WanDa: A Mobile Application to Prevent Wandering

Berardina De Carolis[1], Vincenzo Gattulli[1], Donato Impedovo[1] and Giuseppe Pirlo[1]

[1] Department of Computer Science, University of Bari Aldo Moro (Bari)

Abstract. The nature of Big Data tends to collect a huge quantity of useful information about human life. Implementing Artificial Life applications inherent to health could improve and sensitize individuals to the future. In fact, would be useful to implement an application that monitors people with neurodegenerative diseases when they are away from home to monitor Wandering. In this paper, an application called "WanDa" is proposed that monitors and prevents deviations from the usual path in real-time and, if wandering is detected, can guide the elderly person to a safe place and alert caregivers or relatives. The application uses the sensors and technologies of a generic Android smartphone and has a very simple interface to manage wandering behaviors. We tested the application from two perspectives: the accuracy of the algorithm in detecting wandering behaviors and the user experience. In both cases, WanDa was judged positively (Questionnaires performed by caregivers of patients who take the test), showing that it can be a useful support for managing, monitoring, and reporting wonder episodes.

Keywords: Health and Social Care, Smartphone, Wandering, Android Application, Automated Wandering Behavior, Behavioral Biometrics.

1 Introduction

Artificial life is in overall evolution. In this area, it is possible to implement applications that succeed in improving the human standard of living. Behavioral Biometrics tends to fruit big data and patterns to be able to identify a specific behavior. Indeed, in this paper, we would like to monitor wandering. Wandering is often identified as a symptom of Alzheimer's disease or dementia. It is a widespread phenomenon in the elderly with dementia and occurs when a person roams around and becomes lost or confused about his location. People with dementia often wander because they are stressed, looking for something, engaging in past routines, or, other times, they may wander without aim at all. On the other hand, the sufferer tends to forget where they are going and what they were trying to do, wandering around and losing track of where they are. Wandering is also defined as "a dementia-related locomotor behavior syndrome having a frequent, repetitive, time-disordered and spatially disoriented nature that manifests in patterns of Lapping (more on that later), Random and Pacing" [1]. Wandering is also significantly common also among children with *ASD (Autism Spectrum Disorder)* and those with behavioral and developmental problems than among other children. Wandering among children with ASD tends to increase with increasing severity of ASD symptoms and decreasing developmental level and it is higher among males and children who have attention deficit hyperactivity disorder, anxiety, depression, or oppositional behavior [2]. Even wholly healthy, unimpaired people/children can find themselves at a loss, perhaps in times of high stress, which, induced by everyday life, can lead to a state of confusion at any time while they are headed toward a destination. Traditional methods impose physical restrictions to prevent wandering. However, physical, and psychological problems caused by physical restraints make traditional solutions impractical or ineffective in protecting vagrants. As an alternative, preventive measures have been recommended to maximize autonomy by minimizing the risk to older persons prone to wandering in home, community, and nursing facility settings. Wandering is a phenomenon that needs to be controlled because prohibiting patients from wandering could mean the loss of benefits associated with walking, including improved circulation and oxygenation and decreased risk of contractures. Interventions for wandering have prevented the behavior through physical and pharmacological restrictions. Apart from the known harmful effects of restraints, such as bedsores, anxiety, physical violence, falls, and high morbidity and mortality rates, the intervention is ineffective. Non-pharmacological interventions are a safer option and include electronic tagging and tracking devices, behavioral approaches, exercise, music therapy, aromatherapy, access camouflage, and environmental modifications [3]. This work aims at addressing one of the significant concerns of caregivers, which is to prevent and detect wandering episodes of individuals for whom they are in care. Thanks to smartphone technologies, it is possible to ease and support caregivers by assisting and monitoring the paths of assisted persons. In fact, in this work, we propose a solution offered through an Android application called WanDa which, thanks to a sought-after compromise between the efficiency of the techniques used and the accuracy of the results, succeeds in helping to manage the phenomenon of wandering. The application consists of three complementary software modules, thanks to the GPS hardware present in any Android smartphone, the app detects situations of wandering to present, when appropriate, quick solutions to the users to help them manage the situation of possible panic. The paper is structured as follows: the second section illustrates the state of the art of algorithms used by applications created for this purpose. Section three illustrates the design and implementation of the WanDa application. Then, in Section four tests in real-world contexts with experimental results will be illustrated. Finally, Section five illustrates the conclusions and planned future work.

2 Related Work

The design of a system depends substantially on the target scenario. As discussed by E. Batista et al. [4], there are two scenarios with a wide range regarding the location of roaming behavior: outdoor and indoor. The objective of this study is to provide a solution-oriented to the first category. To develop a solution that can prevent elderly people with dementia from transgressing the boundaries, which is often related to getting lost or other adverse events without timely care services, Q. Lin et al.[5] propose a Data Mining-based approach to build a personalized and secure outdoor geofence by extracting individuals' historical GPS trajectories, that is a kind of virtual boundary delineated around an area of interest that can be created with a variety of different techniques, such as Wi-Fi, cellular network, RFID and GPS. To build a secure external geofence customized for each elderly person, the method of geofencing is applied. It is proposed to model the movement of each elderly person as a graph, whose vertices refer to frequently visited places and the edges of this graph are paths between two adjacent locations. The qualitative evaluation results showed that the method is feasible for constructing customized secure geofences based on individuals' historical GPS trajectories. This secure geofence can be used to detect online whether an elderly person with dementia has moved across the boundary of the secure geofence [5]. Very similar work has been done by D. Zhang et al. [6] to provide appropriate real-time care services to elderly people who suffer from physical or cognitive disabilities and who often have difficulty navigating activities and remembering landmarks. They propose a disorientation detection method that detects outliers within GPS trajectories. Precisely, the first model of an individual's movement trajectories as a graph based on his historical GPS tracks by developing a method called iBDD that can detect two categories of peripheral trajectories in a uniform framework in real-time. To perform this detection, they project the historical movement trajectories of each elderly person onto a digital map and model the regular movement of each elderly person as a graph in which the vertices represent the places attended and the edges correspond to the paths between the places. Using the real-world GPS datasets of ten individuals, they demonstrated that iBDD could achieve a 95% disorientation detection rate [6]. Y. Chang [7] proposed real-time anomaly detection by considering user trajectories as input, and the anomaly is identified in output. Trajectories are modeled as a discrete-time series of axis-parallel edges ("*boxes*") in two-dimensional space. A trajectory is represented by a series of boxes, each with six attributes: The maximum longitude value, The minimum longitude value, The maximum latitude value, The minimum latitude value. A box is started with an acquired GPS point. When two boxes overlap, their similarity is equal to 1. The farther apart the two boxes are, the lower their similarity is. For 2D trajectories, the similarities can be decomposed into longitude and latitude similarity. Experimental results show that accuracy is 97.6 percent (*with Axis Adjustment*), and recall is 98.8 percent for participants with cognitive impairment [7]. The IRoute system invented by S. Hossain et al. [8] implements the Belief-Desire-Intention (BDI) architecture. It tracks the GPS position of the person with dementia in real-time and updates predictions based on changes in position. A deviation from the predicted path is considered abnormal behavior. As an intervention technique, a correct route is provided to induce the person with dementia to follow it. Failure to follow the guided route triggers the system to notify caregivers. Running on a GPS-equipped cell phone, the BDI agent is responsible for route prediction using user input (*list of travel locations and activities, frequency of events, start times, and destination locations*) and stored routes from previous trips (*a set of timestamps and GPS locations to a destination*) [8]. After showing the state of the art, some techniques chosen for the implementation of the WanDa application will be explained in more detail, namely the wandering detection algorithm called θ_WD and the *C-SIM* similarity measure. The θ_WD algorithm proposed by Lin et al. [9] is a wandering trajectory classification algorithm designed for the outdoor environment. They configured the algorithm by considering the angle between two trajectories considering lapping and pacing patterns as indicators. The authors [9] define the algorithm for wandering detection as "a ring trip, with each ring consisting of a series of track segments, blocked by two adjacent sharp points within a given distance interval." In summary, the algorithm has been implemented such that one loop, with four segments divided by sharp points, is sufficient to detect a wandering trace. According to the definition, repetitive loops are required to label a trace as stray. The algorithm attempts to identify only whether a trajectory contains loops and uses patterns to obtain a more generalized solution. The authors assumed that, in a wandering trajectory, there are points (those referred to as *"sharp points"*) with significant changes in direction, equal to vector angles equal to or greater than 90 degrees. Continuous acute points create at least one loop in the wandering trajectory. The angle between consecutive vectors (thus from three consecutive points) is an indicator of the direction of motion. If the angle between them is greater than 90° and this is repeated 4 times within the sub-segments (whose length is prefixed/parameter), then the algorithm returns the positive presence of a wandering situation [9]. Regarding the similarity between paths, the method illustrated in the article *R. Mariescu-Istodor* et al. [10] that illustrates its simplicity of implementation and at the same time, provides a fast algorithm with complexity in linear time. When overlapping two paths, the real path and the ideal path, the following measure is very useful. The amount of overlap measures how similar the paths are. Specifically, Cell Similarity (C-SIM) is a similarity measure that considers the area traversed by the two paths. It uses a grid to calculate a cell representation for the two trajectories and then measures how many cells are in common relative to the total number of distinct cells. In theory, an infinitely long path can exist in a single cell simply by moving in a circle within the cell. This type of behavior is sometimes noticed when the user is stationary, but the GPS signal fluctuates. Often points close to each other end up in different cells due to arbitrary grid division. This can produce errors when comparing routes. In principle, two people walking hand in hand can never share a cell. To resolve this and compensate for the arbitrary division of a grid, which can then allow points as far apart as 1 mm to lie in different cells, CSIM uses morphological dilation with a square (3 x 3) structural element. When using these grid-based approaches, the size (parameter L) of the cell acts as a distance threshold, and points mapped to the same cell are considered identical. The CSIM formula is essentially Jaccard's coefficient modified to handle dilated cells [11]:

$$S(C_A, C_B) = \frac{|(C_A \cap C_B) \cup (C_A \cap C_B^d) \cup (C_B \cap C_A^d)|}{|(C_A \cup C_B|} \quad (1)$$

From the point of view of effectiveness, C-SIM is the least affected by sampling rate variations and performs quite well under conditions of noise and point shift [10]. The θ_WD algorithm proposed by Lin et al. [9] will be used within one of the modules for wandering detection in the WanDa application. More precisely, it has been adapted in the component defined as "knowledgeless," which provides as input to the algorithm the GPS points of the user's walk that the smartphone acquires in real-time. While with regard to the path similarity measurement, the C-SIM algorithm [10] (*https://github.com/uef-machine-learning/C-SIM*) written in Java was chosen and after modifying it by implementing the sliding window method, it was included within the module called "with knowledge," precisely because it can detect wandering occasions thanks to the knowledge of the paths, the similarity of which with respect to the user's recent walk is carried out precisely thanks to the just-discussed C-SIM algorithm.

3 WanDa Mobile Application

The WanDa application is divided into two parts: the first intended for users and the second one intended for caregivers or parents in the case of child users. Users are provided with a simple interface consisting only of the central button for starting the Wandering service in the background, as well as related buttons for interacting with the map (focus and zoom in on the current location, zoom map, north up). Also designed for users is the alarm screen that appears on full screen whenever the system detects lost possibilities with appropriate buttons that can help them manage the situation. Also thought for users is the mode that makes Google Maps start navigation to one of the destinations previously stored in the settings and finally also the call to a preset phone number (relative, caregiver, rescue...). On the other hand, for the part of the application dedicated to caregivers, they can input settings (accessed from the button at the top right of the main screen), which will allow them to add information such as emergency numbers, user's usual routes, usual places, residence, and so on. Moreover, caregivers can use the app for localizing the assisted user in case of need (help request, wondering detection). The screens of the WanDa app will be illustrated below (Fig. 1).

Fig. 1. a) Main screen, b) Usual routes entry screen, c) Settings, d) Report screen, e) Screen for entering usual places, f) Alarm (six),

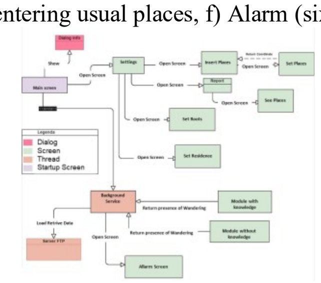

Fig. 2. WanDa Application Class Diagram

The (Fig. 2) shows the diagram of the classes, that is, how the classes used for detection interact with each other.

3.1 Main screen

The main screen contains simple graphical buttons with large print. The main screen includes a large central button that allows you to start the wandering detection service, a settings button, and an info button that disappears once the service is started (*This was done first and foremost so that the user cannot absentmindedly press it while walking*). On the first startup, you will be asked for geolocation permissions. After clicking the middle button, the Google map will be launched (Google Maps API with some simple features such as Zoom and map rotation) where some routes entered by the caregiver (*through the Gson class, which is an open-source Java library for serializing and* *deserializing Java objects to JSON*) and stored locally (*3.3 Usual routes entry screen*) will be present in memory. Concerning the bottom section, there will be two buttons: "SOS" is used to initiate a phone call to the number set for emergencies; "routes icon" which allows to open another screen that contains the list of saved places and if an item is clicked on it, it makes the Google Maps app open with already set up walking navigation via to the relevant coordinates that were previously saved in the places entry section.

3.2 Settings screen

The settings screen is accessible only when the app is not in the Wandering Detection state; it is divided into several sections. The first section allows for the entry of both an emergency phone number and the maximum minute by which you want the user to return to the home residence. The second section relates to habitual "*Places and Routes,*" the first item is a link that leads to another screen used for entering the user's habitual routes, and the second item is a link that leads to a screen where the name and coordinates of places can be recorded, and the third item provides the option of setting the residence. The third section contains a link that leads to the user's report.

3.3 Usual routes entry screen

The screen for entering routes features a Google map in the center and buttons about the features implemented for entering the user's usual routes. First, it is a + button to be able to add a new route, the starting point of which can be entered by clicking on the desired map point. In addition, there is a "Cancel" button that allows the user to delete the entire set of various points entered. These, once saved by clicking "Save Route," will be automatically overlaid on the map present on the main screen. This step is used for caregivers.

3.4 Alarm screens

The alarm screens appear one consequent to the other. When the system detects a wandering state, a screen appears with two buttons. The first one is used to say that "*you are actually lost*" and immediately opens the management screen of the dangerous situation, while the one further down, implying that it is a false positive wandering situation, closes the alarm screen. The action of the first button will open another screen with three buttons. The first one immediately initiates a call to the preset emergency telephone number. The second one refers to the screen containing the usual locations to initiate the relevant navigations. Finally, the third one at the bottom is used to close the screen. When launched, the smartphone's vibration is activated for four seconds, as well as playing a ringing sound to call the user's attention. The management of feedback given to the user in case of wandering has been studied by experts in the field. In fact, the caregiver must make sure that the sound is bailed on the device and that it is loud. In addition to the loud sound emitted, vibration of the device has also been added and in addition a design with large, colorful words to alert caregivers or the patient himself if he is still conscious.

3.5 Screen for entering usual places

This screen consists of a field to enter the name you want to give to the usual place (*to perhaps help better remind the user which place it is*), a button that redirects to another where you center the place with the cursor placed in the middle. Finally, a button to store the place with the information entered just described. This, once clicked, in addition to storing the various information locally, causes the name of the place just added to appear in a list placed at the bottom. If the relevant item in the list is clicked on, it is deleted both from it and from the local memory of the smartphone.

3.6 Report screen

The report screen, accessible through the settings section, contains a list of all reports related to walks started by the user, such that it has as headers the date and time related to the time when that tracking session was started. After clicking on an item, a screen will open regarding the session and contain information. More specifically, it will be indicated, with date and time, when the user departed from the usual routes and whether he returned, when he made abnormal movements, and when he was around the residence. Finally, if you want to delete a report, hold down on the one you want to delete from the list.

3.7 Starting wandering detection

When you click on the center green button on the main screen, a popup appears asking the user to enable it if GPS is not active. If yes, the application switches to the active detection state. Mobile devices have several techniques for keeping tabs on their geographical location; in addition to GPS, they can take advantage of some information about the Wi-Fi network to which they are connected. However, if you have no special requirements, the easiest way to monitor the user's location is to use the LocationManager service. One of the service tasks that will run in the background (dedicated to wandering detection) is to communicate with this service, activating it during initialization and listening for its updates. For wandering detection, two modules have been designed that work consequently with each other. The first is called the "Module with knowledge," which manages to detect wandering possibilities by already knowing the routes that the user usually takes (Entered previously by the caregiver); the second is called the "Module without knowledge," that is, the one that works without the knowledge of the usual routes that the user takes. The "Module with knowledge" first calculates the length of the recent route taken by the user, which corresponds to the set of GPS points acquired from the smartphone that is stored (*always and only the last x points, where x is a predefined parameter*). Then, each route stored in the habitual will be overlaid with the user's route in real-time (*3.8 Sliding Window Algorithm*). Suppose none of the maximum similarities obtained from this comparison is greater than or equal to the value set as a threshold (decided after several tests). In that case, a timer of 1 minute (the time the user must re-enter the routes entered) is started. If one of the similarities has a value greater than the set threshold during the expiration of the time, then the timer will be canceled, asserting that the user has re-entered the set route. The alarm screen will appear if the minute is exceeded, and the user has not re-entered the preset routes. After this first step is carried out by the "Module with knowledge," further verification will be carried out by the "Module without knowledge." It can detect, based on the "*morphology*" of the user's route, possible wandering. The system would report wandering if the angles formed by the GPS positions include circular-type trajectories or anomalies within the path. The detection of these unusual movements is left to a special algorithm that, if successful, triggers the wandering alarm with the consequent appearance of the alarm screen. This algorithm is referred to as *θ_WD*. In addition, the calculation of the scalar product and the norm useful for calculating acute angles has also been included, which, if greater than three, cause the function to return a positive outcome. Finally, a third module that works by simply setting a timer whose duration is set from the application settings has also been implemented. The timer starts as soon as the user initiates Wandering detection from the button on the main screen. The alarm, in this case, is triggered only if the user does not return to the residence within the time set by the caregiver. The WanDa detector is activated by the caregivers with the purpose of monitoring from the starting point the entire pathway the patient performs. The WanDa detector will always be turned off by the caregivers.

3.8 Sliding Window Algorithm

By testing the C-SIM algorithm first among samples of sets of GPS points and then among data received by the application, it was found that the algorithm, in returning the percentage of equality between two routes, considers their length of them. However, the latter feature made the similarity untrue since the length of the user's last steps was often somewhat shorter than the length of the various stored routes. Therefore, the C-SIM similarity algorithm was adapted to the "sliding window" approach. This means that the algorithm compares the user's last steps and all the various possible fragments for each of the paths. In addition, as a matter of efficiency, this will be done only with those that are longer, or slightly shorter, than the length of the walk. Each comparison made with each portion then returns an index concerning its similarity and the last steps taken. Finally, the highest similarity that was obtained is retained for each path. All this is then done to proportionally scale the comparison between the various routes concerning the distance of the walk taken by the user. Otherwise, an incorrect similarity would be obtained due precisely to the different lengths of both in terms of meters.

4 Testing and Results

The test of the application was divided into five parts. The first three experiments were performed by domain experts (*Heuristic Tests*) with the purpose of testing the modules with knowledge and without through the simulator offered by Android Studio, while the last two tests were performed with real users. The tests are named *"Heuristic Tests on the modules with and without knowledge", "Tests with different parameters", "Tests on two GPS track datasets ", "Real-world Testing" and "Questionnaire results"* with SUS, UEQ and NPS Questionnaire to evaluating WanDa usability. Heuristic tests were conducted for the first three tests (4.1, 4.2, 4.3), they were carried out by a group of system experts that do not have any wandering pathology. The tests 4.4 were performed in real contexts by 12 patients that have wandering pathology. The 4.5 test was performed by caregivers of patients. The first three experiments were carried out by experts, they are analyzing that do not consider saved datasets

4.1 Heuristic tests on forms with and without knowledge

These tests, for both modules were successful, it was noticed both numerically and visually that the evaluator was proceeding in the route showing green dots in real-time. In Fig. 2 (Module with knowledge - left) we can see that the user followed the route precisely and starts to deviate from it by turning left for a few meters. In the upper left corner, the similarity of 1 is shown, a value that will remain for a few more steps if he continues that route. In Fig. 2 (Form with knowledge - right), on the other hand, the user has deviated quite a bit from the previously entered route (colored yellow), and in fact the similarity is very close to the default 0.6 set threshold.

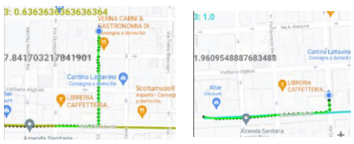

Fig. 2. Example Module with knowledge

Fig. 3. Example Module without knowledge

In Fig. 3 (Module without knowledge - left) since the repetition of the path occurred relative to only the first 50 meters, the second module gave wandering alerts given the sudden change of course that occurred several times. In Fig. 3 (Module without knowledge - right) there is a circle-shaped path forming some rather sharp angles that triggered the wandering alert.

4.2 Testing with different parameters

The evaluators experimentation will set parameters that will be evaluated in the actual tests. The first parameter concerns the number of GPS positions (steps). Several tests were carried out with 30, 40, 50 and 60 steps. The results shown in Table 1:

Table 1. Number of tracks correctly recognized per parameter GPS positions to be kept

	Parameter Values			
Recognized traces	30	40	50	60
	8/10	9/10	8/10	7/10

The value 40, through experimental traces fed into the Android emulator, is the value that gave the best results, and this is due to the fact that if the number of steps is too short, especially below 30 steps. With too high a value would have the opposite effect of taking too many past points making the similarity untrue, in addition to the fact that it would make the calculation operated by the various modules more onerous. The second parameter relates to the size of the grid cells used by the "*knowledgeless module*." Detecting this parameter experimentally was difficult since it will depend on it how far the user can stray from the predetermined routes.

Table 2. Number of traces recognized correctly regarding the width of grid cells

	Parameter Values			
Recognized traces	20	30	35	40
	1/10	7/10	9/10	8/10

The value of 37 meters is ideal, since in practice it would correspond to allowing parallel routes even up to almost 80 meters away. Parallel routes are still considered viable and accepted as like the route set by the caregiver. As can be seen from the table above, even values close to 37 (30 and 40). In the case of small villages, this metric corresponds to one or even two parallel routes (albeit close to each other) that still allow some freedom of decision in choosing an alternative route but at the same time prevent going very far. Finally, regarding the similarity threshold value, during the various tests it was set equal to 0.6 given parameter 37 for the width of each cell.

4.3 Testing on two GPS track datasets

In this subchapter, paths were simulated thanks to the emulation of GPS locations in Android Studio with the purpose of verifying the reactions and outputs of the application. The purpose of this test is to ascertain the behavior of the application with complete paths and in different situations. Two datasets will be extracted for this test:

- *Dataset1:* 10 GPS routes where there is no presence of wandering;
- *Dataset2:* 10 GPS routes with wandering situations.

Table 3 shows the number and relative percentage of tracks recognized correctly by the app. An accuracy of 85% was obtained with this test:

Table 3. Trace set confusion matrix

	Dataset1	Dataset2
Recognized	8	9
Not Recognized	2	1

Regarding the first dataset there were two traces that were not recognized exactly. This happened because at the moment when the user was re-entering the path (module with knowledge), the similarity did not go back up correctly (which it would have safely done a few steps further, however) causing the one-minute timer to start and expire. Regarding the second dataset, it was the second module that sensed the wandering at the moment when the user slowed down right under a balcony that also made the points less accurate thus forming "*sharp corners*." The datasets referencing this experiment are not public. They were created by a group of experts in the evaluation of the heuristic test (the same experiments considered for experiments 4.1 and 4.2). This group of people can test the application before using the patients in the real world experiment (4.4).

4.4 Real-world testing

Single test: In this subchapter, the WanDa application was tested with 12 real users in real contexts. The 12 users were made to perform tasks such as, for example, following certain set routes or parallel routes (*Task without wandering*), and tasks such as getting lost on the way or making abnormal movements such as turning around on a point or stopping for a long time (*Task with wandering*) with the purpose of testing the application. The results obtained from this test have been summarized in Table 4. An accuracy of 96% was obtained from this real-world test.

Table 4. Confusion matrix on real-world testing

	Task without wandering	Task with wandering
Recognized	11	12
Not Recognized	1	0

As can be seen from Table 4, the real-world test found very good results, confirming the validity of the parameters set in the previous tests (*number of steps to keep in memory, cell width, and threshold value (4.2 Tests with different parameters)).* The only unrecognized path is inherent in the first module, which incorrectly recognizes a path without wandering as wandering. Analyzing the data on the server remotely, a slight shift caused the similarity to decrease, which triggered the alarm screen. *Group* *test*: This test was carried out which consists of performing the same tests as the single test but moving as a group (*like a group of students to school*). The purpose of this experiment is to test whether the distance between users, a few meters apart, could affect the detection of wandering. The results highlighted how, due in part to the accuracy of the various smartphones, that the points detected when stopping, especially in areas with interference, were 3 or 4 meters apart between the various smartphones of the group members. However, this equally allowed wandering to be detected only when appropriate, achieving 96% accuracy concerning the first module. On the other hand, when it comes to the second module, depending on the angles in the walk, wandering was still detected by the various smartphones, although on a few occasions there happened to be a few tens of seconds delay in detection from one device to another.

4.5 Questionnaire results

The SUS, UEQ, and NPS questionnaires were administered to each user to measure the users' subjective impression of using the app and to understand how effective the GUI was. Regarding the *UEQ (User Experience Questionnaire)* the app presented some problems related to perspicuity for some users. But for most of them, on the other hand, it was found to be efficient (it responds abruptly to detection). Finally, it was found to have high originality about how it works. Next, the *SUS (System Usability Scale)* questionnaire was submitted, finding an overall score of 73. Looking at the rankings of the SUS questionnaire scores, the application ranks on grade B, thus with the relative adjective of "Good," presenting a usable interface. Finally, the *NPS (Net Promoter Score)* questionnaire was also submitted, and by calculating the score based on the number of detractors, passives, and promoters, it turns out that the score is 75. Regarding the questions related to originality and innovation, it was inferred that the application was also found to be very original and proposed a solution that many users had not seen or otherwise heard about before.

5 Conclusions and Future Work

The study enabled the implementation and testing of the WanDa application that was able to detect the presence of wandering by individuals with neurovegetative diseases. The application, after being carefully set up and adjusted by the caregiver, provides high accuracy in detecting wandering, and this can help and support both actors, with all the benefits that come with it. The solution, observing the various tests and questionnaires, presents some limitations. In fact, as discussed in the experimentation chapter, some situations were found to be mistakenly recognized as pertaining to wandering episodes. In addition, through SUS, NPS, and UEQ questionnaires administered to users, it was found that the application interface could be improved for some aspects. This, no doubt, indicates that the application could be further improved in the future not only from the UX point of view but also as far as functionalities are concerned. To this aim, we will conduct a study for a longer period in cooperation with a local association that assists people with dementia. From its results we will refine the application to better meet the needs of final users. Moreover, we will investigate also how the application could take advantage of using smart bands and smartwatches. The WanDa application is not yet available, some parts of it will be improved and later other modules will be added such as the fall detection module.

Acknowledgments

This work is supported by the Italian Ministry of Education, University and Research within the PRIN2017 - BullyBuster project - A framework for bullying and cyberbullying action detection by computer vision and artificial intelligence methods and algorithms.

References

1. Lin, Q., Zhang, D., Chen, L., Ni, H., Zhou, X.: Managing elders' wandering behavior using sensors-based solutions: A survey. Int J Gerontol. 8, 49–55 (2014). https://doi.org/10.1016/J.IJGE.2013.08.007.
2. Wiggins, L.D., DiGuiseppi, C., Schieve, L., Moody, E., Soke, G., Giarelli, E., Levy, S.: Wandering Among Preschool Children with and Without Autism Spectrum Disorder. J Dev Behav Pediatr. 41, 251–257 (2020). https://doi.org/10.1097/DBP.0000000000000780.
3. Adekoya, A.A., Guse, L.: Wandering behavior from the perspectives of older adults with mild to moderate dementia in long-term care. Res Gerontol Nurs. 12, 239–247 (2019). https://doi.org/10.3928/19404921-20190522-01.
4. Batista, E., Casino, F., Solanas, A.: On wandering detection methods in context-aware scenarios. IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications. (2016). https://doi.org/10.1109/IISA.2016.7785349.
5. Qiang, L., Xinshuai, L., Weilan, W.: GPS Trajectories Based Personalized Safe Geofence for Elders with Dementia. 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). 505–514 (2018). https://doi.org/10.1109/SmartWorld.2018.00111.
6. Lin, Q., Zhang, D., Connelly, K., Ni, H., Yu, Z., Zhou, X.: Disorientation detection by mining GPS trajectories for cognitively-impaired elders. Pervasive Mob Comput. 19, 71–85 (2015). https://doi.org/10.1016/J.PMCJ.2014.01.003.
7. ChangYao-Jen: Anomaly detection for travelling individuals with cognitive impairments. ACM SIGACCESS Accessibility and Computing. 25–32 (2010). https://doi.org/10.1145/1873532.1873535.
8. Hossain, S., Hallenborg, K., Demazeau, Y.: iRoute: Cognitive Support for Independent Living Using BDI Agent Deliberation. Advances in Intelligent and Soft Computing. 90, 41–50 (2011). https://doi.org/10.1007/978-3-642-19931-8_6.
9. Lin, Q., Zhang, D., Huang, X., Ni, H., Zhou, X.: Detecting wandering behavior based on GPS traces for elders with dementia. 2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012. 672–677 (2012). https://doi.org/10.1109/ICARCV.2012.6485238.
10. Mariescu-Istodor, R., Fränti, P.: Grid-based method for GPS route analysis for retrieval. ACM Transactions on Spatial Algorithms and Systems. 3, (2017). https://doi.org/10.1145/3125634.
11. Kumar, A., Lau, C.T., Chan, S., Ma, M., Kearns, W.D.: A unified grid-based wandering pattern detection algorithm. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. 2016-October, 5401–5404 (2016). https://doi.org/10.1109/EMBC.2016.7591948.