Automated classification of pre-defined movement patterns: A comparison between GNSS and UWB technology

Rodi Laanen[1][0000−0003−3544−0914], Maedeh Nasri[2][0000−0002−8547−0946], Richard van Dijk[1][0000−0002−5796−7883], Mitra Baratchi[1][0000−0002−1279−9310], Alexander Koutamanis[3][0000−0002−0355−1276], and Carolien Rieffe[2,4][0000−0002−7584−6698]

[1] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

[2] Unit of Developmental and Educational Psychology, Institute of Psychology, Leiden University, Leiden, The Netherlands

[3] Department of Management in the Built environment, Delft University of Technology, Delft, The Netherlands

[4] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands

Abstract. Advanced real-time location systems (RTLS) allow for collecting spatio-temporal data from human movement behaviours. Tracking individuals in small areas such as schoolyards or nursing homes might impose difficulties for RTLS in terms of positioning accuracy. However, to date, few studies have investigated the performance of different localisation systems regarding the classification of human movement patterns in small areas. The current study aims to design and evaluate an automated framework to classify human movement trajectories obtained from two different RTLS: Global Navigation Satellite System (GNSS) and Ultra-wideband (UWB), in areas of approximately 100 square meters. Specifically, we designed a versatile framework which takes GNSS or UWB data as input, extracts features from these data and classifies them according to the annotated spatial patterns. The automated framework contains three choices for applying noise removal: (i) no noise removal, (ii) Savitzky Golay filter on the raw location data or (iii) Savitzky Golay filter on the extracted features, as well as three choices regarding the classification algorithm: Decision Tree (DT), Random Forest (RF) or Support Vector Machine (SVM). We integrated different stages within the framework with the Sequential Model-Based Algorithm Configuration (SMAC) to perform automated hyperparameter optimisation. The best performance is achieved with a pipeline consisting of noise removal applied to the raw location data with an RF model for the GNSS and no noise removal with an SVM model for the UWB. We further demonstrate through statistical analysis that the UWB achieves significantly higher results than the GNSS in classifying movement patterns.

Keywords: Spatio-temporal data · Automated machine learning · Supervised learning · GNSS · UWB.

1 Introduction

Advanced localisation systems have enabled capturing trajectories of moving entities providing insights in various areas such as understanding periodic behavioural patterns [2], pedestrian movement dynamics [27], and anomaly detection [33]. Using such positioning systems in enclosed spaces helps uncover social dynamics information, for instance by tracking the movement behaviours of children in a schoolyard during recess [20, 25] or the elderly in nursing homes [18, 23].

Global Navigation Satellite Systems (GNSS) are one of the most recently used localisation systems which demonstrated strong promises in urban computing and transportation [6, 8]. However, a GNSS might not provide the desired positioning accuracy in small areas such as schoolyards and nursing homes. Namely, under ideal circumstances, GNSS devices can achieve a positioning accuracy of approximately 2-3 meters [7, 35], and the positioning error can deteriorate up to 20 meters or more due to, for instance, multi-path or ionospheric scintillation [7, 14]. To mitigate this problem, several studies [16, 21, 24] have demonstrated promising results regarding the positioning accuracy through Ultra-wideband (UWB) technology by reporting positioning accuracies between a 10-15 cm range.

In addition to positioning, GNSS and UWB trajectories can be used to analyse human movement behaviour and unravel complex spatio-temporal patterns. Extracting these patterns can provide useful insights in, for instance, children's use of space and interactions with the schoolyard environment during recess [25] and the development of personalised geofences for elderly people suffering from Alzheimer's disease [18]. However, to the best of our knowledge, there is no study directly comparing the performance of these two technologies in a movement pattern classification task.

To address this gap, the current paper aims to design an automated framework to classify pre-defined human movement patterns collected with GNSS and UWB technologies and to evaluate their respective classification performances. We will address the following questions: (i) How to design an entire pipeline to collect and annotate the GNSS and UWB spatial-temporal data, pre-process and classify them correctly? (ii) How to select the optimum hyperparameter configurations for the data segmentation, noise removal and classification stages within the pipeline which result in the best performance? (iii) Which technology, GNSS versus UWB, achieves higher results in terms of movement pattern classification in small areas of approximately 100 square meters? Moreover, this paper contains the following contributions:

- We collected and introduced a GNSS and a UWB data set both consisting of 104 trajectories based on 4 annotated pre-defined human movement patterns.
- We designed an automated framework which aims to correctly classify pre-defined movement patterns collected with GNSS and UWB technology. The automated framework contains three options for applying noise removal: (i) no noise removal, (ii) Savitzky Golay filter on the raw location data [8] or (iii) Savitzky Golay filter on the extracted features [6] as well as three options regarding the classification model: Decision Tree (DT), Random Forest (RF) or Support Vector Machine (SVM).

- We performed hyperparameter optimization to automate the full trajectory processing pipeline, including data preparation, noise removal and classification stages for both UWB and GNSS trajectories.

The remainder of this paper is organized as follows: In Section 2, we give a brief overview of related works. Section 3 contains the overview of our method. The details of our experiments and the evaluation of the proposed method are presented in Section 4. Lastly, the conclusion and future research directions are discussed in Section 5.

2 Related works

Recent studies compared the positioning accuracy of GPS and UWB data within the sport domain [4, 38]. Waqar et al. [38] compared the positioning accuracy from a GPS and a UWB system on a tennis field. Based on their quantitative analysis, the authors concluded that UWB outperforms GPS in terms of localisation accuracy [38]. In the study of Bastida et al. [4], the two technologies were assessed on a soccer field where the UWB system proved to achieve higher localisation accuracy compared to the GPS system [4]. Despite the interesting results found in these two studies on positioning accuracy and performance, they do not directly address movement pattern classification tasks performed through GNSS and UWB systems.

Several studies focused on classification of transportation modes via GPS trajectories by designing a competitive framework with classical machine learning algorithms [8], and one with a Convolutional Neural Network [6]. Another interesting framework which clustered basic patterns via deep representation learning is proposed in the work of Yao et al. [40]. However, the aforementioned three studies [6, 8, 40] are restricted to GNSS trajectories as input and not adapted to UWB trajectories, and secondly, hyperparameters present within each respective framework are either determined manually or without motivation and do not include automated hyperparameter optimization.

3 Methodology

In this section, we first introduce the general outline of the automated framework (see Figure 1). Next, we discuss the following parts of the automated framework in more detail: raw trajectories, data preparation, features, trajectory features, noise removal and normalization.

3.1 Automated framework

The proposed automated framework, depicted in Figure 1, inspired by Etemad et al. [8], automatically takes raw discrete trajectories collected with GNSS or UWB technology as input, pre-processes the raw spatio-temporal data and feeds the obtained data to a classifier. Additionally, we investigated the effect of a noise removal filter, i.e., Savitzky Golay filter which is discussed in Section 3.1, on the performance of the entire automated framework by implementing three pipeline configurations in parallel: (i) extracting features without applying noise removal, (ii) applying a Savitzky Golay filter directly on the raw location data [8] or (iii) extracting features from the raw spatio-temporal data followed by applying a Savitzky Golay filter on the extracted features [6]. The parallel configurations are denoted by green, red and blue lines in Figure 1, respectively. Furthermore, the automated framework includes three classification algorithms: DT, RF or SVM. In this respect, the automated framework yields nine possible pipeline configurations.

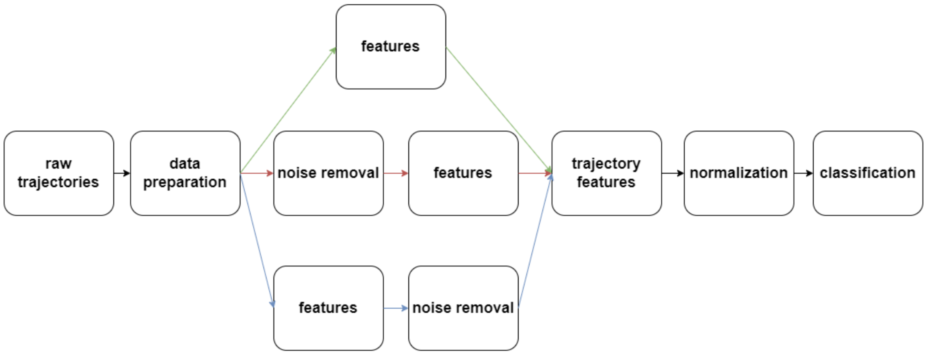


Fig. 1: Automated framework, inspired by Etemad et al. [8], which takes raw spatial-temporal data from GNSS and UWB systems and ultimately learns from the corresponding annotated movement patterns. Applying no noise removal, the Savitzky Golay filter on the raw location data and the Savitzky Golay filter on the features are indicated by the green, red and blue lines, respectively.

As indicated in Figure 1, the automated framework contains 7 distinct steps: 1) raw trajectories, 2) data preparation, 3) features, 4) noise removal, 5) trajectory features, 6) normalization and 7) classification. These steps, except for the classification stage, are explained in detail in the following subsections.

Raw trajectories: Raw trajectories are acquired from localisation systems (i.e., GNSS and UWB) within R timestamps. Each discrete raw trajectory $traj$ contains the spatio-temporal list of points of one moving object as follows:

$$traj = \{P_1, P_2, ..., P_R\},\ \ P_r = (c1_r, c2_r, t_r),\ \ r \in [1, R] \tag{1}$$

where P_r defines the 2-D position of the object at timestamp t_r. For GNSS trajectories each position is equal to $(c1_r = lat_r, c2_r = lon_r, t_r = t_r)$ and for UWB trajectories each position is in Euclidean space as $(c1_r = x_r, c2_r = y_r, t_r = t_r)$. The number of positions R of the GNSS trajectory can be smaller than the number of positions R of the UWB trajectory because of the difference in data rate (1 Hz versus around 6 Hz for UWB).

Data preparation: The data preparation part of the automated framework involves segmenting a raw discrete trajectory $traj$ into M segments with varying size R_m. Specifically, a raw discrete trajectory $traj$ is split into $traj \mod M$ segments with size $\lfloor traj/M \rfloor + 1$ and the remaining segments with size $\lfloor traj/M \rfloor$. Hence, the segments do not overlap time-wise. The whole set of segments covers the trajectory:

$$traj = \{traj_1, traj_2, ..., traj_M\} \tag{2}$$

Features: From each segment of a GNSS or a UWB trajectory, we extract three features: average velocity, change of velocity and change of angle since these features capture alternations of movement behaviours [40]. Regarding a GNSS trajectory segment, each record $r \in [1, R_m]$, where R_m refers to the total number of records within this GNSS trajectory segment, contains the GNSS coordinates in (lat_r, lon_r), and the corresponding timestamp t_r, from which the average velocity is computed as follows:

$$v_r = \frac{haversine((lat_{r+1}, lon_{r+1}), (lat_r, lon_r))}{t_{r+1} - t_r}, \quad v_1 = 0 \tag{3}$$

where v represents the average velocity, t_r describes the timestamp $r \in [2, R_m]$ indicates a single record of a trajectory segment and lat_r and lon_r denote the latitude and longitude, respectively. By first converting the latitude and longitude coordinates to an Euclidean plane projected on the surface of the earth, the $haversine$ function determines the Euclidean distance between two consecutive GNSS points.

Additionally, we computed the angle between two subsequent GNSS points by:

$$X_r = (0, lon_r), \quad X_{r+1} = (0, lon_{r+1}), \quad Y_r = (lat_r, 0), \quad Y_{r+1} = (lat_{r+1}, 0) \tag{4}$$

$$a_r = arctan2(haversine(Y_{r+1}, Y_r), haversine(X_{r+1}, X_r)), \quad a_1 = 0 \tag{5}$$

where a_r denotes the angle between record $r+1$ and r and $r \in [2, R_m]$.

For each discrete UWB trajectory segment, a single record, $r \in [1, R_m]$, where R_m refers to the total number of records within this UWB trajectory segment m, contains the position of an active tag in (x_r, y_r) coordinates with corresponding timestamp t_r. We computed the average velocity v_r according to the following equations:

$$v_r = \frac{d_r}{t_{r+1} - t_r}, \quad v_1 = 0, \quad d_r = \sqrt{(x_{r+1} - x_r)^2 + (y_{r+1} - y_r)^2} \tag{6}$$

where x and y denote the x and y coordinates, respectively. d_r refers to the Euclidean distance between record $r+1$ and r and $r = [2, R_m]$. Besides, the angle between two consecutive UWB points is calculated as follows:

$$a_r = arctan2((y_{r+1} - y_r), (x_{r+1} - x_r)), \quad a_1 = 0 \tag{7}$$

Regarding the three features: average velocity, change of velocity and change of angle, we computed the average velocity of the discrete GNSS trajectory segments with Equation 3 and for the discrete UWB trajectory segments with Equations in 6. For both the discrete GNSS and UWB trajectory segments, we computed the change of velocity and change of angle for two subsequent records as follows:

$$v_{\Delta_r} = v_r - v_{r-1}, \quad v_{\Delta_1} = 0 \tag{8}$$

$$a_{\Delta_r} = a_r - a_{r-1}, \quad a_{\Delta_1} = 0 \tag{9}$$

where $v_{\Delta r}$ denotes the change of velocity and $a_{\Delta r}$ represent the change of angle, both for $r \in [2, R_m]$.

Trajectory features: We utilized statistical measures which provide information about the entire trajectory $traj$ or about segments $traj_m$ of the trajectory in question as described by Etemad et al. [8]. In this respect, we selected these features to gain global information about the (segment) trajectories. For the whole (segment) trajectory, 5 global features [8] per set of extracted features were derived: the minimum, maximum, mean, median, and standard deviation. Additionally, we computed 5 local features [8] per set of extracted features: five percentiles (10, 25, 50, 75, 90). Hence, the global trajectory is extended with 3 extracted features x 5 statistical measures, while a local trajectory segment is extended with 3 extracted features x 5 percentile measures.

Noise removal: Since both RTLS use electromagnetic waves, the collected trajectories might include positional errors due to e.g., conductors (e.g., metal and water), air humidity, multi-path interference and battery levels [7,14,16,28]. A promising candidate is the Kalman Filter [3,13,15] that has been used for several purposes such as noise removal. However, according to Kennedy [15], the Kalman Filter can only find an optimal solution if the noise within a signal is normally distributed. Given that the shape of the underlying noise distribution of the raw spatio-temporal data or the extracted features is unknown [6], the Kalman Filter might not provide optimal results in terms of noise removal. The Savitzky Golay filter mitigates this issue since it does not assume a distribution of the signal's noise [6,15]. Another advantage is that the pattern of the original signal will be maintained after applying the Savitzky Golay filter [6]. Therefore, the Savitzky Golay filter [31] is adopted in the noise removal stage.

This filter selects a subset of data points around the target point defined by the sliding window size and fits a polynomial on the respective subset [32]. As a final step, the target point gets substituted for the computed outcome of the polynomial [32]. Both the sliding window and the degree of the fitted polynomial are crucial hyperparameters, which will be further discussed in Section 3.2.

Normalization: Min-max normalization transforms all features into the range $[0, 1]$ by mapping the lowest value to 0, the highest value to 1 and the remaining values between 0 and 1. In this way, all generated features will be projected to the same range ($[0,1]$), while the relationship between the original feature set values is preserved [12].

3.2 Automated hyperparameter optimisation

Previously mentioned studies [6, 8, 40], which inspired our design of the automated framework, manually performed hyperparameter Optimisation (HPO). A manual approach demands a thorough understanding regarding the impact of each hyperparameter and can become a time-consuming and inefficient task which might result in less optimal and accurate outcomes [9]. In the current study, Sequential Model-based Algorithm Configuration (SMAC) [19] is adopted to make the HPO automated for the data preparation, noise removal, and classification stages.

SMAC is a general-purpose HPO library based on the Bayesian optimization framework [19]. It allows for optimizing various types of hyperparameters (i.e., categorical, numerical and conditional). When dealing with a full pipeline, flexibility regarding the implementation of hyperparameter types is extremely crucial. Notably, design choices such as the search space benefit from the hyperparameter type flexibility provided by SMAC. Thereby, SMAC includes the option to define conditional hyperparameters which allow for selecting, for instance, the type of machine learning algorithm.

3.3 Selection of hyperparameters

To optimize the pipeline, we need to define the search space incorporating all hyperparameters. First, the data preparation stage contains a hyperparameter that specifies the segmentation size of each discrete trajectory $traj_m$. Second, two hyperparameters are present in the noise removal stage: one that determines the respective sliding window-size of the targeted data points and another one which optimizes the degree of the fitted polynomial. Last, in the classification stage, we automatically let the hyperparameters of each respective classifier be optimized. Via SMAC, we aim to achieve higher classification performances assessed with the used score metrics in this study. Detailed information about the hyperparameters and their respective types and value ranges are described in Table 1.

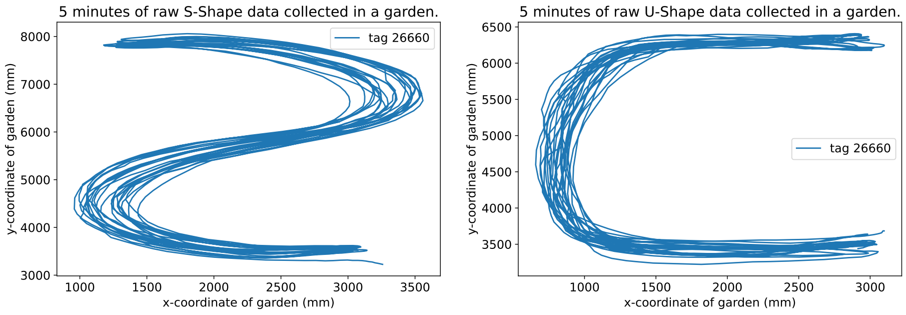
4 Experiments

In this section, first, the data collection sessions under controlled conditions are described. Next, the performance metrics to assess each pipeline configuration and the experimental settings including the bootstrapping protocol are introduced. Finally, we will report and interpret the obtained results in our experiments. All experiments are implemented in Python 3.9.7[5] with scikit-learn 1.0.2[6], SciPy 1.7.1[7] and SMAC1.3.3[8]. The source code from this study is available on GitHub[9].

4.1 Data sets

We collected spatio-temporal GNSS data points (i.e., a list of (*longitude*, *latitude*, *timestamp*)) with an 1 Hz sampling rate using a Samsung Galaxy Tab A7 (2020) SM-T500 64 GB tablet [30] and the Sensor Logger software[10]. We used the Pozyx Creator Kit [29] to collect UWB data points (i.e., a list of (x, y, *timestamp*)) with an average sampling rate of 5.91 Hz per active tag. All practical details regarding the GNSS and UWB system configurations and the UWB implementation are described in [17].

During the data collection sessions, we adopted the same movement patterns as Yao et al. [40], i.e., *Straight* and *Circling* and further extended the *Bending* pattern by defining two new patterns: *S-Shape* and *U-Shape*, as represented in Figure 2. We opted for the *S-Shape* and *U-Shape* as *Bending* patterns since they can be composed of parts of the *Straight* and *Circling* patterns, resulting in four different classes to be recognized, and making the classification task more challenging.



(a) *S-Shape*

(b) *U-Shape*

Fig. 2: Visualizations of 5 minutes of (a) *S-Shape* and (b) *U-Shape* patterns collected via UWB technology.

We collected trajectory data consisting of the movements of four adults who walked a single pattern continuously for 5 minutes by holding a GNSS and a UWB

[5] https://www.python.org

[6] https://scikit-learn.org

[7] https://docs.scipy.org

[8] https://github.com/automl/SMAC3

[9] https://github.com/rlaanen/uwbtrajectorypatterns

[10] https://www.tszheichoi.com/sensorlogger

recording device. In this respect, we obtained two data sets each consisting of 19 *Straight*, 25 *circling*, 30 *S-Shape* and 30 *U-Shape* movement patterns, resulting in a total of 104 5-minute-long discrete GNSS and another 104 5-minute-long UWB trajectories.

4.2 Performance metrics

Four performance metrics were implemented to assess the performance of the trained model on the test data set. Throughout our experiments, we considered the following four metrics: macro precision, macro recall, macro F_1 and Matthew's Correlation Coefficient (MCC). We opted for the macro F_1 score since macro-averaging treats all four patterns on an equal basis by being insensitive to class imbalance [34]. Furthermore, the F_1 score represents the harmonic mean of the precision and recall. Therefore, we included the macro precision and macro recall score metrics. Thereby, we adopted an extended version of the MCC metric which applies to multi-class classification tasks [10]. This score is formulated as follows:

$$MCC = \frac{c \cdot s - \sum_k^K p_k \cdot b_k}{\sqrt{(s^2 - \sum_k^K {p_k}^2) \cdot (s^2 - \sum_k^K {b_k}^2)}} \tag{10}$$

where c represents the total number of correctly predicted instances, s refers to the total number of instances, K is the total number of classes, k refers to a single class, b_k denotes the true occurrences of class k and p_k describes the number of times class k was predicted [26]. This equation explicitly reflects that a high-performance score is correlated with the number of correctly predicted instances [11]. Additionally, the MCC is reported to be robust against potential performance bias from an imbalanced data set [5].

4.3 Experiment settings

As discussed in Section 3, the data preparation, noise removal and classifier steps within the automated framework are controlled by hyperparameters. We decided to automate the HPO process with SMAC [19]. All hyperparameters including their respective value range and type are represented in Table 1.

Although SMAC automates the HPO, it remains a time-consuming task [9] and setting no random-seed in the SMAC framework makes the optimisation non-deterministic [19]. For these reasons, we first assessed the relationship between the wallclock-time and the MCC score in order to obtain the optimum wallclock-time. Second, we implemented an adaptation of the bootstrapping protocol proposed in the work of Wang et al. [36] to obtain sufficient and fair distributions of results and apply a significance test afterwards.

The wallclock-time represents the time in seconds between starting the SMAC optimisation and terminating it. In order to find the wallclock-time SMAC needs to find near-optimal hyperparameter values, we proceeded as follows: First, we split the entire data set into 67% training and 33% test set, and then, with

Table 1: hyperparameters including their respective value ranges and types influencing the data preparation, noise removal and classifier steps within the automated framework. Given that the Savitzky Golay filter imposes an odd-sized window, we opted for a Categorical type of hyperparameter.

Method	**Hyperparameter**	**Value Range**	**Type**
Partitioning Instances	split	[1, 10]	Uniform Integer
Savitzky Golay filter	window_length	[1, 3, ..., 27, 29]	Categorical
	polyorder	[1, 10]	Uniform Integer
Decision Tree	max_depth	[5, 50]	Uniform Integer
	min_samples_leaf	[1, 10]	Uniform Integer
	min_samples_split	[2, 10]	Uniform Integer
	criterion	[”gini”, ”entropy”]	Categorical
Random Forest	n_estimators	[5, 100]	Uniform Integer
	min_samples_leaf	[1, 10]	Uniform Integer
	max_depth	[5, 50]	Uniform Integer
	min_samples_split	[2, 10]	Uniform Integer
	criterion	[”gini”, ”entropy”]	Categorical
Support Vector Machine	C	[0.1, 100]	Uniform Float
	kernel	[”linear”, ”poly”, ”rbf”, ”sigmoid”]	Categorical

increment-steps of 25 seconds, we performed 15 independent optimisation runs per wallclock-time on the pipeline configuration yielding the most hyperparameters (i.e., noise removal with an RF model). This method resulted in a wallclock-time of 500 seconds at which the highest performance reflected in the MCC score was reached. This optimal value is adopted in the bootstrapping protocol.

A single pipeline optimization within the bootstrapping protocol [36] consists of the following steps: First, we split the entire data set again into a 67% training and 33% test set after which we exclusively utilise the training set to perform the optimisation with SMAC. For the validation method, we opted for a stratified 10-fold cross-validation with an MCC score metric since k-fold cross-validation segments the training data set in folds, each containing approximately the same distribution of class labels as in the training data set [26]. Given that SMAC targets HPO as a minimisation problem [19], we aggregated over the 10 resulting performances and subtracted the outcome from 1 as an indicator of the respective pipeline configuration performance.

In this respect, we let SMAC optimise a single pipeline configuration 15 times on the training data set. Next, from the 15 generated ones, we randomly sample 5 incumbents from which we select the one yielding the lowest performance score. We repeated this process 50 times for every pipeline configuration. Subsequently, we retrained 50 models according to the 50 best-found incumbents on the training data set. Finally, we evaluated the 50 models on the independent test set and averaged over the 50 scores.

4.4 Results

In this section, we first describe the GNSS and UWB results separately. Subsequently, we compare the two technologies according to the pipelines which achieved the highest performance scores.

GNSS results: Via the above-mentioned bootstrapping protocol in Section 3, we optimised the nine possible pipeline configurations on the GNSS data set. The averaged scores on the test data set reflected in macro precision, macro recall, macro F_1 and MCC which are based on the 50 best-found hyperparameter settings according to the bootstrapping protocol are demonstrated in Table 2.

Table 2: GNSS results on the test data set for each pipeline configuration. The precision, recall and F_1 scores are calculated on a macro basis. The asterisk indicates that this particular pipeline configuration gives statistically higher results

		Metric			
Model	**Noise Removal**	precision	recall	F_1	MCC
DT	no noise removal	48.52±3.34	46.38±1.86	45.08±2.24	27.08±3.14
	noise removal on raw location data	57.26±11.91	56.48±10.55	54.64±11.18	40.94±14.39
	noise removal on features	56.37±11.38	55.58±9.03	54.33±9.60	39.61±11.82
RF	no noise removal	46.89±5.08	47.30±4.47	44.10±5.94	28.83±5.72
	noise removal on raw location data	**62.04±4.19***	**66.88±4.36***	**62.31±4.18***	**50.13±5.73***
	noise removal on features	58.74±8.83	60.06±8.00	57.47±7.79	44.75±10.60
SVM	no noise removal	46.10±4.14	47.03±3.84	44.33±4.61	27.46±5.36
	noise removal on raw location data	59.10±7.81	59.21±4.94	57.25±6.12	44.89±7.00
	noise removal on features	55.96±7.15	55.14±6.31	53.63±7.26	38.46±8.04

We performed a statistical significance test with a significance level of 0.05 as described in the work of Wang et al. [37] to determine if a certain pipeline configuration gave statistically higher performance than another one. First, we checked if the underlying distribution of each set of 50 performance scores was uniformly distributed. By performing an Anderson-Darling test [1], we concluded that the sets were not sampled from a Gaussian distribution. Given that the distributions were extracted from the same population, we decided to utilise the non-parametric Wilcoxon signed-rank test [39].

Based on Table 2, the pipeline configuration yielding noise removal on the raw location data in combination with an RF model gives statistically higher results for the precision, recall, F_1 and MCC scores than the other possible pipeline combinations. Thus, for the GNSS, the highest performance over all four score metrics was achieved with a pipeline configuration yielding noise removal on the raw location data in combination with an RF model. Additionally, the advantage of applying the Savitzky Golay filter on the raw location data or the extracted features in terms of acquiring a higher performance accuracy is reflected in the results demonstrated in Table 2. For all four score metrics, the lowest scores were obtained with pipelines yielding no noise removal application.

UWB results: Analogously, we assessed the 9 pipeline configurations on the collected UWB data. The Anderson-Darling test confirmed that the sets per pipeline configuration were not Gaussian distributed. For this reason, we used the Wilcoxon signed-rank test for the statistical analysis. The averaged test data set results over the 50 independent retrained models are reported in Table 3 in which bold-faced scores again indicate the highest performance and additional asterisks refer to the pipeline configuration that performs statistically better than other pipeline configurations per metric:

Table 3: UWB results on the test data set for each pipeline configuration. The precision, recall and F_1 scores are calculated on a macro basis.

		Metric			
Model	**Noise Removal**	precision	recall	F_1	MCC
DT	no noise removal	80.28±5.19	79.14±5.65	78.87±5.83	71.90±7.49
	noise removal on raw location data	78.97±7.20	76.79±8.01	75.56±9.01	70.02±10.61
	noise removal on features	74.24±10.24	69.22±13.04	68.19±14.04	60.81±16.26
RF	no noise removal	86.78±1.95	86.00±2.52	85.87±2.35	81.39±3.01
	noise removal on raw location data	85.78±5.76	**89.11±3.87**	85.31±6.53	82.06±7.07
	noise removal on features	83.15±5.54	71.15±11.00	68.72±13.18	67.73±11.74
SVM	no noise removal	89.74±1.64	87.24±3.09	**85.92±3.60**	**82.85±3.91**
	noise removal on raw location data	**89.95±3.54**	86.21±6.73	84.99±8.11	82.32±8.05
	noise removal on features	89.68±3.51	83.17±7.83	83.82±8.23	80.01±8.87

Based on Table 3, we can conclude that (i) noise removal on the raw location data with an SVM gives the highest results for the precision, (ii) noise removal on the raw location data with an RF gives the highest results for the recall, and (iii) no noise removal with an SVM gives the highest results for the F_1 and MCC. However, these particular pipeline configurations are not statistically the ones with the best performance per score metric. Therefore, we opted for the pipeline configuration that yields the highest results across the majority of score metrics, which is the configuration consisting of no noise removal in combination with an SVM.

Comparison between GNSS and UWB: Regarding the GNSS, we concluded that the pipeline configuration containing noise removal on the raw location data with an RF model gave the highest performance results for all four performance metrics based on Table 2. As for the UWB, we concluded that the pipeline configuration consisting of no noise removal with an SVM model gave the highest performance results for the majority of performance metrics (i.e., F_1 and MCC) based on Table 3. Therefore, we decided to compare the GNSS versus the UWB technology based on these respective pipeline configurations. Given that the sampled results per pipeline configuration are not uniformly distributed and we want to compare samples from two different populations, we opted for the non-parametric Mann-Whitney U test [22] with a standard confidence level of 95%.

For the precision, we observed a positive effect for the UWB with a significance level of 99%. As for the recall, an identical positive effect for the UWB could be observed yielding a significance level of 99%. Regarding the F_1 score, a positive effect could be observed in favour of the UWB with a significance level of 99%. Likewise, for the MCC metric, a positive effect with a significance level of 99% was present for the UWB. Thus, the UWB pipeline statistically outperforms the GNSS pipeline across all four score metrics.

5 Conclusion and further research

In this study, we designed and evaluated an automated framework to classify pre-defined human movement patterns collected with GNSS and UWB technology. The automated framework takes raw spatio-temporal data from both the GNSS and UWB systems, pre-processes this raw data to compute segments as well as global and local features, and at the end classifies the data into the pre-defined trajectory movement patterns. The automated framework allows for splitting the original trajectory into segments and applies three options for noise removal: (i) no noise removal, (ii) Savitzky Golay filter on the raw location data or (iii) Savitzky Golay filter on the extracted features, and three options for classification algorithms: (i) DT, (ii) RF or (iii) SVM. Additionally, we collected a total of 104 5-minute-long discrete GNSS and similar discrete UWB trajectories consisting of 4 unique movement patterns. Based on these collected discrete trajectories, we tested nine different pipeline configurations of our automated framework via a bootstrapping protocol to assess which one yielded the best performance. The hyperparameter optimalisation was automated by the SMAC method.

For the GNSS technology, we achieved the highest performance with a pipeline consisting of noise removal applied on the raw location data in combination with an RF model. Regarding the UWB technology, the pipeline with no noise removal in combination with an SVM model gave the highest performance for the F_1 and MCC score metrics. We compared these two pipeline configurations and concluded, based on a significance level of 99% , that UWB achieved significantly better than GPS in classifying the pre-defined movement patterns in areas of approximately 100 square meters.

The movement patterns were limited to only four basic pre-defined movement patterns in our experiments. In future work, we could include more data with various complex movement patterns to make the pipeline configuration more applicable to real-life complex human movement behaviour. Thereby, a larger amount of trajectory data collections could enable future studies to investigate the use of more advanced classification models based on Recurrent Neural Networks (RNN). Local trajectory shape recognition could also be the solution to identify more complex trajectories. Here, sequence labelling can play a role to learn algorithms capable of classifying sequential patterns such as LSTM's, to recognize typical behaviour of humans based on long-lasting complex trajectories. Furthermore, in real-world scenarios, movement pattern data might not always include the ground truth. Thus, further research might explore the use of unsupervised classification algorithms [40], instead of supervised algorithms, to cluster trajectories in various patterns. Finally, the usage of additional sensors such as an accelerometer and a gyroscope might improve the performance of GNSS and UWB systems.

References

1. Anderson, T.W., Darling, D.A.: Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes. The annals of mathematical statistics pp. 193–212 (1952)
2. Baratchi, M., Meratnia, N., Havinga, P.J.: Recognition of periodic behavioral patterns from streaming mobility data. In: International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services. pp. 102–115. Springer (2013)
3. Barrios, C., Motai, Y.: Improving estimation of vehicle's trajectory using the latest global positioning system with kalman filtering. IEEE Transactions on Instrumentation and Measurement **60**(12), 3747–3755 (2011)
4. Bastida Castillo, A., Gómez Carmona, C.D., De la Cruz Sánchez, E., Pino Ortega, J.: Accuracy, intra-and inter-unit reliability, and comparison between gps and uwb-based position-tracking systems used for time–motion analyses in soccer. European journal of sport science **18**(4), 450–457 (2018)
5. Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics **21**(1), 1–13 (2020)
6. Dabiri, S., Heaslip, K.: Inferring transportation modes from gps trajectories using a convolutional neural network. Transportation research part C: emerging technologies **86**, 360–371 (2018)
7. Dabove, P., Di Pietra, V.: Towards high accuracy gnss real-time positioning with smartphones. Advances in Space Research **63**(1), 94–102 (2019)
8. Etemad, M., Soares Júnior, A., Matwin, S.: Predicting transportation modes of gps trajectories using feature engineering and noise removal. In: Canadian conference on artificial intelligence. pp. 259–264. Springer (2018)
9. Feurer, M., Hutter, F.: Hyperparameter optimization. pp. 3–38. Springer (2019)
10. Gorodkin, J.: Comparing two k-category assignments by a k-category correlation coefficient. Computational biology and chemistry **28**(5-6), 367–374 (2004)
11. Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756 (2020)
12. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
13. Kalman, R.E.: A new approach to linear filtering and prediction problems (1960)
14. Kaplan, E.D., Hegarty, C.: Understanding GPS/GNSS: principles and applications. Artech house (2017)
15. Kennedy, H.L.: Improving the frequency response of savitzky-golay filters via colored-noise models. Digital Signal Processing **102**, 102743 (2020)

16. Kuhn, M., Zhang, C., Merkl, B., Yang, D., Wang, Y., Mahfouz, M., Fathy, A.: High accuracy uwb localization in dense indoor environments. In: 2008 IEEE International Conference on Ultra-Wideband. vol. 2, pp. 129–132. IEEE (2008)
17. Laanen, R.: Classification of pre-defined movement patterns: A comparison between gnss and uwb technology. LIACS (2022)
18. Lin, Q., Liu, X., Wang, W.: Gps trajectories based personalized safe geofence for elders with dementia. In: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 505–514. IEEE (2018)
19. Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F.: Smac3: A versatile bayesian optimization package for hyperparameter optimization. In: ArXiv: 2109.09831 (2021), https://arxiv.org/abs/2109.09831
20. Luo, Y., Irvin, D., Buss, N., Gutierrez, A., Rous, B.: Outdoor playground localization system for tracking young children using ubisense sensor network. In: 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). pp. 7–12. IEEE (2020)
21. Mahfouz, M.R., Zhang, C., Merkl, B.C., Kuhn, M.J., Fathy, A.E.: Investigation of high-accuracy indoor 3-d positioning using uwb technology. IEEE Transactions on Microwave Theory and Techniques **56**(6), 1316–1330 (2008)
22. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. The annals of mathematical statistics pp. 50–60 (1947)
23. Marins, F., Rodrigues, R., Portela, F., Santos, M., Abelha, A., Machado, J.: Extending a patient monitoring system with identification and localisation. In: 2013 IEEE International Conference on Industrial Engineering and Engineering Management. pp. 1082–1086. IEEE (2013)
24. Mimoune, K.M., Ahriz, I., Guillory, J.: Evaluation and improvement of localization algorithms based on uwb pozyx system. In: 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). pp. 1–5. IEEE (2019)
25. Nasri, M., Tsou, Y.T., Koutamanis, A., Baratchi, M., Giest, S., Reidsma, D., Rieffe, C.: A novel data-driven approach to examine children's movements and social behaviour in schoolyard environments. Children **9**(8), 1177 (2022)
26. 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., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
27. Petre, A.C., Chilipirea, C., Baratchi, M., Dobre, C., van Steen, M.: Chapter 14 - wifi tracking of pedestrian behavior. In: Xhafa, F., Leu, F.Y., Hung, L.L. (eds.) Smart Sensors Networks, pp. 309–337. Intelligent Data-Centric Systems, Academic Press (2017). https://doi.org/https://doi.org/10.1016/B978-0-12-809859-2.00018-8, https://www.sciencedirect.com/science/article/pii/B9780128098592000188
28. Pozyx: Pozyx. https://www.pozyx.io (2022)
29. Pozyx: Pozyx creator kit. https://www.pozyx.io/creator (2022)
30. Samsung: Samsung galaxy tab a7 lite. https://www.samsung.com/nl/tablets/galaxy-tab-a/galaxy-tab-a7-lite-lte-gray-32gb-sm-t225nzaaeub/ (2022)

31. Savitzky, A., Golay, M.J.: Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry **36**(8), 1627–1639 (1964)
32. Schafer, R.W.: What is a savitzky-golay filter?[lecture notes]. IEEE Signal processing magazine **28**(4), 111–117 (2011)
33. Shih, D.H., Shih, M.H., Yen, D.C., Hsu, J.H.: Personal mobility pattern mining and anomaly detection in the gps era. Cartography and Geographic Information Science **43**(1), 55–67 (2016)
34. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information processing & management **45**(4), 427–437 (2009)
35. Tomaštík Jr, J., Tomaštík Sr, J., Saloň, Š., Piroh, R.: Horizontal accuracy and applicability of smartphone gnss positioning in forests. Forestry: An International Journal of Forest Research **90**(2), 187–198 (2017)
36. Wang, C., Bäck, T., Hoos, H.H., Baratchi, M., Limmer, S., Olhofer, M.: Automated machine learning for short-term electric load forecasting. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 314–321. IEEE (2019)
37. Wang, C., Baratchi, M., Bäck, T., Hoos, H.H., Limmer, S., Markus, O.: Towards time-series-specific feature engineering in automated machine learning frameworks for time-series forecasting. IEEE (under review)
38. Waqar, A., Ahmad, I., Habibi, D., Phung, Q.V.: Analysis of gps and uwb positioning system for athlete tracking. Measurement: Sensors **14**, 100036 (2021)
39. Wilcoxon, F.: Individual comparisons by ranking methods. In: Breakthroughs in statistics, pp. 196–202. Springer (1992)
40. Yao, D., Zhang, C., Zhu, Z., Hu, Q., Wang, Z., Huang, J., Bi, J.: Learning deep representation for trajectory clustering. Expert Systems **35**(2), e12252 (2018)