

Improving Human Activity Recognition Using Embedded Smartphone Sensors

Bob Oldengarm
University of Twente
7522NB Enschede
The Netherlands

b.m.j.oldengarm@student.utwente.nl

1. ABSTRACT

This paper describes the process and results of finding the most accurate set of features of the data captured by embedded smartphone sensors to recognize six different activities of daily living. The sensor data of the gyroscope and the accelerometer are processed and trained in the J48 and Naive Bayes classifiers to recognize laying, standing, sitting, walking, going upstairs and going downstairs. Starting with 272 features, around half of these are eliminated by using a Ranker method based on the information gain. Afterwards a Wrapper Subset Evaluator is applied and results in the most accurate set of features for the six activities in both classifiers. By training the classifiers with these sets of best features the accuracy improved up to 28.92%, resulting in an overall accuracy for all activities ranging from 95.32% up to 99.97%.

Keywords

Human activity recognition, smartphone sensors, accelerometer, gyroscope, classifiers, J48, NaiveBayes

2. INTRODUCTION

Human activity recognition (HAR) has become increasingly relevant in the last years in different fields such as health monitoring, sports, smart environments, security and elderly care [5]. Many activities can be studied to recognize, including walking, running, laying down, cycling, going upstairs and going downstairs. Although this field of human monitoring is emerging, it is still a challenging field of research. Different approaches have been taken in previous research to monitor human activities. Computer vision based systems to detect suspicious human activities, activity recognition systems based on WiFi signals and human monitoring using body-worn sensors are examples of previous research approaches. However, these proposed solutions either are location specific or require additional hardware. A widely deployable activity recognition system will therefore need a more user-friendly and usable solution.

Such a widely deployable activity recognition system could be achieved by using smartphones. The smartphone usage has rapidly increased over the last years. In early 2019 around 76 percent of the population of advanced

economies owned a smartphone. With more than three quarters of the population owning a smartphone, it would be a viable and low-cost solution.

A smartphone nowadays contains different sensors such as an accelerometer, gyroscope, magnetometer, proximity sensor and a microphone. Being able to extract the produced data from these sensors during different activities can lead to highly accurate activity recognition. This field of smartphone activity recognition has already been explored during different researches. However, the majority of these studies focus on the feasibility of using smartphone sensors for HAR¹. These studies have shown that the embedded smartphone sensors can accurately be used to recognize human activities. Although it has been shown that embedded smartphone sensors can be used for HAR, there is still research that can be done regarding the best features of these sensors to use for each specific activity. This knowledge can then be applied by different applications which are not using a smartphone. An application in elderly care can for example be interested in when an elderly is laying at an unusual time, which could indicate that they are helpless laying on the floor. The features suggested for the recognition of laying down can then be applied in this application. So in conclusion the main research question for this research will be: **Which sets of extracted features from embedded smartphone sensors are most accurate to recognize different activities of daily living?**

3. RELATED WORK

Different studies have been conducted in the field of HAR.

3.1 Camera Based Approaches

Jalal et al.[5] developed a video sensor-based human activity recognition system to monitor in elderly care. They used depth video sensors to create and capture depth silhouettes. Using these silhouettes they created human skeletons which can be processed to recognize activity and live log the elderly. Mean recognition rates of 93 percent were achieved using this technique. A human activity recognition system using a mobile camera was proposed by Song and Chen[6]. They captured the human body using the mobile camera and combined the information of location, pose and elapsed time to recognize the activity. They achieved a recognition rate of 94.8 percent during their conducted experiments.

3.2 WiFi Based Approaches

Li et al.[8] showed in their conducted research that WiFi signals can be used to detect human motions and activities. Their system uses the phase and amplitude information from the channel state information. This information

¹Human Activity Recognition

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

33rd Twente Student Conference on IT July 3rd, 2020, Enschede, The Netherlands.

Copyright 2020, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

contains the effect of the activity and also the environment, from which they extract the signal segments that belong to the human activity. During the simulation, a mean accuracy of 96,6 percent in line-of-sight and a mean accuracy of 92 percent in not line-of-sight was achieved. Arshad et al.[2] also utilized the channel state information, however they utilized all available subcarriers of a WiFi signal. Subcarriers are signal carriers which are carried on top of the main signal carrier to transmit additional information. These subcarriers and main carrier are demodulated separately at the receiving end of the signal. This research was novel as they utilized all subcarriers of the WiFi signal instead of only a small subset of subcarriers. Each subcarrier provides more information about the human activity which will benefit the accuracy. They achieved an average accuracy of 97 percent for multiple communication links.

3.3 Body-worn Sensor Based Approaches

Scheurer et al.[10] researched if Gradient Boosted Trees was an effective decision algorithm for recognising 17 different activities of firefighters. Data of wireless inertial sensor units was used in these Gradient Boosted Decision Trees and compared against k-Nearest Neighbors and Support Vector Machines algorithms. They concluded that the Gradient Boosted Trees outperformed the other two algorithms.

3.4 Smartphone Based Approaches

Kwapisz et al.[7] proposed a human activity recognition system based on the accelerometer in a smartphone. They collected data from twenty-nine users and summarized it into ten seconds data intervals. After classifying they achieved a mean accuracy of over 90 percent for each activity. Bayat et al.[3] also used the accelerometer in smartphones and its data to predict physical human activities. They proposed a low-pass filter for the raw data which isolates the gravity acceleration from the body acceleration. They selected five classifiers and combined them into an optimal classifier. This method reached an mean accuracy of 91 percent. Ronoa and Cho[9] proposed a system in which they used accelerometer and gyroscope sensor data. Firstly a continuous hidden Markov model is applied to separate moving and non-moving activities. Secondly continuous hidden Markov models are applied for classification, to recognize the executed activity. By applying this two stage continuous hidden Markov model, an accuracy of 91 percent was achieved.

4. METHODOLOGY

4.1 Data collection

The first step of the research was to collect the sensor data. Two sensors were used to collect the data during the different activities.

4.1.1 Accelerometer

A triaxial accelerometer will be used to measure acceleration. This accelerometer not only detects acceleration, but also captures vibration and tilt. This can precisely determine movement and orientation along the x-axis, y-axis and z-axis. Figure 1 shows the orientation within a smartphone.

4.1.2 Gyroscope

A gyroscope is much like an accelerometer as it also provides orientation details and direction, but the gyroscope does this with greater precision. The biggest difference is

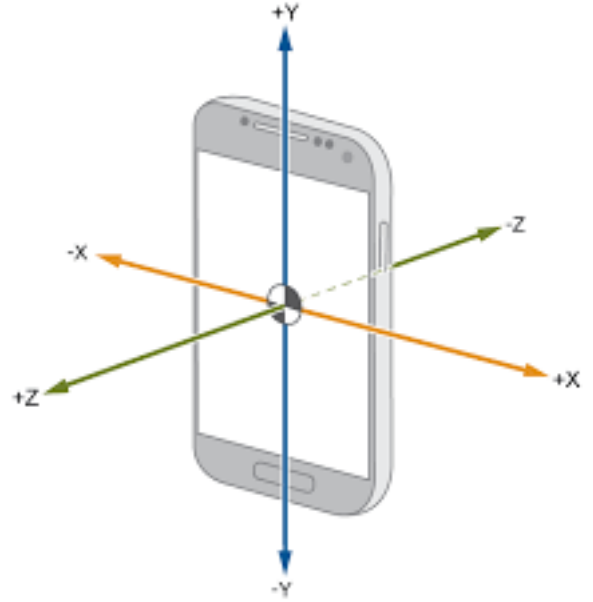


Figure 1. Orientation of accelerometer within a smartphone.

Name	Description
Standing	The user stands right up
Sitting	The user sits
Laying	The user lays down
Walking	The user moves with a slow pace
Going_upstairs	The user walks up the stairs
Going_downstairs	The user walks down the stairs

Table 1. Description of performed activities

that the gyroscope can measure angular velocity, whereas the accelerometer is not able to measure this.

4.1.3 Gathering the data

The initial plan was to build an app for this research to gather the sensor data during the six different activities which are depicted in Table 1. This app was indeed built and was able to retrieve the sensor data from the sensors during the activities. However, gathering this data, labelling it correctly and extracting the features from the data turned out to be too time-consuming within the given time frame of this research. Therefore an existing labelled data set was used during this research[1]. This data set consists of sensor data of 30 people who performed the six activities wearing a smartphone on their waist. The gyroscope captured the 3-axial angular velocity and the accelerometer captured the 3-axial linear acceleration, both at a rate of 50 times per second. After capturing the data, the signals of the accelerometer and the gyroscope were pre-processed using noise filters and sampled in sliding windows of 2.56 seconds with a 50% overlap. This duration and overlap was chosen as the average moving speed of human beings is 1.5 steps per second[4], so this ensures that a full walking cycle of two steps is captured. To separate the 3-axial linear acceleration signal into gravitational acceleration and body acceleration, a low-pass filter was used which was shown to be successful in the research of Bayat et al.[3]. A filter with 0.3Hz cutoff frequency was used on the gravitational force as this force only has low frequency components and maintains a constant gravity signal[1]. In each sliding window a vector of features was

Name	Description
tBodyGyro	Angular velocity of the body
tBodyGyroJerk	Rate at which the angular velocity changes
tBodyAcc	Acceleration of the body
tBodyAccJerk	Rate at which the acceleration changes
tGravityAcc	Gravitational acceleration

Table 2. Naming and description of the features

calculated by using the variables from the time and frequency.

4.2 Feature extraction

4.2.1 Naming of the signals

This section will elaborate on the naming and specification of the different features. The tBodyGyro indicates the angular velocity captured by the gyroscope. As mentioned before, the acceleration signal of the accelerometer was separated into a body acceleration signal, tBodyAcc, and a gravitational acceleration signal, tGravityAcc. The tBodyAcc and tBodyGyro were derived to obtain the jerk of these signals, which indicates the rate at which the acceleration changes, tBodyAccJerk, and the rate at which the velocity changes over time, tBodyGyroJerk. All these five signals have their components in the X, Y and Z directions, which are indicated by -X, -Y and -Z respectively. For example, tBodyAccJerk-X indicates the rate at which the body acceleration changes over time in the X-axis and tBodyGyro-Z indicates the angular velocity in the Z-axis. Table 2 summarizes the naming and description of these signals. Furthermore the magnitude of these five different three-dimensional signals (tBodyGyro, tBodyGyroJerk, tBodyAcc, tGravityAcc and tBodyAccJerk) were calculated using the Euclidean norm, resulting in tBodyGyroMag, tBodyGyroJerkMag, tBodyAccMag, tGravityAccMag and tBodyAccJerkMag. The Euclidean norm is calculated by taking the square root of the Euclidean inner product.

4.2.2 Naming of the variables

Different variables can be estimated from the aforementioned signal vectors. Table 3 shows the different variables that were estimated from the signals.

Name	Description
Mean()	Mean value
Std()	Standard deviation
Mad()	Absolute deviation of the median
Max()	Largest value in the array
Min()	Smallest value in the array
Sma()	Signal magnitude area
Energy()	Energy measure
Iqr()	Interquartile range
Entropy()	Signal entropy
ArCoeff()	Autoregression coefficients
Correlation()	Correlation between two signals
Angle()	Angle between two vectors

Table 3. Clarification of the variable names of the signals

Most of these variables are well known, however some of them need some clarification.

- **Signal magnitude area:** This is defined as the sum of the three axis divided by the number of samples in

a sliding window. This is calculated by the equation

$$SMA = \frac{1}{N} \sum_{i=1}^N (|x(i)|) + (|y(i)|) + (|z(i)|)$$

where N denotes the number of samples and x(i), y(i) and z(i) denote the value of x-axis, y-axis and z-axis respectively.

- **Energy:** The energy of a signal is the area under the squared magnitude of that specific signal.

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt$$

- **Autoregression coefficients:** The following function

$$y(t) = \sum_{i=1}^p \alpha(i)y(t-1) + \epsilon(t)$$

represents an autoregressive model. This model uses previous observations in time to predict the value at the next time step. $y(t)$ is the signal, $\alpha(i)$ are the AR-coefficients, $\epsilon(t)$ is assumed to be the noise on the signal and p is the order of the filter, which is 4 in the case of this research. So this model uses the past 4 values of the signal to estimate the current value of $y(t)$.

Following this signal and variable naming convention, there will be 40 variables for the five signals in Table 2. Furthermore there will be 13 variables for the five magnitude vectors discussed in Section 4.2.1. Finally, there will be 7 angle variables, where the angle between vectors of the mean gravity and the mean of the other signals is estimated. This results in 272 features to use for classification.

4.3 Classification

4.3.1 Type of classifiers

A classifier can be trained with training data to identify in which class a performed activity belongs when providing it the sensor data of the performed action. Weka is a very powerful machine learning tool that can be used to train such classifiers and will therefore be used in this paper. Different types of classifiers are available within Weka.

- **Functions:** These classifiers use models with mathematical functions to classify the data. Logistic and MultilayerPerceptron are examples of such classifiers.
- **Bayes:** Bayes classifiers use numeric estimator precision values based on the training data to classify the data. The best-known Bayes classifier is Naive-Bayes.
- **Trees:** Tree classifiers create decision trees to decide in which class the given data belongs. Examples of these classifiers are J48 and LMT.

Tests with the gathered sensor data will have to be performed to determine which classifiers to use for this research. Accuracy but also time efficiency will be taken into account here as there is a limited time frame for this research.

When it is clear which classifiers will be used, the main data set will be transformed. As the main data set contains all six different activities and therefore cannot be analysed properly regarding a specific activity, the main

data set will be transformed into six different data sets. The main data set will be copied and activities not researched in the data set will be labelled as 'OTHER'. Through this approach it is easier to analyse which features are best for each activity, as with more activities in one data set this information could be influenced by the other activities.

4.3.2 Preliminary feature removal

As there will be 272 features in total from which can be chosen, it will be too time-consuming to apply attribute selection with all of the 272 features. Therefore some of the features will have to be removed preliminary before applying the attribute evaluator on the data. This can be done in Weka by applying a single-attribute evaluator method with ranking. This method will rank the features according to their information gain or the correlation between the feature and the class. Using this method it is possible to eliminate irrelevant features for each data set.

4.3.3 Best feature set selection

After using the ranking method, a Wrapper subset evaluator is applied with a Best First search method. This Wrapper subset evaluator evaluates different feature sets by using a learning scheme. The accuracy of the learning scheme for a set of attributes is estimated by the use of cross validation. This evaluator will repeatedly try to find a better subset of features and eventually outputs the most accurate set of features for the activity.

5. RESULTS

This section shows the results of the classifying process.

5.1 Determination of classifiers

To determine which classifiers will be used, an accuracy test was done on the main data set. The results of this test can be found in Table 4.

Classifier	Accuracy (%)	Duration (seconds)
NaiveBayes	86.6159	6
Logistic	97.9461	2050
J48	95.7291	16
MultiLayerPerceptron	97.7017	5432
LMT	96.8971	3145

Table 4. Accuracy and duration of the different classifiers

Logistic, LMT and MultiLayerPerceptron turned out to take too much time for the given time of this research. Even after eliminating more than half of the features before applying the subset evaluator, it would still take more than 24 hours to apply the subset evaluator on one data set, let alone that it would have to be done on all six data sets. Therefore NaiveBayes and J48 were chosen for the scope of this research. Both classifiers will be trained and tested by using 10-fold cross-validation. This splits the provided data set into ten subsamples. One of these subsamples will serve as testset and the other nine subsamples will serve as trainingsets to train the classifier. This process is repeated ten times after which the average accuracy of these ten folds will be the final accuracy of the classifier.

5.2 Preliminary feature elimination

The feature elimination was executed by applying an Info Gain Attribute Evaluator over the data sets with a Ranker search method. Applying an offset of 0.15 as minimum Info Gain would eliminate around half of the features and

was therefore chosen in all six data sets. Table 5 shows the six activities and the number of features that were left in the data set.

Activity	Number of features left
Going downstairs	132
Going upstairs	147
Laying	143
Sitting	120
Standing	129
Walking	138

Table 5. The six data sets with the amount of features left after preliminary feature elimination

This preliminary feature elimination reduced the average computation time from 7320 seconds to 1260 seconds for finding the best set of features for the J48 classifier. The average computation time for finding the best feature set for the Naive Bayes classifier was reduced from 3960 seconds to 695 seconds.

5.3 Best feature set selection

After removing the features with the least info gain, the Wrapper Subset Evaluator with a Best First search method was applied. This was done for the J48 classifier and the NaiveBayes classifier on all six data sets.

5.3.1 Going Downstairs

Table 6 shows the best sets of features for recognizing going downstairs for the J48 and the Naive Bayes classifiers.

J48	Naive Bayes
tBodyAcc-max()-X	tBodyAcc-arCoeff()-X,4
tBodyAcc-correlation()-X,Y	tBodyAcc-correlation()-X,Z
tGravityAcc-max()-Y	tGravityAcc-arCoeff()-Y,2
tGravityAcc-min()-Z	tGravityAcc-arCoeff()-Z,4
tBodyGyro-correlation()-Y,Z	tBodyAccJerk-arCoeff()-X,2
tBodyGyroJerk-arCoeff()-Z,1	tBodyAccJerk-arCoeff()-Y,2
tBodyAccJerkMag-energy()	tBodyAccJerk-arCoeff()-Z,4
tBodyGyroMag-iqr()	tBodyGyro-correlation()-Y,Z
tBodyGyroJerkMag-min()	tBodyGyroJerk-arCoeff()-Y,3
	tBodyGyroJerk-correlation()-X,Y
	tBodyAccMag-std()
	tBodyAccJerkMag-arCoeff()3
	tBodyGyroJerkMag-arCoeff()3

Table 6. Going downstairs: Best set of features for J48 and Naive Bayes classifiers

5.3.2 Going Upstairs

Table 7 shows the best sets of features for recognizing going upstairs for the J48 and the Naive Bayes classifiers.

J48	NaiveBayes
tBodyAcc-arCoeff()-Y,2	tBodyAcc-mean()-Y
tGravityAcc-std()-Z	tBodyAcc-energy()-X
tGravityAcc-min()-Y	tBodyAcc-correlation()-X,Z
tGravityAcc-min()-Z	tGravityAcc-entropy()-X
tGravityAcc-arCoeff()-Z,1	tGravityAcc-arCoeff()-Y,2
tGravityAcc-correlation()-X,Y	tGravityAcc-arCoeff()-Z,1
tBodyAccJerk-mean()-Y	tGravityAcc-arCoeff()-Z,2
tBodyAccJerk-mad()-X	tGravityAcc-correlation()-Y,Z
tBodyAccJerk-arCoeff()-Y,1	tBodyAccJerk-max()-X
tBodyGyro-std()-Y	tBodyAccJerk-max()-Z
tBodyGyro-mad()-Y	tBodyAccJerk-arCoeff()-Z,2
tBodyGyro-iqr()-Z	tBodyAccJerk-correlation()-X,Y
tBodyGyro-arCoeff()-X,2	tBodyAccJerk-correlation()-X,Z
tBodyGyro-arCoeff()-Y,1	tBodyGyro-iqr()-Z
tBodyGyroJerk-mean()-X	tBodyGyroJerk-arCoeff()-X,4
tBodyAccJerkMag-mad()	tBodyGyroJerk-arCoeff()-Y,2
tBodyAccJerkMag-iqr()	tBodyGyroJerk-correlation()-X,Z
tBodyGyroMag-iqr()	tBodyAccJerkMag-arCoeff()3
	tBodyAccJerkMag-arCoeff()4
	tBodyGyroMag-arCoeff()3
	tBodyGyroMag-arCoeff()4
	tBodyGyroJerkMag-arCoeff()3
	tBodyGyroJerkMag-arCoeff()4
	angle(tBodyGyroJerkMean, gravityMean)

Table 7. Going upstairs: Best set of features for J48 and Naive Bayes classifiers

5.3.3 Laying

Table 8 shows the best sets of features for recognizing laying down for the J48 and the Naive Bayes classifiers. J48 only has one feature in its best feature set, namely the minimal value of the gravity acceleration in the x-axis. This is rather logical, as there is almost no change in acceleration in the x-axis when laying down compared to the other activities.

J48	NaiveBayes
tGravityAcc-min()-X	tBodyAcc-correlation()-X,Y
	tBodyAcc-correlation()-Y,Z
	tGravityAcc-min()-X
	tGravityAcc-iqr()-Y
	tGravityAcc-correlation()-Y,Z
	tBodyAccJerk-arCoeff()-Z,4
	angle(tBodyGyroJerkMean, gravityMean)

Table 8. Laying: Best set of features for J48 and Naive Bayes classifiers

5.3.4 Sitting

Table 9 shows the best sets of features for recognizing sitting for the J48 and the Naive Bayes classifiers.

J48	NaiveBayes
tGravityAcc-max()-Z	tBodyAcc-arCoeff()-X,4
tGravityAcc-min()-Y	tBodyAcc-arCoeff()-Y,4
tGravityAcc-arCoeff()-Y,2	tBodyAcc-correlation()-X,Y
tBodyGyro-mean()-Y	tBodyAcc-correlation()-X,Z
tBodyAccMag-max()	tGravityAcc-mean()-Y
tBodyAccJerkMag-arCoeff()1	tGravityAcc-max()-Y
	tGravityAcc-min()-Z
	tGravityAcc-arCoeff()-Z,3
	tGravityAcc-correlation()-X,Y
	tBodyAccJerk-correlation()-X,Y
	tBodyGyroJerk-entropy()-X
	tBodyGyroJerk-arCoeff()-X,1
	tBodyGyroJerk-arCoeff()-X,4
	tBodyGyroJerk-correlation()-X,Y
	tBodyGyroJerk-correlation()-Y,Z
	tBodyGyroMag-arCoeff()1
	angle(tBodyGyroMean, gravityMean)
	angle(tBodyGyroJerkMean, gravityMean)
	angle(Y,gravityMean)

Table 9. Sitting: Best set of features for J48 and Naive Bayes classifiers

5.3.5 Standing

Table 10 shows the best sets of features for recognizing standing for the J48 and the Naive Bayes classifiers.

J48	NaiveBayes
tBodyAcc-std()-X	tBodyAcc-arCoeff()-X,1
tBodyAcc-max()-Z	tBodyAcc-arCoeff()-Y,4
tBodyAcc-entropy()-Y	tBodyAcc-correlation()-X,Y
tBodyAcc-arCoeff()-X,1	tGravityAcc-max()-Z
tBodyAcc-arCoeff()-Y,2	tGravityAcc-min()-Y
tBodyAcc-correlation()-X,Z	tGravityAcc-arCoeff()-Z,4
tGravityAcc-max()-Y	tBodyAccJerk-arCoeff()-X,2
tGravityAcc-max()-Z	tBodyGyro-arCoeff()-Y,1
tGravityAcc-energy()-Y	tBodyGyro-arCoeff()-Y,4
tGravityAcc-entropy()-X	tBodyGyroJerk-arCoeff()-X,1
tGravityAcc-arCoeff()-X,3	tBodyGyroJerk-arCoeff()-X,2
tGravityAcc-arCoeff()-X,4	tBodyGyroJerk-arCoeff()-X,3
tBodyAccJerk-std()-Z	tBodyGyroJerk-arCoeff()-X,4
tBodyAccJerk-mad()-X	tBodyGyroJerk-arCoeff()-Z,4
tBodyAccJerk-max()-Z	tBodyGyroJerk-correlation()-X,Y
	tBodyGyroJerk-correlation()-Y,Z
tBodyAccJerk-min()-Z	tBodyGyroMag-arCoeff()1
tBodyAccJerk-arCoeff()-X,1	tBodyGyroJerkMag-arCoeff()1
tBodyGyro-arCoeff()-Z,3	tBodyGyroJerkMag-arCoeff()2
tBodyGyroJerk-energy()-Y	tBodyGyroJerkMag-arCoeff()3
tBodyGyroJerk-iqr()-Y	angle(tBodyGyroJerkMean, gravityMean)

Table 10. Standing: Best set of features for J48 and Naive Bayes classifiers

5.3.6 Walking

Table 11 shows the best sets of features for recognizing walking for the J48 and the Naive Bayes classifiers.

J48	NaiveBayes
tBodyAcc-std()-X	tBodyAcc-mean()-X
tBodyAcc-mad()-Y	tBodyAcc-energy()-X
tBodyAcc-correlation()-X,Y	tBodyAcc-arCoeff()-Y,3
tGravityAcc-mean()-X	tBodyAcc-arCoeff()-Y,4
tGravityAcc-min()-Z	tBodyAcc-correlation()-X,Y
tGravityAcc-energy()-X	tGravityAcc-sma()
tGravityAcc-entropy()-Y	tGravityAcc-entropy()-X
tGravityAcc-arCoeff()-Y,1	tGravityAcc-arCoeff()-X,1
tBodyAccJerk-min()-X	tGravityAcc-arCoeff()-Y,1
tBodyAccJerk-correlation()-X,Y	
tBodyGyro-std()-Y	tBodyAccJerk-arCoeff()-Z,4
	tBodyAccJerk-correlation()-X,Y
tBodyGyro-max()-Y	tBodyGyro-mean()-X
tBodyGyro-entropy()-X	tBodyGyro-arCoeff()-Y,4
tBodyGyro-correlation()-Y,Z	
tBodyGyroJerk-std()-Y	tBodyGyro-arCoeff()-Z,2
	tBodyGyro-correlation()-X,Y
tBodyGyroJerk-mad()-Y	tBodyGyroJerk-mad()-X
tBodyGyroJerk-max()-Z	tBodyGyroJerk-correlation()-X,Z
tBodyGyroJerk-correlation()-X,Y	tBodyGyroJerk-correlation()-Y,Z
tBodyGyroJerk-correlation()-X,Z	
tBodyAccJerkMag-std()	tBodyAccMag-arCoeff()1
tBodyAccJerkMag-mad()	angle(tBodyGyroJerkMean, gravityMean)
tBodyAccJerkMag-arCoeff()4	
tBodyGyroMag-mean()	

Table 11. Walking: Best set of features for J48 and Naive Bayes classifiers

5.4 Accuracy improvement

This section describes the improvement of the accuracy of the J48 and Naive Bayes classifiers when trained and tested with the best set of features.

5.4.1 Improvement of J48 classifier

Table 12 shows the accuracy of the J48 classifier for all activities when used with all 272 features and with only their best features. All accuracies are increased, except the accuracy of laying which remained the same. This can be explained by the fact that the tree built in the J48 classifier for this activity consists of only one feature, tGravityAcc-min()-X, which is the same as the feature in the best feature set.

	J48 All features	J48 Best features
Going downstairs	97.463%	98.409%
Going upstairs	98.354%	98.898%
Laying	99.973%	99.973%
Sitting	97.565%	98.422%
Standing	98.109%	98.681%
Walking	97.661%	98.422%

Table 12. Accuracies of the J48 classifier with all features and only the best features

5.4.2 Improvement of Naive Bayes classifier

Table 12 shows the accuracy of the Naive Bayes classifier for all activities when used with all 272 features and with

	NB All features	NB Best features
Going downstairs	75.626%	98.041%
Going upstairs	75.544%	97.824%
Laying	84.290%	99.878%
Sitting	66.553%	95.478%
Standing	83.991%	95.321%
Walking	79.108%	97.851%

Table 13. Accuracies of the Naive Bayes classifier with all features and only the best features for each activity.

only their best features. All accuracies are substantially increased to a minimum accuracy of 95.321%.

5.4.3 Overall improvement

Table 14 shows the overall improved accuracy of both classifiers for each activity.

Activity	Improvement NB	Improvement J48
Going downstairs	22.416%	0.946%
Going upstairs	22.280%	0.544%
Laying	15.588%	0%
Sitting	28.924%	0.857%
Standing	11.330%	0.571%
Walking	18.743%	0.762%

Table 14. Overall improvement of using the best features for the Naive Bayes classifier and the J48 classifier.

The accuracy of the Naive Bayes classifier is significantly improved with improvements ranging from 11% to almost 29%. The improvements of the J48 classifier are lower, however these accuracies were already minimal 97.46% when performed with all features.

Table 15 shows the best accuracies of both classifiers. The bold percentage in each row points out the most accurate classifier for the activity. The J48 classifier is still the best performing classifier for all six activities. However, the differences in accuracies of both classifiers are significantly lower now.

	NB	J48
Going downstairs	98.041%	98.409%
Going upstairs	97.824%	98.898%
Laying	99.878%	99.973%
Sitting	95.478%	98.422%
Standing	95.321%	98.681%
Walking	97.851%	98.422%

Table 15. Accuracies of the Naive Bayes and J48 classifiers with only the best features for each activity.

Similar work in Human Activity Recognition using smartphone sensors such as Kwapisz et al. [7], Bayat et al. [3] and Ronoa and Cho[9] showed an average accuracy of 90% to 92%. The described results show that the approach in this research outperforms those approaches as it reached accuracies ranging from 95.32% up to 99.97% by determining the best sets of features for each activity.

6. CONCLUSION

In conclusion, the accuracy of the J48 and Naive Bayes classifiers can be improved for almost all activities by determining the best set of features from the data sets. By first eliminating the least informative features using a Ranker search method and thereafter running a Wrapper Subset Evaluator with a Best First search approach it is possible to increase the accuracy of the Naive Bayes classifier up to 28.9%. The J48 classifier has only minor improvements as it already has a minimum accuracy of 97.46%. However, the accuracy still improves with at least 0.5% when the best set of features is used in the training process of the classifier. The J48 classifier still outperforms the Naive Bayes classifier in terms of accuracy for all activities. However, the accuracies of both classifiers are closer together when trained with only the best feature sets. The J48 classifier is best when accuracy is the main objective for an application. If computation time is an issue for an application, then the Naive Bayes classifier is the best choice as it delivers high accuracies within the fastest time.

7. FURTHER RESEARCH

Due to time restrictions of this research there was no time available to improve the more time consuming classifiers such as MultiLayerPerceptron or Logistic. This could be done in further research where there is more time available. Suggestions for further research could also be to skip the step of preliminary feature elimination, which was also used in this research due to time constraints. However, even higher improvements of accuracy might be achieved when skipping this step as features with less information gain can contribute to the overall accuracy. Although the information gain of these features is low, they can be the determining factor in the algorithms when improving the accuracy. A last suggestion for further research is to find the best set of features for combined activities, as this paper only focuses on the best features for the separate activities.

8. REFERENCES

- [1] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones., 2013.
- [2] S. Arshad, C. Feng, Y. Liu, Y. Hu, R. Yu, S. Zhou, and H. Li. Wi-chase: A WiFi based human activity recognition system for sensorless environments. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, 6 2017.
- [3] A. Bayat, M. Pomplun, and D. A. Tran. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones. *Procedia Computer Science*, 34:450–457, 2014.
- [4] C. BenAbdelkader, R. Cutler, and L. Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 372–377. IEEE.
- [5] A. Jalal, S. Kamal, and D. Kim. A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments. *Sensors*, 14(7):11735–11759, 7 2014.
- [6] Kai-Tai Song and Wei-Jyun Chen. Human activity recognition using a mobile camera. In *2011 8th International Conference on Ubiquitous Robots and*

- Ambient Intelligence (URAI)*, pages 3–8. IEEE, 11 2011.
- [7] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 3 2011.
 - [8] H. Li, X. He, X. Chen, Y. Fang, and Q. Fang. Wi-Motion: A Robust Human Activity Recognition Using WiFi Signals. *IEEE Access*, 7:153287–153299, 2019.
 - [9] C. A. Ronao and S.-B. Cho. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models. In *2014 10th International Conference on Natural Computation (ICNC)*, pages 681–686. IEEE, 8 2014.
 - [10] S. Scheurer, S. Tedesco, K. N. Brown, and B. O’Flynn. Human activity recognition for emergency first responders via body-worn inertial sensors. In *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 5–8. IEEE, 5 2017.