# Automatic Detection of Timed-up and Go Tests with IMU Sensor Data

Franz Kroll, Sven Löffler, Ilja Becker and Petra Hofstedt
*Brandenburg University of Technology, Cottbus, Germany*

Keywords: Timed-up and Go, Machine Learning, Wearable Devices, Activity Recognition, Accelerometer, Gyroscope, Physical Activities, Assessment System, Automatic Segmentation.

Abstract: The evaluation of the current state and examination of geriatric patients is a time consuming process for the medical staff and the patients. The independence of the geriatric patients is further reduced through the interruptions in their daily life caused by these examinations. One of these evaluation techniques is the so called Timed-Up and Go test (TUG). The test uses a simple sequence of motions to assess the fall risk of a person. Advances in the technology of wearable sensors and machine learning make it possible to automate these evaluation methods with a compact system. This paper continues the research that was already done in the fields of human activity recognition and automatic TUG detection and proposes a novel method for the automatic detection of the Timed-Up and Go without interrupting the daily life of the patients.

## 1 INTRODUCTION

The number of geriatric patients in German hospitals alone rose by around 80 percent from 2006 to 2015 (Augurzky et al., 2017). This number will probably continue to rise in the future due to the aging population in Germany. This phenomenon can also be seen in other countries.

To make the everyday life easier for geriatric patients and the associated nursing staff, the automation of geriatric techniques and examination procedures should be investigated. One of these geriatric procedures is the so called Timed-Up and Go test (TUG). It was originally developed by Podsiadlo and Richardson in 1991 as an extension to the Get-Up and Go test (Podsiadlo and Richardson, 1991). The person to be examined stands up from a chair, walks 3 m, turns around, walks back and sits down. The time needed for this movement is measured and subsequently gives a rough measure for the fall risk of the person.

In the following we will briefly examine the research that has already been done in the fields of human activity recognition and automation of the TUG. However of the in Section 2 examined papers, none present a complete solution for the automatic detection of the TUG or they additionally use video cameras. Our approach deliberately avoids the use of cameras in favor of a less complex and more privacy friendly system.

The rest of this paper will introduce and evaluate a method for the realization of a completely automated TUG that doesn't interrupt the daily life of the patients. This is done with the help of sensor data (collected with an inertial measurement unit (IMU)) and different machine learning methods. Our scenario also represents a realistic scenario for the use of such methods in a real life situation, because the collected data for the training and evaluation of the classifiers represents a quite diverse dataset from multiple, different sources.

**Outline.** The rest of this article is structured as follows: The next Section gives an overview of related work corresponding to human activity recognition and the automated detection and segmentation of Timed-Up and Go tests. Section 3 introduces the used hardware, software and the experimental setup. The following Section 4 describes the developed procedure in detail, while Section 5 shows the experimental results. The last section gives a conclusion and also presents possible future extensions and improvements of the developed method.

## 2 RELATED WORK

Many papers have already dealt with the topics of human activity recognition and the automation of the TUG. Comparatively, approaches, where video cam-



Kroll, F., Löffler, S., Becker, I. and Hofstedt, P.
Automatic Detection of Timed-up and Go Tests with IMU Sensor Data.
DOI: 10.5220/0010775300003123
In *Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2022) - Volume 1: BIODEVICES*, pages 15-24
ISBN: 978-989-758-552-4; ISSN: 2184-4305

eras are used are considered as well. Although the proposed method forgoes the use of video cameras or other ambient sensors (see Section 3 for an explanation), they are used in a lot of other similar approaches.

The following sections first describe related work in the field of human activity recognition and secondly in the field of the automation and automatic evaluation of the TUG.

## 2.1 Human Activity Recognition

A fair amount of research has been done in the field of human activity recognition. This section presents methods with relevancy to the human activity recognition method presented in this paper.

A lot of systems use smartphones, presumably because their ease of use. With a smartphone and its inbuilt sensors, one can build a system with only one hardware component. However the reliance on a smartphone makes these systems not very fitting for the use with elderly persons or persons with dementia. One of these systems was developed by Kwapisz et al. It uses a single smartphone carried in the front pants leg pocket for data collection. Furthermore it uses a multilayer perceptron among other algorithms and reached an overall accuracy of 91.7% while classifying different daily activities (Kwapisz et al., 2011).

The research in the works of (Attal et al., 2015), (Liu et al., 2017) and (Paraschiakos et al., 2020) all concentrated on developing a system for tracking the activity of elderly people using wearable sensors. All three systems propose a system in which two or more wearables are attached to the body of the person. Attal et al. examined the use of the k-nearest neighbors (k-NN), support-vector machine (SVM), Gaussian mixture model (GMM), and random forest (RF) algorithms for the evaluation of the sensor data collected from healthy subjects. The system is able to recognize twelve different activities with k-NN reaching the highest accuracy and precision (99.25% and 98.85%, respectively) after a previous feature extraction step. Paraschiakos et al. proposed a similar approach focusing only on a random forest classifier. They found that the combination of data from an accelerometer mounted on the ankle and wrist showed the best results. The system is able to detect 16 movements with an accuracy of above 85% when additional physiological data is used. The accuracy is further increased when the number of classes is reduced, with seven classes (lying down, sitting, standing, household, walking, cycling, jumping) providing the best results.

Contrary to the other systems, (Liu et al., 2017) used a rule-based algorithm. It works by first detecting if the filtered data from the wearables represents a static posture or dynamic activity. Static postures are detected by estimating the orientation of the different wearables. The rule-based algorithm detects activities and activity transfers by examining the transitions between the postures. Overall they reached a detection rate of 97.2%.

## 2.2 Automating the TUG

To complete the previous section, this section examines papers which already dealt with the topic of an automatic segmentation of the different parts of the TUG or automatic fall risk calculation for the participating persons.

(Green, 2018) developed a method, which is able to automatically segment the data (the test was split into seven segments: Sitting, Standing-Up, Walking-Forward, Turning, Walking-Back, and Sitting-Down) of a TUG. The start of the segments could be recognized with an accuracy of 83.6% and their duration with an accuracy of 83.4%. However the system relied on video-based training data, which makes the initial data collection more time consuming.

(Nguyen et al., 2015) also developed a similar system. Two modified TUGs with an extended length of 5 m and 10 m of the walking path of the TUG were tested. The system using the longer walking path of 10 m was able to detect a simplified set of activities (standing, walking, turning, and sitting) with a sensitivity and specificity of 100%. However the system used 17 inertial motion sensors. Data from the sensors was detrended, normalized and band pass filtered. This reveals kinematic peaks, which were used to identify the different activities by picking out the minima or maxima next to these peaks.

A different approach was considered by (Seo et al., 2019). Its goal was to develop a fall prediction model based on the instrumented TUG (iTUG) (it improves the limitations of the standard TUG (Weiss et al., 2011)). 69 subjects were included in the study and performed a yearly iTUG using an IMU sensor system. All in all 26 people fell during the duration, 43 didn't experience a fall. The developed logistic regression model was able to distinguish between fallers and non-fallers with an accuracy of 69.9% and used five variables (duration of the total and the sit-to-stand phase, peak velocity of trunk sagittal plane and range of motion of trunk horizontal plane during gait phase and peak turn velocity during the turn to sit phase) for the classification.

As one can see, much research has been done in the field of automatic TUG-detection. However none of the examined systems present a fully automatic TUG detection without any user interaction.

# 3 WEARABLE AND EXPERIMENTAL SETUP

This section describes the hardware and software that was used in more detail. The following section describes the "Movesense Active" wearable used for the data collection in more detail. The second section depicts the experimental setup used for the data collection and for testing the developed method.

## 3.1 Movesense Wearable

The system uses a wearable from the company "Suunto" (Suunto, 2021). The so-called "Movesense Active" (see Figure 1) is based on a Nordic Semiconductor system on a chip and has the following main features (Suunto, 2020b).

The main system on a chip is a nRF52832 and integrates a 32-bit ARM-Cortex-M4, 64 kB of on-chip RAM, 512 kB of on-chip FLASH and a Bluetooth Low Energy (BLE) Radio. Its sensors include an accelerometer, gyroscope and magnetometer (as a combined inertial measurement unit). Other (not used) sensors are a temperature sensor and a heart rate sensor.

Further notable features are 3 Mbit of EEPROM logging memory and a status LED. The wearable is powered by an exchangeable CR 2025 Lithium coin cell battery. Its long battery runtime (up to multiple months, depending on the use case) and multiple sensors, combined with the ease of use make the Movesense Active a great choice for an autonomous movement detection system. Multiple hardware (a belt and different attachment straps) for attaching the wearable to the body is also included in the developer kit.


Figure 1: Movesense Active wearable (Suunto, 2021).

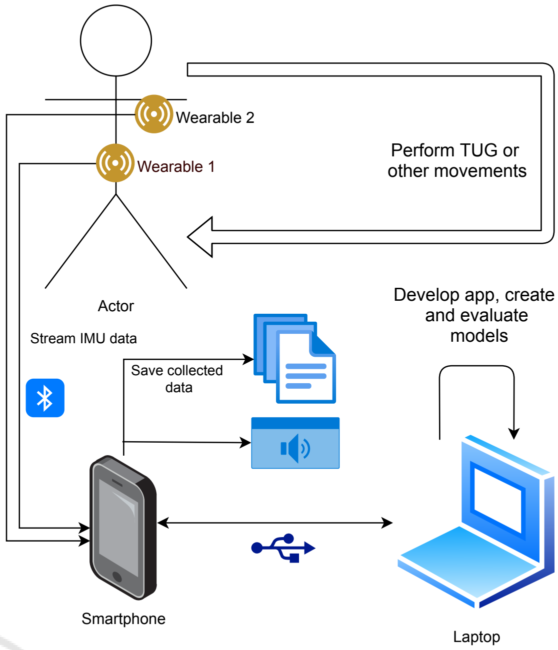


Figure 2: Visualization of the experimental setup.

As described in Section 2, other work in this field often uses additional video cameras or ambient sensors. The proposed method forgoes this choice of hardware on purpose for a more compact and easy to use system. This also makes the system more compliant with current rules of data protection and privacy, as less personal data is collected.

## 3.2 Software and Experimental Setup

More hardware and software was used in addition to the actual Movesense Wearables. This section shows the complete experimental setup. Figure 2 shows a visualization of the experimental setup which is explained in detail in the following paragraphs.

Two different smartphones (Google Pixel 3a and Xiaomi Redmi Note 5) based on the Android operating system were used for the data collection and recording via Bluetooth. However any modern Android smartphone with a BLE version of greater than 4.2 should suffice for the data collection. An Android application was developed in Android Studio and with the help of the provided Movesense software development kit (SDK) (Suunto, 2020a) to realize the data collection in a user-friendly way. The SDK offers multiple ways of interacting with the wearable. The developed application communicates with the wearable via Bluetooth and receives data from up to two Movesense wearables. The raw sensor data is then formatted and saved on the device in the CSV format.

At first a Bluetooth connection is established with up to two wearables. The wearables are worn by the user on a belt (centered on the front pelvis) and right upper arm (attached to the sleeve with the provided clip and with the wearable facing outwards). Afterwards the connection status is displayed in the mobile application and a query of the battery status is possible. The data collection can be started through a different menu. The smartphone then receives data from the multiple sensors (X,Y and Z axes from accelerometer, gyroscope and magnetometer) continuously, formats the data and saves it to a file on the smartphone. If required, a simultaneous audio recording is also possible, this function was used for evaluation of the TUG detection (see Section 5.2). The audio is recorded using the microphone of the smartphone. The data files can be subsequently read from the smartphone via a USB connection.

The application could also be expanded to a fully working prototype in a further development stage. For that, another menu could be easily integrated into the app. This menu could start or stop the automatic detection of TUGs. A second additional view could show a simple menu with an overview of the last detected TUGs and their calculated durations. In a later stage the algorithm could also be implemented on any device with enough computation capacity (for example a Raspberry Pi or other small computation devices with the required connectivity could be used as a hub and detection unit with automatic submission of the recorded TUG sequences to the medical staff).

# 4 RECOGNITION OF TUG SEQUENCES

In this section the method developed for the human activity recognition and automatic TUG detection is introduced.

The next section details the data collection and the type of data that was collected. Next the preprocessing of the data is described, including data formatting and feature extraction with the help of the Short-time Fourier transform. Finally, the human activity classification and afterwards the TUG detection algorithm are presented.

## 4.1 Data Acquisition

Data is needed for the training of the different machine learning methods and verification of the developed method. Therefore multiple datasets were collected from volunteers. This section describes the different datasets, the amount of collected data and how it was collected.

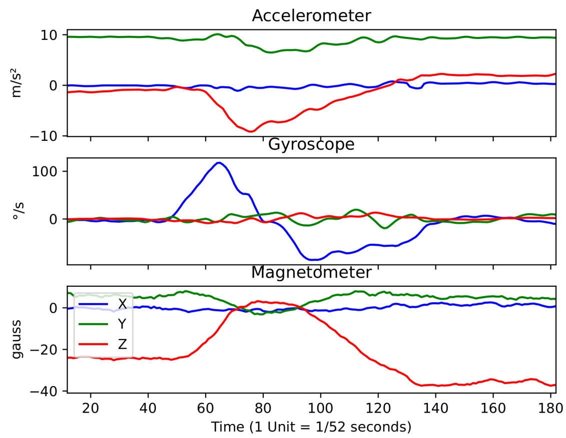


Figure 3: Accelerometer data from the movement of standing up from a chair. Recorded by a wearable worn on the belt.

Data was collected from two different sources. One of the collected datasets came from volunteers from the department of “Programming Languages and Compiler Construction” (Brandenburg University of Technology). Multiple students and staff collected the majority of the datasets and all of the audio data recordings. The other source of data was the nursing home “ASB Alten- und Pflegeheim Betriebs gGmbH Haus Abendsonne” in Frankfurt (Oder). Several elderly inhabitants collected data consisting of different movements and activities.

People were able to record different movements on their own without supervision, which could introduce possible errors in the data collection. This was done to comply with the public health guidelines resulting from the COVID-19 pandemic at the time of the experiments. The list of possible movements was: walking, sitting, standing, getting up, sitting down and complete TUG sequences. Sitting and standing were later combined into “Resting Position”. The recording of movements at the ASB in Frankfurt-Oder was supervised by the local nursing stuff, which received a brief introduction prior to the data collection.

In total data was recorded from 21 (of which eight were recorded by elderly inhabitants) different persons. All in all 432 different movement files (every file contains a single movement, e.g. walking, resting position, turning, getting up, sitting down or TUG) could be extracted from the recorded data. Figure 3 shows an example of a filtered data recording from all three sensors (accelerometer, gyroscope, magnetometer) of a user standing up from a chair (all three sensor use the same labels for the x-axis). The different phases of getting up from the chair are clearly visible in the sensor data. The wearer leans forward after sitting still, gets up, then leans backwards and finally stands still again.

The final developed algorithm only uses the data from the accelerometer and gyroscope and omits the data collected by the magnetometer. We found that the data from the magnetometer is subject to interferences in the current environment, more specifically from electronics and ferromagnetic materials and thus doesn't provide reliable enough data for training the different machine learning algorithms.

Finally, there was another type of data collected for the verification of the human activity recognition which consisted of the sensor data stream combined with an audio recording. This audio recording was synchronized with the sensor data and contained comments from the user that documented which movement were executed. Both the audio and movement files are annotated with timestamps in Universal Time Coordinated (UTC) for synchronization. Almost arbitrary movements and different TUGs were recorded in this step to allow for a realistic evaluation.

## 4.2 Preprocessing of the Data

The collected sets of raw sensor data were evaluated and the usable segments extracted. Afterwards its features were extracted with the help of the Short-time Fourier transform. This process is described in this section.

**Formatting of the Data.** First the collected data had to be surveyed and formatted manually. Sometimes the recorded data contained assumed measurement errors (spikes or discontinuities in the data) or human errors made while collecting the data. Data with such errors was discarded and not used in the training of the machine learning algorithms. Movement data was extracted from the remaining datasets. When extracting the movements, care was taken to choose roughly the same size of excerpts whenever possible. However the sizes of the excerpts range from around 74 to 514 samples ($\approx$ 1.4 s to 9.9 s at a sampling rate of 52 Hz) because of the different lengths of the movements. Datasets containing complete TUG sequences were used for the extraction of movements as well as for the verification of the automatic TUG detection.

**Sliding Window Approach and Short-time Fourier Transform.** To reduce the input size for the machine learning algorithms and to extract the most important features (namely the change of the frequency spectrum over time), the Short-time Fourier transform

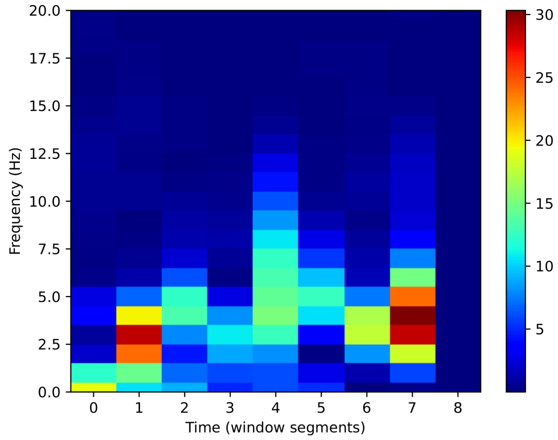


Figure 4: Visualization of a Short-time Fourier transform of data from one gyroscope axis of a single movement.

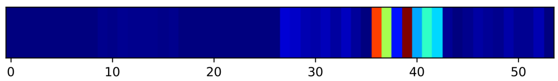


Figure 5: Complete calculated averages of the previously shown data in Figure 4 from a walking sequence. Averaged data from the accelerometer ($x = 0, \ldots, 26$) and gyroscope ($x = 27, \ldots, 53$) is shown.

(STFT) was used. The Python code used for the implementation of the STFT was adapted from (Nelson, 2014).

First, the size of the individual STFT segments is calculated by dividing the input size by four. This guarantees a similar output size for movements of different original lengths. Every movement is then split into multiple windows of this predetermined size. Afterwards we can calculate the STFT of every segment.

Figure 4 shows a visualization from an example of the Short-time Fourier transform applied to sensor data from a walking movement (specifically from one of the axis of the accelerometer) from a healthy test subject. The x-axis of the diagram represents the different segments that result from the sliding window algorithm used by the STFT. The y-axis shows the calculated intensity of the different frequency contents in each step.

Data from all six axes is then averaged, meaning that the average frequency intensity of every window segment (x-axis of Figure 4) is calculated. The example in Figure 4 results in nine average values. This is then done for every axis of the accelerometer and gyroscope, resulting in 54 values for our example (see Figure 5). This approach was inspired by the method used in (Mühle, 2019) for pre-processing audio sensor data.

All in all every movement is converted to an array containing 48 to 54 average values. Not all arrays have size 54 due to fluctuations in the size of the different movements. In that case the last entries are filled with high values to guarantee similar input sizes for the next step. These pairs (each consisting of an array and a label) can then be used for training the different machine learning models.

### 4.3 Classification of Activities

Four different machine learning algorithms were evaluated for the classification of the sensor data. All models were trained with the help of the Scikit-learn library (Pedregosa et al., 2011) for Python. More precisely, the models "RandomForestClassifier", "MLPClassifier", "GradientBoostingClassifier" and "KNeighborsClassifier" were used.

All models use the same training and test data as input (see Section 4.1) and output the determined class. All classifiers used a split of 70% for training data and 30% for test data, resulting in 130 movement files used for the evaluation and 302 files used for training the classifiers. The hyperparameters of each classifier were optimized with the help of the "RandomizedSearchCV" method. It performs a cross-validated randomized search on a specified list of possible hyperparameters. The TUG detection combines these four classifiers with the help of the "VotingClassifier", which is also provided by the Scikit-learn library (see Section 4.4 for further explanations). All classifiers have five possible classes as their output: getting up, walking, sitting down, resting, position and turning.

### 4.4 TUG Sequence Detection Algorithm

Finally, we reach the last step in our pipeline, the actual detection of TUG sequences. The developed algorithm is represented by the flow chart in Figure 6. The algorithm is split into four main parts: First the stream of sensor data is split into smaller windows with a sliding window algorithm. Then the features of each segment are extracted with the Short-time Fourier transform. After that the resulting data is classified with the VotingClassifier. Afterwards the sequence of movements is checked for a possible TUG sequence (TUG detection). The numbers in the flow chart correspond with the numbers in the following headings.

**Sliding Window based Data Segmentation (1).** First, we split the stream or file of input data into smaller segments. This is done with a sliding window approach. Multiple window sizes were tested before choosing the final window-size of 132 sampling points ($\approx 2.5\,s$). The size of the windows was determined empirically while giving care to choose roughly the same size as the movements that were used as input. The chosen window size also corresponds with the average number of movements in a standard TUG sequence. The windows are overlapping by 50% to allow for the recognition of the movements that would otherwise stretch across two windows. In the next step the segments are further analyzed with the STFT.

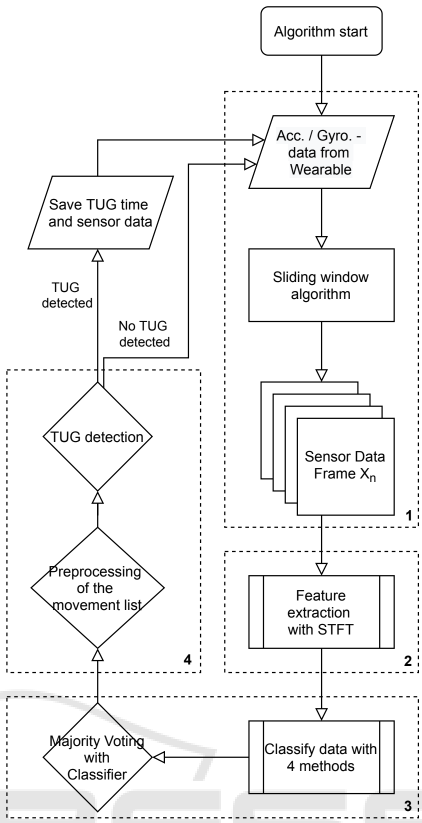


Figure 6: Flow chart of the TUG detection algorithm.

**Preparation for Classification (2).** The next step applies the previously mentioned STFT to the data. The STFT is calculated of every segment as described in Section 4.2. The data is then treated in the same way as the training and test data before and the average of every STFT is calculated and added together.

**Classification of Movements (3).** In this step, the four previously described machine learning methods (random forests, k-nearest neighbors, multilayer perceptron, gradient boosting classifier) are combined.

The VotingClassifier simply combines the four algorithms by performing a prediction using all four classifiers. After that the probabilities for each class are stored and the class with the highest value of the sums of the stored probabilities of the predictions is chosen (this is called "soft voting" by Scikit-learn). For example if the four methods produce the following summed probabilities of Getting Up $= 0.73$, Walking $= 0.9$, Turning $= 0.3$, Sitting Down $= 0.65$ and Resting Position $= 0.1$, then the class "Walking" is chosen because it has the highest overall probability. The final predicted movement is then stored in a list for further processing and the actual TUG detection.

**Actual TUG Detection (4).** Finally, the created sequence of movements from the last step is checked for a possible TUG sequence. This is relatively simple and done with the help of a list containing possible TUG sequences.

This list was constructed empirically with possible TUG sequences. Every TUG sequence in the precompiled list is then checked against the recent movements from the last step. Duplicate entries are combined beforehand (e.g. "getting up, walking, walking, walking, sitting down" $\rightarrow$ "getting up, walking, sitting down"). A TUG is found if one of the possible TUG sequences is contained in the recent list of movements. Afterwards the time the original sequence of movements took is calculated with the help of timestamps in the sensor data. (These timestamps are sent with every data packet of the sensor as UTC in milliseconds.)

# 5 EXPERIMENTAL RESULTS

The developed method was evaluated in two different ways. First the human activity recognition itself is assessed. In a second step the complete TUG detection algorithm is evaluated with the help of additional sensor and audio data.

## 5.1 Human Activity Recognition

The human activity recognition was evaluated separately from the complete TUG detection algorithm. All of the four machine learning algorithms were evaluated separately from each other as well as the combined classifier (the way they are used in the TUG detection algorithm). First we compare the overall results with the help of the precision, recall and F-score measures (see equations 1 to 3). An explanation of these measures can be found in (Sokolova and Lapalme, 2009). Afterwards the confusion matrices of the random forest, gradient boosting classifier and the combined VotingClassifier are examined in detail.

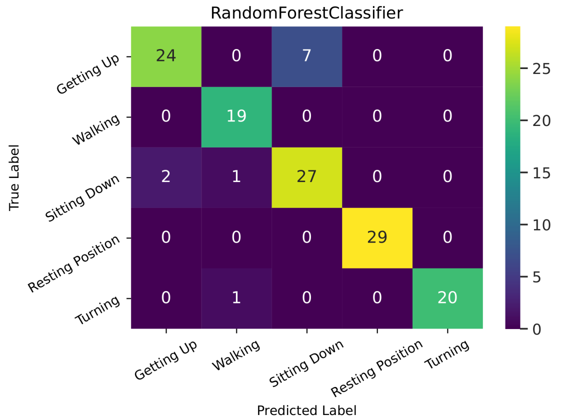


Figure 7: Confusion matrix of the human activity recognition (random forest classifier).

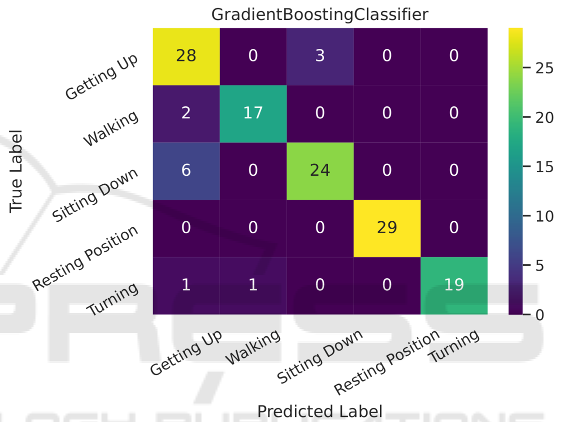


Figure 8: Confusion matrix of the human activity recognition (gradient boosting classifier).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F\text{-}measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

where:
$TP$ = True positive
$FP$ = False positive
$FN$ = False negative

All of the four classifiers were evaluated in the same way. Table 1 shows the precision, recall and F-score of all the tested machine learning methods. The scores were calculated with the help of the "precision_recall_fscore_support" method provided by the scikit-learn library. One can see that the random forest algorithm gives the best results overall (regarding precision, recall and F-score), followed by the gradient boosting algorithm. Nevertheless the k-NN and multilayer perceptron produce respectable results as

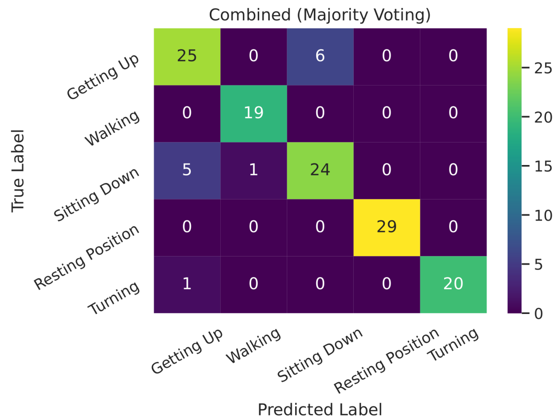


Figure 9: Confusion matrix of the human activity recognition (combined VotingClassifier).

Table 1: Precision, recall and F-score of all evaluated machine learning methods. See equations 1 to 3 for the used measures.

| | Precision | Recall | F-score |
|---|---|---|---|
| Random Forest | 92.44% | 92.53% | 92.23% |
| k-NN | 87.44% | 87.62% | 87.41% |
| Gradient Boosting | 91.8% | 90.05% | 90.7% |
| MLP | 87.28% | 86.9% | 86.87% |
| Combined | 89.69% | 89.54% | 89.44% |

well. Despite the slightly better results of some classifiers, all results are approximately located in the same range. Interestingly some of the simpler methods, like k-NN and random forest outperform the multilayer perceptron despite its much higher computational expense and bigger model size.

Figure 7 (random forest classifier) and 8 (gradient boosting classifier) show the confusion matrices of the two best performing methods. The first thing that stands out regarding the confusion matrices is that almost all errors made by the classifiers concern the confusion of getting up and sitting down, most likely due to the similarity of the sensor data of these two movements, which is amplified in the feature extraction step. However the classifiers make almost no errors regarding the other movements.

Figure 9 shows the combined results of all four classifiers (using the VotingClassifier provided by scikit-learn). Unfortunately the previously mentioned confusion of getting up and sitting down remains. However this combined classifier combines the strengths of all classifiers and is able to improve the results of the classification slightly. This is not necessarily reflected in the traditional evaluation methods, but proven true in the TUG detection results.

Table 2: Results of the TUG detection evaluation.

| | Total Time | Lab. TUGs | Det. TUGs | Other TUGs |
|---|---|---|---|---|
| Person 1 | 21 min 34 s | 14 | 9 | 5 |
| Person 2 | 4 min 36 s | 4 | 0 | 2 |
| Person 3 | 10 min 38 s | 7 | 5 | 0 |
| Person 4 | 11 min 1 s | 6 | 4 | 0 |
| Person 5 | 5 min 30 s | 4 | 1 | 0 |
| Person 7 | 29 min 57 s | 13 | 6 | 3 |
| $\sum$ | 1 h 23 min 16 s | 48 | 25 | 10 |
| Person 6 | 16 min 33 s | 14 | 1 | 0 |
| $\sum$ (with person 6) | 1 h 39 min 49 s | 62 | 26 | 10 |

It has to be mentioned that the results of the human activity recognition are highly dependent on the chosen window size (see Section 4.4). This means that probably a different size has to be chosen when using different training data or data from different age groups.

## 5.2 TUG Detection

The TUG recognition accuracy was evaluated with specially collected sensor data. We collected sensor data which was annotated by the user collecting the data (see Section 4.1). In doing so 22 datasets (with the individual lengths ranging from 1 min 55 s to 11 min 11 s) were gathered. All in all data was recorded by seven persons, every person recorded two or more datasets. These audio recordings were then parsed manually to a text file containing the movement events and their corresponding timestamps and contained multiple, random TUG sequences. These files were then compared with the results from the TUG detection algorithm. The algorithm was previously modified to be able to load a file of sensor data and check this file for possible TUG sequences.

Table 2 shows the results from the TUG detection evaluation with the help of real world data. The summed lengths of all data recordings of each person are given in the second column of the table. The third column ("Lab. TUGs") shows the number of TUGs labeled by the user. Column four ("Det. TUGs") shows the number of TUGs that were labeled by the user and recognized by the algorithm, while column five ("Other TUGs") shows TUGs which weren't labeled by the user but consist of TUG-like movements which were additionally recognized by the algorithm. Overall this shows promising results in detecting TUG sequences in the daily life of people. The algorithm misses some of the executed TUG sequences and produces a detection rate of the labeled

TUGs of 52.08% (without person 6, see last paragraph for an explanation). This is however a good result, as the main goal of the algorithm is to recognize some of the TUG-like sequences and not every single sequence executed by the person wearing the wearables. The results of the TUG recognition also seems to vary from person to person, indicating further needed research in the determination of constants like the window size. The quality of the results differs a lot from the results of the previously evaluated movement detection. This could be explained by the different data structure, movements have to be detected in a continuous data stream containing movements, whereas the movements used for training the machine learning algorithms were manually selected and trimmed.

Some movement sequences contained in the evaluation data are very similar to the movements found in a typical TUG sequence, but weren't specifically labeled as a TUG sequence by the person recording the data. When taking these additional recognized TUGs into account the algorithm is able to collect more data about possible TUGs, as shown in the last column of Table 2 (labeled as "Other TUGs"). This also yields a detection of a TUG in almost every data recording, further proving the feasibility of the algorithm. Overall the algorithm shows a good detection rate when keeping the relatively short length of the test data in mind. The algorithm should more than fulfill its goal to detect TUGs when running for at least a few hours in a real life scenario.

It has to be mentioned here that TUG sequences which are normally too long (this means sequences where the walking path is longer than 3 m) get recognized as well. This can however be easily mitigated by taking the walking speed of the person into account. One possible solution for that is explained in more detail in Section 6.1. Testing data was also only collected from healthy individuals. Further testing should also include data from older and geriatric people.

Special attention must also be given to the results of person 6. At first none of the labeled TUGs could be detected in the data. After evaluating the data further, it is suspected that the orientation of the wearables was possibly swapped when collecting the data. However, after swapping the axis of the sensor data to the suspected correct orientation of the sensors, the results didn't improve much. Due to the stark difference in result quality, we consider this an outlier. For that reason Table 2 includes a second summary without the results of person 6. While we suspect a technical or human error due to the unsupervised experimental setup, further research with more participants and a supervised experiment should bring confirmation to our suspicion.

# 6 CONCLUSION AND FUTURE WORK

In this paper we presented a method for an automatic TUG detection that works without interrupting the daily life of the patients. This section first presents a summary of the findings and results. The last section shows some ways in which the developed method could be improved.

The evaluation of the human activity recognition showed very promising results with the performance metrics of precision and recall exceeding values of 90%. The only notable errors are made concerning the two movements of getting up and sitting down. The human activity recognition can not only be used for the developed TUG detection, but could also be used for a standalone activity recognition of elderly or geriatric persons. The actual TUG detection delivers good results as well. More than half of the performed and labeled TUG sequences are recognized. Movement sequences with high similarity to the TUG (but which weren't labeled by the testers) are recognized as possible TUG sequences as well.

The only downside being that the method also recognizes sequences which contain walking movements longer than 3 m. A possible solution for this problem is briefly described in Section 6.1. All in all it is definitely possible to detect and evaluate TUG-like sequences without disrupting the daily life of the patients or additional user inputs. Nevertheless some improvements are required for a fully autonomous system. Some of these possible improvements are shown in the next section.

## 6.1 Future Work

All in all the developed method represents a good way for the automatic detection of the TUG. Nonetheless, there is room for improvement and extension of the presented method. This section describes how the method could be further improved.

At first, the developed method should be ported to and tested on a standalone hub (for example a Raspberry Pi, see Section 3.2). This would remove the need for a smartphone and make the system more autonomous. It also increases the usability for different age groups and for people with dementia.

The developed method has the disadvantage that it detects TUG sequences that are longer than 3 m (see

Section 5.2). This could be mitigated by also calculating the walking speed of the wearer of the wearable from the collected IMU data. This could be done with the help of the accelerometer data (for example by detecting the frequency of peaks in the accelerometer data of the walking sequences). One can then calculate the walking distance when the walking speed and the length of the walking sequence is known. Afterwards the algorithm could discard detected TUG sequences containing walking lengths which deviate too much from the standard walking length of 3 m. Libraries like GaitPy (Czech and Patel, 2019) for Python also provide methods for the extraction of gait characteristics from accelerometer data.

Our developed method is also limited to the detection of the individual phases of the TUG and the time the user took to complete the test. A future expansion of the algorithm could also calculate and evaluate further fall risk indicators. For example the movement or turn speed in every phase of the detected TUG could be calculated. Every phase can then be further evaluated with the help of a machine learning algorithm and expert knowledge from medical staff with experience in the field of fall risk assessment. For this more test data from geriatric persons would be needed. In combination with this the TUG sequence could also be displayed visually for a more intuitive interpretation. Approaches, for example from (Seo et al., 2019) already tried to use an extended TUG to discern people with higher fall risk from those with a lower fall risk. The system measures multiple variables in each of the single phases of the TUG. These in turn are analyzed with regression to determine the fall risk.

## ACKNOWLEDGEMENTS

Research reported in this publication was partially supported by the Central Innovation Programme for small and medium-sized enterprises (SMEs) of the German Federal Ministry for Economic Affairs and Energy under grant number 16KN075223 (MoDiSeM - Spur).

## REFERENCES

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., and Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors*, 15(12):31314–31338.

Augurzky, B., Hentschker, C., Pilny, A., and Wübker, A. (2017). *Krankenhausreport*. Barmer, Berlin.

Czech, M. and Patel, S. (2019). Gaitpy: An open-source python package for gait analysis using an accelerometer on the lower back. *Journal of Open Source Software*, 4:1778.

Green, A. M. (2018). Automatic “Timed-Up and Go” (TUG) Test Segmentation. Master’s thesis, Massachusetts Institute of Technology.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82.

Liu, J., Sohn, J., and Kim, S. (2017). Classification of daily activities for the elderly using wearable sensors. *Journal of Healthcare Engineering*, 2017:1–7.

Mühle, M. (2019). Entwicklung eines Lernverfahrens zur Beurteilung von Kompressoren anhand akustischer Signale. Master’s thesis, BTU Cottbus-Senftenberg.

Nelson, K. (2014). Short time fourier transform using python and numpy. https://kevinsprojects.wordpress.com/2014/12/13/short-time-fourier-transform-using-python-and-numpy/. Accessed: 02.09.2021.

Nguyen, H., Ayachi, F., Rahimi, F., Boissy, P., Jog, M., Blamoutier, M., and Duval, C. (2015). Auto detection and segmentation of physical activities during a timed-up-and-go (tug) task in healthy older adults using multiple inertial sensors. *Journal of NeuroEngineering and Rehabilitation*.

Paraschiakos, S., Cachucho, R., Moed, M., van Heemst, D., Mooijaart, S., Slagboom, E., Knobbe, A., and Beekman, M. (2020). Activity recognition using wearable sensors for tracking the elderly. *User Modeling and User-Adapted Interaction*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Podsiadlo, D. and Richardson, S. (1991). The timed up and go: a test of basic functional mobility for frail elderly persons. *Journal of the American Geriatrics Society*, 39(2):142—148.

Seo, J., Kim, T., Lee, J., Kim, J., and Tack, G. (2019). Fall prediction of the elderly with a logistic regression model based on instrumented timed up & go. *Journal of Mechanical Science and Technology*, 33.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Suunto (2020a). Movesense-device-lib. https://bitbucket.org/suunto/movesense-device-lib/src/master/. Accessed: 15.07.2021.

Suunto (2020b). *Movesense Sensor*. Suunto, Vantaa, Finland. v2.0.

Suunto (2021). Movesense - Open Wearable Tech Platform. https://www.movesense.com/. Accessed: 15.07.2021.

Weiss, A., Herman, T., Plotnik, M., Brozgol, M., Giladi, N., and Hausdorff, J. (2011). An instrumented timed up and go: The added value of an accelerometer for identifying fall risk in idiopathic fallers. *Physiological measurement*, 32:2003–18.