Random Forest Algorithm for Human Activities and Postural Transitions Recognition

1st Lucas Scheerer
Computer Science
University of Kassel
Kassel, Germany
uk077861@student.uni-kassel.de

2nd Cagri Isikver
Electrical Communication Engineering
University of Kassel
Kassel, Germany
uk097840@student.uni-kassel.de

3rd Balaji Shankar
Electrical Communication Engineering
University of Kassel
Kassel, Germany
uk096888@student.uni-kassel.de

4th Imran Khan
Electrical Communication Engineering
University of Kassel
Kassel, Germany
uk097434@student.uni-kassel.de

***Abstract*—The technological advances of recent years have allowed the use of precise sensors in smartphones to record more data, which can be used to analyse a person's behavior. Access to more precise data, paired with the heightened interest in machine learning techniques, increases the importance of human activity recognition in fields such as healthcare, fitness or many more. Many techniques have been proposed and examined to further increase the accuracy and efficiency of recognition. This study discusses the viability and efficiency of a random forest classifier as a simple activity recognition algorithm. For this purpose the segmentation, feature extraction, importance calculation, and different optimisation techniques such as random search, halving random search, and grid search are performed and compared. The results of this study show, that a random forest classifier can determine the type of activity with over 90% accuracy, which can be increased slightly more by applying different optimisation methods.**

***Index Terms*—Activity Recognition, Random Forest, feature importance**

I. Introduction

The field of human activity recognition (HAR) is a quickly developing area of study that aims to recognise and comprehend the different movements that people make on a daily basis. Numerous applications, such as those in healthcare, sports science, ergonomics, and smart environments, depend on this field of study. The applications of HAR aim to increase safety and maximise performance by precisely identifying these activities and their transitions. In order to recognise human activities, various methods have been suggested.

This paper investigates the performance of the random forest classifier, a simple machine learning algorithm, in recognising human activities and postural transitions (HAPT). The data set used in this paper was recorded by a smartphone-based recognition method, which includes the data read by the accelerometer and gyroscope of a smartphone that 30 volunteers were wearing during the experiments [1]. During these, the volunteers performed six activities, which are labelled as sitting, standing, lying, walking, walking upstairs, and walking downstairs. The postural transitions between the static postures (sitting, standing, and lying) were also recorded during the experiment and labelled as stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand for a total of 12 different classes of activities [1].

This paper consists of the following sections: Section I is the introduction to this work, while section II includes a review of the state-of-the-art of current human activity recognition methods with discussions of the results. Section III describes the random forest classification algorithm. Section IV describes the implementation of this algorithm used in this study, including segmentation, feature extraction and importance calculation. Additionally a comparison of different hyperparameters, the investigation of potential over-fitting or under-fitting phenomena, and fine tuning of the model are discussed in this section. The results of this classifier are discussed and evaluated in sections V and VI. The later section also includes the conclusion of this study.

II. Related Work

HAR is an area that is increasing in importance day by day and has been the subject of various studies. In order to increase the efficiency and performance of the classification algorithms employed in this field, many approaches and designs have been proposed, and different methods have been used. Attal et al. [2] conducted a comparative study of various classification techniques, including methods like k-Nearest Neighbour (k-NN), Support Vector Machines (SVM), and Random Forest, as well as unsupervised approaches such as k-Means and Hidden Markov Models (HMM). Their results highlighted the better performance of k-NN and HMM, especially when used with effective feature extraction methods. This work demonstrated the importance of method selection in HAR, with supervised techniques generally outperforming unsupervised ones. Similarly, Shian-Ru et al. [3] explored video-based HAR, utilising Gaussian Mixture Models (GMM), dynamic time warping (DTW), and HMMs to ease the difficulties of recognising activities under varying conditions, such as changes in human appearance and the need for clear view-points. The study pointed out that ongoing refinement in video-based methods is required to increase accuracy and robustness. Additionally, Robertson et al. [4] proposed a method for recognising activities in video footage, combining probabilistic searches with HMMs to model human activities as sequences of actions, which proved effective in contexts like automated sports commentary and surveillance.

Deep learning has been in use for HAR, especially through the use of Convolutional Neural Networks (CNNs). Ronao et al. [5] introduced a CNN-based approach for HAR using smartphone accelerometer and gyroscope data, achieving a high accuracy of 95.75% through meticulous hyper-parameter tuning. Chen et al. [6] similarly adapted CNNs to process tri-axial acceleration signals, performing an accuracy of 93.8% without the need for traditional feature extraction. These studies demonstrated CNNs' efficiency on capturing data and performing high accuracy in HAR systems. Ziaeefard et al. [7] added another opinion by focusing on semantic features in HAR, demonstrating that incorporating high-level features such as body pose, related objects, and scene context generally leads to better recognition performance compared to low-level features alone. Gupta et al. [8] reviewed HAR methods using wearable sensors, emphasising the effectiveness of combining feature selection techniques like Relief-F and SFFS with classifiers such as Naive Bayes and k-NN, achieving an accuracy score of 98% in classifying daily activities. These approaches collectively highlight the development of HAR, where both traditional and deep learning methods play important roles in advancing accuracy and efficiency.

Feng et al. [9] proposed an ensemble learning algorithm for HAR using Random Forest. It is stated that the algorithm combines different Random Forest classifiers created for different sensor feature sets to utilise a stable classifier with high accuracy. Results show that Random Forest classifiers outperformed other machine learning algorithms like k-NN, SVM, and Bayes Net with an accuracy of 93.44% and a comparatively faster training time. Similarly, Nia et al. [10] combined Artificial Neural Networks (ANNs) with Random Forest to build a high-accuracy HAR system. They suggested that ANNs' ability to extract complex features can perfectly work for real-world applications with Random Forest's collective prediction skills. It is observed that for Inertial Measurement Unit (IMU) data, the ANN-Random Forest algorithm performed with an accuracy up to 98.84%, making it a vary reliable HAR system. Nurwulan et al. [11] compared different machine learning techniques such as ANN, k-NN, SVM, linear discriminant analysis (LDA), and Bayes with Random Forest for HAR using the same data set as our study. It is shown that Random Forest outperformed other techniques with 87.16% accuracy.

III. RANDOM FOREST ALGORITHM

Belgiu and Drăguţ [14] describe the random forest classifier as an ensemble classifier that randomly selects samples from the provided training data to construct a set of classification and regression trees. Both the total number of trees as well as the number of features used to split each individual node are determined by the user. This ensures a high degree of variance among the generated decision trees, whose decisions are averaged to determine the class assigned to each new input. Ali et al. [12] explain further, that random forests work by executing the following steps: First a number of samples is randomly chosen among the available training data. These samples then serve as training data to construct a decision tree, using a a set number of features to split the data at each node. The features used for this split are also randomly chosen and their best spread is calculated and applied to the node. This continues, until the tree can no longer continue to grow. Once a tree is fully grown a new set of samples is selected and another tree is constructed, until the required number has been reached. Ali et al. [12] add that random forest classifiers are well suited to higher-dimensional data modelling and that the combination of bagging and decision trees allows a random forest to perform better in most cases than decision trees on their own would.

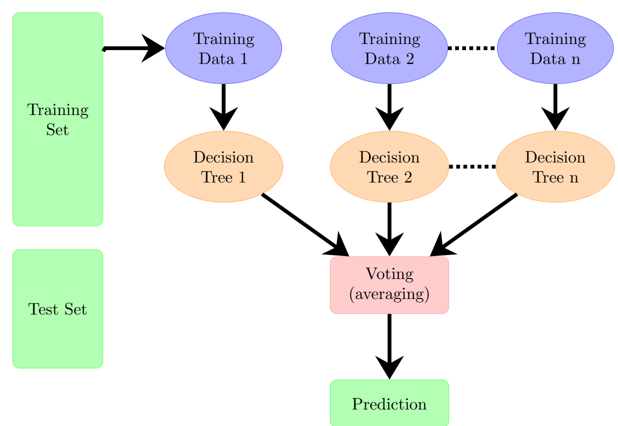

Fig. 1. Random Forest Prediction process from Training Data sets. [13]

By solving the combined output of multiple decision trees, a random forest can achieve higher accuracy by dealing with complex datasets, reducing over-fitting by averaging their predictions, resistance to noise, effectively handling large numbers of input or high-dimensional data, and handling missing values in the data. With its excellent performance and accuracy, it has various real-time applications in banking to calculate credit scores and detect fraud in transactions. It has also been used in health care for disease diagnosis and assessing risk factors. Other use-cases of random forests include face recognition and object detection.

Random forests do have some challenges. The most commonly faced issue is time consumption. Even though it can handle its large data sets more accurately, this will slow the data processing as they are calculating for each individual decision tree. They also need heavy resources to store more data. Random forest is a more complex process when completed with decision trees.

IV. Methodology

The HAPT data set provided by Reyes-Ortiz et al. [1] consists of the measurements of the accelerometer and gyroscope of a smartphone, which were recorded at 50 Hz during 61 different experiments. This data set includes multiple features as well as one training and one test set. The exact process used to segment the data set and compute the feature values is, however, unknown. Since this work includes the training, testing, and evaluation of a random forest classifier, a closer understanding of the features used during those steps is essential. For this reason, both the segmentation step and the feature extraction were implemented in the pre-processing step of the python program for this project.

During the pre-processing step, the id of the experiment is used to group the raw data of the accelerometer and the gyroscope by writing the contents of both sensor files of each experiment into one data frame. Additionally, file labels.txt is used to assign each measurement the matching activity ID, if one exists. This creates 61 data frames, containing a full-time series of measurements from both sensors as well as the activity during which each sample was recorded. This allows the application of a sliding window algorithm, which is used to segment each experiment into smaller time frames. These are constructed in a way that ensures that each window includes approximately five seconds, or 250 measurement cycles, and has a 25% overlap with each of its neighbours.

After each experiment is segmented, the feature values of all individual windows are computed. The chosen features consist of the axial mean, median, standard deviation, and minimum and maximum values. Additionally, the covariance between the six different measurements is determined and also used as a feature.

Since the windows consist of small time series, a fast Fourier transformation (FFT) can be used to calculate the corresponding frequency domains. After the transformation, the same methods from before are used to compute additional features. Afterwards, the number of time steps belonging to each activity in a window is counted, and the most common activity is chosen as the label for the window. Since not all measurements in the data set have been assigned to one of the twelve actions, there are some windows that have no information about any of the activities. This means that these windows are unable to contribute to the classification process and are thus removed from the set of windows used in testing and training. This is meant to lessen the risk of potential classification errors caused by learning a behaviour that does not match any action. Finally, the feature values and activity ID of each window are written into a file and saved for later use.

Before the random forest classifier can be used, a training and test set must be computed. For this purpose, a built-in function of sci-kit-learn is used to randomly split the windows that were produced during preprocessing. To guarantee that the experiments can be replicated, the same random seed was chosen each time, ensuring a random but repeatable split of data between training and test sets.

After determining which windows will be used to train the classifier, the sci-kit module is used to initialise a random forest instance with the default settings and a known random seed to ensure that all results can be replicated. Then the random forest is trained on the feature values and activity IDs before predicting the activities belonging to each window of the test set. To evaluate the performance of the classifier, the accuracy score of the predictions, the train score, the confusion matrix, and the importance of each individual feature are computed and displayed.

In order to optimise the parameters of the classifier, a random search function is used to train and test the classifier with different parameter sets and determine the one that provides the best accuracy. Similarly, a halving random search is also used to test different parameter values and their effect on the accuracy of the classifier. After both random search variations have determined the sets of parameters that achieved the highest accuracy, a grid search is used for further fine-tuning by testing parameter combinations close to the previously determined sets.

V. Evaluation

After running the classifier as described in the previous section the results of the different variations are collected and compared.

A. Classification Report

TABLE I
F1-score of the random forest depending on its parameters

Activity	Default	RS	HRS	RGS	HGS
Walking	0.92	0.94	0.94	0.94	0.94
Walking upstairs	0.94	0.92	0.94	0.94	0.94
Walking downstairs	0.94	0.92	0.94	0.94	0.94
Sitting	0.91	0.91	0.91	0.92	0.91
Standing	0.90	0.92	0.91	0.92	0.92
Laying	0.96	0.97	0.97	0.97	0.97
Stand to sit	0.80	0.75	0.80	0.80	0.80
Sit to stand	0.80	0.71	0.80	0.80	0.80
Sit to lie	0.54	0.57	0.59	0.56	0.57
Lie to sit	0.70	0.73	0.67	0.67	0.73
Stand to lie	0.47	0.47	0.48	0.47	0.47
Lie to stand	0.62	0.62	0.60	0.62	0.62

After running the random forest classifier with different sets of parameters, the achieved F1-scores of the activities have been noted and displayed in table I. The different sets of parameters, which are shown in table II include the default parameters used by sklearn, as well as the parameter sets computed with a random search (RS), a halving random search (HRS), and grid search iterating on the results of both the random search (RGS) and the halving random search (HGS). As can be seen in table I, the different implementations of the classifier achieve similar F1-scores for each of the individual actions, deviating only one to two percentages in most cases, for example in the classes "Laying", "Walking" and "Sitting". The largest difference between the implementations was observed in the class "Sit to stand", where the random search implementation achieves a F1-score 9% lower than the other parameter sets. Among the different implementations

TABLE II
PARAMETERS OF THE DIFFERENT RANDOM FOREST IMPLEMENTATIONS

	Default	RS	HRS	RGS	HGS
n_estmators	100	1733	4455	1733	4580
max_depth	∞	100	50	103	53
min_samples_split	2	2	5	2	3
min_samples_leaf	1	2	2	1	1
bootstrap	True	False	False	False	False

the random search achieves the two values with the highest difference compared to the other versions, namely in the classes "Stand to sit" and "Sit to stand". Both the default and random search implementations are among the lowest F1-scores in six of the twelve actions and only five times among the highest. Among the other implementations, the ones that utilise a grid search were among the implementations with the highest F1-score in nine of twelve activities and only two times among the lowest F1-scores. This difference in performance is underlined by the different test and training scores shown in table III, which show that the random forest implementations using a grid search achieve a higher test score than the other parameter sets. These implementations do, however, seem to be prone to overfitting, while the random forest versions using a random search and a halving random search seem to achieve lower training and test scores.

TABLE III
TRAIN AND TEST SCORES FOR THE DIFFERENT RANDOM FOREST IMPLEMENTATIONS

	Default	RS	HRS	RGS	HGS
Training score	1.0	0.9997	0.9997	1.0	1.0
Testing score	0.9034	0.9153	0.9136	0.9162	0.9162

B. Feature Importance

Since the 114 features differed in their influence on the decisions of the classifiers, the most influential features were determined and compared. For this purpose, table IV shows the five most important features and the corresponding importance score for each of the different random forest implementations. Features starting with the letters "fft" were extracted from the raw data after its fast Fourier transformation. The letters "xa" indicate that the feature was extracted from x-axis measurements of the accelerometer, where "yg" would indicate that a feature was extracted from a gyroscopic measurement along the y-axis. The last part of the feature name is the function used to determine its value, a feature that ends in "mean" for example, is the mean value of the corresponding measurement over the entire window. The results in table IV show that measurements of the accelerometer are more influential than the gyroscopic measurements, since there is not a single gyroscopic feature among the five most important. Among these features, some are present in all variations and achieve similar importance scores for each implementation of the classifier. The feature "fft_xa_mean" for example, is the third most important for the default random forest and the second most influential for the other implementations.Similarly, the feature "xa_mean" reached the highest importance score of over 5% in three of the five variations. When observing the features in table IV, it is notable that most of them relate to acceleration data measured along the x axis, with "ya_mean" being the only exception. Additionally, the table shows that the majority of the most important features were extracted from fast Fourier transformed acceleration measurements, hinting at the use of frequency values to determine cyclic movements.

TABLE IV
THE FIVE MOST IMPORTANT FEATURES FOR EACH PARAMETER SET OF THE CLASSIFIER

Default	RS	HRS	RGS	HGS
xa_max (4.78%)	xa_max (5.12%)	xa_mean (5.46%)	xa_mean (5.10%)	xa_mean (5.19%)
fft_xa_var (3.28%)	fft_xa_mean (3.16%)	fft_xa_mean (3.23%)	fft_xa_mean (3.17%)	fft_xa_mean (3.19%)
fft_xa_mean (3.00%)	fft_xa_var (2.95%)	fft_xa_xa_cov (3.02%)	fft_xa_xa_cov (2.85%)	fft_xa_xa_cov (2.93%)
xa_xa_cov (2.83%)	fft_xa_xa_cov (2.95%)	fft_xa_var (2.83%)	fft_xa_var (2.81%)	fft_xa_std (2.76%)
ya_mean (2.59%)	fft_xa_std (2.73%)	fft_xa_std (2.80%)	fft_xa_std (2.66%)	fft_xa_var (2.74%)

C. Interpretation of the results

In order to interpret the results of the classification, the differences and similarities between the different activities should be examined. The confusion matrix displayed in figure 2 shows how often each activity has been guessed correctly and which activities have been guessed by mistake. Each row of this matrix shows how the instances of the corresponding activity have been guessed. The columns of the matrix indicate which action has been guessed. The second row for examples shows that out of 200 instances of the activity "walking upstairs" 184 were classified correctly, while ten windows were mistaken as "walking" and six were classified as "walking upstairs".

Figure 2 shows that transitional activities like "stand to lie" have a higher percentage of instances that are not classified correctly, while dynamic actions like "walking" are often recognised correctly. This hints at the advantage of cyclic movements like walking or static movements like sitting in regular patterns, which seem to be easier to be recognised by the classification algorithms. On the other hand, it is noticeable that the transitional activities are often mistaken for either similar transitions or the activities transitioned between. For example, the activity "sit to lie" is misclassified as "laying" and "stand to lie". From this observation, the following conclusion can be drawn: The random forest models trained in this work make classification errors more often if parts of the actual and the assumed activity are similar or if one activity is partially included in another.

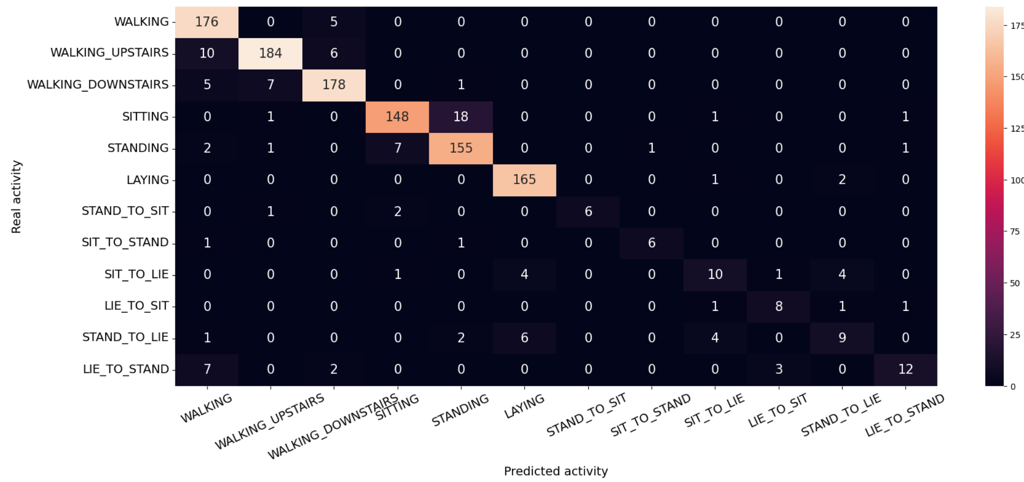

Fig. 2. Confusion matrix of the prediction made with the default random forest classifier

VI. Conclusion

Human activity recognition is a subject that increases its importance day by day, especially in the field of health, and on which there are many studies. Many different algorithms and methods have been used to increase the performance of activity recognition and make it more efficient, and some of these methods are included in this article. In this study, a data set collected with a smartphone-based recognition method, which includes the measurements of a smartphone accelerometer and gyroscope, was examined. The data set contains the measurements of experiments in which 30 volunteers performed 12 daily human activities, including dynamic activities and postural transitions. The classification process was performed using a random forest algorithm, a machine learning technique, combined with segmentation, feature extraction, importance calculation, comparison of different hyperparameters, investigation of potential over-fitting or under-fitting phenomena, and fine tuning of the model.

The classification report showed that the Random Forest model used in this work achieved an accuracy of up to 90.34% with its predictions, which outperforms the RF-classifier from the study by Nurwulan et al. [11], that used the same data set. While fine-tuning the model with different hyper-parameters and optimisation techniques, an accuracy of 91.62% can be achieved when a grid search is used to iterate on the results of a random search or halving random search, resulting in the best performance among all the optimisation techniques examined in the study. Even though the grid search approaches resulted in the highest accuracy gain, they were not able to reduce the overfitting of the model. While the random search on its own provided a lesser accuracy gain, it also resulted in a lower training score, indicating lower chances of overfitting. It is also observed that the random forest algorithm has better recognition performance for the six dynamic and static activities compared to postural transition activities, as the dynamic and static activities gave higher f1-scores. The results of the feature importance calculations showed that the measurements provided by the smartphone's accelerometer contributed more to activity recognition performance than the data provided by the gyroscope. In future studies, different methods and optimisation techniques can be used to increase the accuracy provided by our algorithm.

References

[1] Reyes-Ortiz J., Anguita D., Oneto L., and Parra X. (2015). Smartphone-Based Recognition of Human Activities and Postural Transitions. UCI Machine Learning Repository.

[2] Attal, F., Mohammed, S. S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. Y. (2015). Physical Human Activity Recognition Using Wearable Sensors. Sensors, 15(12), 31314–31338.

[3] Shian-Ru K., Hoang Le U., Yong-Jin L., Jenq-Neng H., Jang-Hee Y. Kyoung-Ho C., (2013). A Review on Video-Based Human Activity Recognition. Computers. 2, 88–131.

[4] Robertson, N., & Reid, I. (2006). A general method for human activity recognition in video. Computer Vision and Image Understanding, 104(2–3), 232–248.

[5] Ronao, C. A., & Cho, S. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. Expert Systems With Applications, 59, 235–244.

[6] Chen, Y., & Xue, Y. (2015). A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. 2015 IEEE International Conference on Systems, Man, and Cybernetics, 1488-1492.

[7] Ziaeefard M., & Bergevin R., (2015). Semantic human activity recognition: A literature review. Pattern Recognition, 48, 2329–2345.

[8] Gupta, P.K., & Dallas, T. (2014). Feature Selection and Activity Recognition System Using a Single Triaxial Accelerometer. IEEE Transactions on Biomedical Engineering, 61, 1780-1786.

[9] Feng, N. Z., Mo, N. L., & Li, N. M. (2015). A Random Forest-based ensemble method for activity recognition. IEEE. -

[10] Nia, N. G., Amiri, A., Nasab, A., Kaplanoglu, E., & Liang, Y. (2023). The Power of ANN-Random Forest Algorithm in Human Activities Recognition Using IMU Data. University of Tennessee.

[11] Nurwulan, N. R., & Selamaj, G. (2020). Random Forest for human daily activity recognition. Journal of Physics Conference Series, 1655(1), 012087.

[12] Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9(5), 272.

[13] (2023). *Random Forest Algorithm* Simplilearn (last accessed 31.08.2024) https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm

[14] Belgiu, Mariana and Drăguţ, Lucian (2016). Random forest in remote sensing: A review of applications and future directions, ISPRS journal of photogrammetry and remote sensing, volume 114 pages 24-31