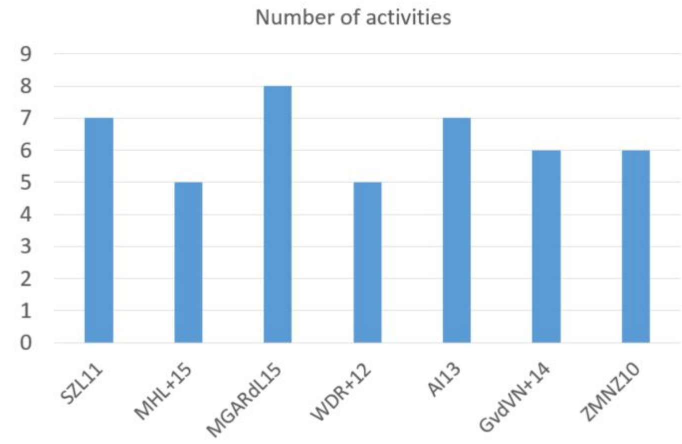


Figure A.14: Overview of the amount of activities monitored in the reviewed literature

Publication	[MGARdL15]	[HL13]	[PGKH09]
Overlap	25; 50	50	50

Table A.1: Overlap of time sampling windows in different publications in percent of window length

Appendix B

Pseudocode

B.1 J48

[Wik16a]

1. Check for the above base cases.
2. For each attribute a, find the normalized information gain ratio from splitting on a.
3. Let a_{best} be the attribute with the highest normalized information gain.
4. Create a decision node that splits on a_{best}.
5. Recur on the sublists obtained by splitting on a_{best}, and add those nodes as children of node.

B.2 Ameva by Gonzales et. al

[GACVO09] *Input:* Data consisting of N examples, l classes, and continuous variables X_i
For every X_i do:

1. Step 1: Initialization of the candidate interval boundaries and the initial discretization scheme.
   (a) Find the maximum (d_k) and minimum (d_o) values of X_i.
   (b) Form a set of all distinct values of X_i, in ascending order, and initialize all possible interval boundaries B with the minimum, maximum, and all the midpoints of all the adjacent pairs in the set.
   (c) Set the initial discretization scheme to $\mathscr{L}$: $|d_o, d_k|$, set $GlobalAmeva = 0$.
2. Step 2: Consecutive additions of new boundaries, which results in the locally highest value of the Ameva criterion.

(a) Initialize $k = 1$;

(b) Tentatively add an inner boundary, which is not already in $\mathscr{L}$, from B, and calculate the corresponding Ameva value.

(c) After all the tentative additions have been tried, accept the one with the highest value of Ameva.

(d) $If(Ameva > GlobalAmeva)$ then update $\mathscr{L}$ with the accepted boundary in step 2(c) and set $GlobalAmeva = Ameva$, else terminate.

(e) Set $k = k + 1$ and go to 2(b)

Output: Discretization scheme $\mathscr{L}(k)$.