Health Informatics Meets eHealth
G. Schreier et al. (Eds.)

doi:10.3233/978-1-61499-645-3-259

A Smartwatch-Based Assistance System for the Elderly Performing Fall Detection, Unusual Inactivity Recognition and Medication Reminding

Markus DEUTSCH[a] and Harald BURGSTEINER[a,1]
[a]*Institute for eHealth, Graz University of Applied Sciences, Graz, Austria*

Abstract. The growing number of elderly people in our society makes it increasingly important to help them live an independent and self-determined life up until a high age. A smartwatch-based assistance system should be implemented that is capable of automatically detecting emergencies and helping elderly people to adhere to their medical therapy. Using the acceleration data of a widely available smartwatch, we implemented fall detection and inactivity recognition based on a smartphone connected via Bluetooth. The resulting system is capable of performing fall detection, inactivity recognition, issuing medication reminders and alerting relatives upon manual activation. Though some challenges, like the dependence on a smartphone remain, the resulting system is a promising approach to help elderly people as well as their relatives to live independently and with a feeling of safety.

Keywords. Falling, Machine Learning, Neural Networks, Mobile Applications, Reminder Systems.

1. Introduction

Elderly people face a number of challenges in their daily lives which makes it necessary for many of them to be looked after by their relatives or health care professionals. Such challenges include forgetfulness and the tendency to fall as well as the fact that elderly people tend to live alone after the death of a spouse. The number of people suffering from dementia in e.g. Austria is expected to double from 120,600 in 2010 to 262,200 in 2050 [1]. The numbers may scale to other industrial countries as well. Forgetfulness is especially risky when those affected are prescribed to a medical therapy, because forgetfulness is one of the main factors that contribute to low medical adherence [2], which is associated with decreased therapeutic success and higher mortality [3]. Their tendency to fall is another risk for elderly people. 30% of people aged 65 years or older fall at least once every year, and 20% of those require medical treatment afterwards [4]. The physical injuries sustained from such incidents are not the only consequence: 60% of elderly people with a history of falling develop a fear of falling [5], which is associated with a loss of physical capabilities which ultimately leads to a decrease in mobility level, therefore making the person more likely fall again [5][6]. In case of falling or any kind of emergency it is important that elderly people have easy access to help, like a phone or a spouse that is able to assist them. With growing age the number of people living alone

[1] Corresponding Author: Harald Burgsteiner, Graz University of Applied Sciences, eMail: harald.burgsteiner@fh-joanneum.at

is growing [7], which makes falling an even more serious danger, because half of elderly people cannot get up by themselves after a fall, even if they are not injured [8].

The advance of smart technologies in recent years offers new ways of enabling elderly people to live a safe and self-determined life with minimal dependence on other people even at a high age. Smartphones and smartwatches make it easy to get in touch with relatives or emergency services if required, and allow them to send notifications to the user's wrist. Furthermore, smartwatches can track the behaviour of their users even if the smartphone is out of reach, detect unusual behaviour and react accordingly.

The purpose of this paper is to investigate the possibilities and challenges when creating an assistance system for elderly people based on consumer devices. It proposes a smartwatch-based assistance system which is able to recognize emergency situations by performing fall detection and inactivity recognition, besides offering the possibility to set medication reminders and get help from relatives on the push of a single button.

2. Methods

To implement the smartwatch-based assistance system ("Carrie") a suitable smartwatch was selected and fall detection, inactivity recognition as well as medication reminders were developed based on current research in the respective areas.

2.1. Smartwatch

An investigation of the smartwatch market showed that a number of watches for iPhones and Android devices exist. For the use in the scenarios described above, a smartwatch with three dimensional acceleration sensors is needed. Additionally, the battery life should supports many days of activity before recharging is required. For the implementation of the smartwatch-based assistance system the Pebble smartwatch was chosen because it is capable of serving both Android- and iOS-powered devices, has a battery runtime of up to 7 days according to the manufacturer, is available for less than 100 Euros and can be customized by creating own apps with an actively maintained SDK. Its ability to vibrate and the built-in acceleration sensor make it possible to alert the user and track his or her activity while wearing the watch. Other smartwatches are usually more expensive, only available for one platform or have a poor battery capacity.

2.2. Fall detection

The objective was to detect falls based on the acceleration data captured by the smartwatch. A lot of research exists in the field of accelerometer-based fall detection and [9] distinguishes between analytical and machine learning-based approaches to fall detection. While the former is mostly threshold-based, the latter uses mathematical models for the automatic classification of movements after a training period. For Carrie an approach based on machine learning utilizing a multilayer perceptron was chosen. To train the multilayer perceptron it was necessary to collect data of activities of the daily life and falls. Seven healthy subjects (aged 14, 23, 25, 68, 83 and two women of 76) were asked to walk, run, get up and sit down while the watch was recording their movements. The three youngest participants were also asked to perform falls to the left, right, front and back. Additionally data was recorded while the watch was vibrating and while the people were idle. Data was collected at 50Hz by the smartwatch's 3D acceleration sensor and transmitted to the smartphone for further processing, because the smartwatch lacks the computing power necessary to perform fall detection [10]. Since the watch is not able to continuously transmit the x-, y- and z-readings of the sensor at a rate of 50 times per second, the Euclidean norm of the values was calculated and sent to the phone where the features needed for training the classifier were extracted:

$$Euclidean\ norm := \sqrt{x^2 + y^2 + z^2} \tag{1}$$

By using the Euclidian norm the information about the direction of the movement and acceleration gets lost. Tests showed that this information is not needed to only detect whether a fall occurred or not. A default input signal (DIS) was extracted from the data stream coming from the smartwatch by identifying the largest deviation from the resting potential (the Euclidean norm at rest is 1000) in the data recorded during the previous 7 seconds. The DIS consists of three seconds of data starting 1.5 seconds before the identified peak and is used to perform further feature extraction.

Three algorithms for fall detection based on a multilayer perceptron with one hidden layer and 10 output neurons (one for each type of movement: idle, walking, running, getting up, sitting down, vibration, falling forward, backward, left and right) were compared. Similar to [11], approach 1 used the DIS directly as network input, thus using 150 input neurons (three seconds of data with 50 samples per second). Approach 2 extracted features from the DIS and used the maximum (A in Figure 1) in the DIS which marks the impact, the minimum value that occurred before A which is the suspected end of the fall (B), the difference in acceleration and time between A and B (intensity of fall and duration of impact), and the duration of the fall which is defined as the time between B and the maximum value before B which is the suspected start of the fall (C). Therefore approach 2 uses 5 input neurons. Approach 3 uses additional features to further improve fall detection: the mean and average values of the DIS were added to incorporate a global

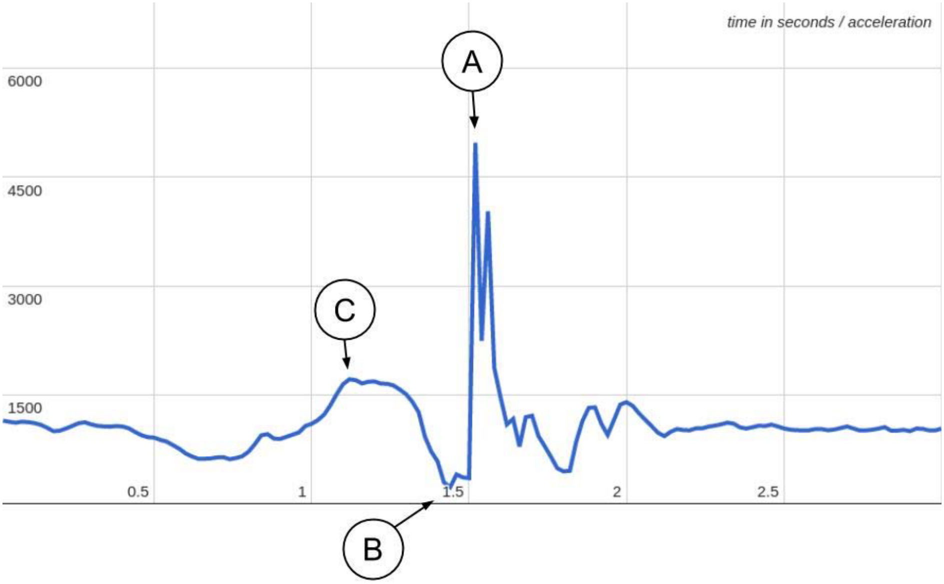

Figure 1: From the DIS features like the suspected impact (A), the end of the fall (B) and the begin of the fall (C) are extracted.

perspective on the sample as well as the difference in acceleration between start and end of the fall (B and C). Eight input neurons were therefore used for classifier 3.

All three algorithms were implemented and an individual neural network was trained and evaluated for each algorithm respectively. Sensitivity and specificity were determined for every algorithm. We also studied the effect of different numbers of neurons on the hidden layer. Algorithm 3 showed slightly better accuracy and significantly better sensitivity compared to the other approaches. It was outperformed slightly by algorithm 1 in terms of specificity, but performed better than all other algorithms when using 10 neurons in the hidden layer. The network was therefore implemented using approach 3 with 8 input neurons, 10 neurons on the single hidden layer and 10 neurons in the output layer.

WEKA (see [12]) was used to create the neural network. Since WEKA does not work on Android out of the box it was stripped of all classes and functionalities that caused compile-time errors when building it for Android. The resulting WEKA package can be used in an Android project to detect falls in real time. After providing WEKA's multilayer perceptron implementation with a network input it returns output values for each possible output class. The class with the highest output value determined the detected movement or fall ("winner-takes-all"). Falls were only reported if the highest resulting output value was associated with a fall and was at least 89%.

2.3. Inactivity recognition

In order to detect suspicious inactivity it is necessary to know the time that has passed since the last activity and usual behaviour patterns of the user. The approach used for inactivity recognition is very similar to the one described by Cuddihy et al. (see [13]), who record the duration of ongoing inactivity twice every hour and match it against a threshold which is calculated based on past inactivity durations that have been recorded on the same time of the day. It was modified to record the duration of ongoing inactivity four times every hour and respect weekly recurring behavioural patterns.

Cuddihy et al. calculate the threshold using the configurable parameters *MP*, *UBP* and *VBP* which are described with the values found reasonable by Cuddihy et al. in Table 1. The interval weights W_r have been adapted to include four more weights in order to respect the hour before and after the given time. The minimum threshold has been set to 30 minutes so that inactivity is never flagged as suspicious if it is 30 minutes or less.

Table 1: Configurable parameters used for inactivity recognition. All parameters and values (except parameter MT and the values for the weights) have been suggested by Cuddihy et al.

Parameter	Description	Value
Maximum Percentile (MP)	The percentile of data considered when determining the threshold. It is used to eliminate outliers.	0.97
Uniform Buffer Percentage (UBP)	The percentage by which the maximum inactivity should always be increased when calculating the threshold.	0.30
Variable Buffer Percentage (VBP)	Determines how much the surrounding intervals affect the threshold. A low VBP increases sensitivity at usually active times.	0.40
Interval Weights (W_r)	Controls the influence of each of the surrounding intervals, where r is relative to the current interval: -4 <= r <= 4.	1,2,3,3,4, 3,3,2,1
Minimum Threshold (MT)	The minimum threshold in minutes that needs to be exceeded in order to create an alert.	30 minutes

The threshold for a time interval i (where i = 0 for 00:00 o'clock, i = 1 for 00:15 o'clock) is calculated using the formula

$$threshold_i = MAX(M_i+UB_i+VB_i,\ MT) \qquad (2)$$

where M_i is the longest inactivity duration that has been recorded one hour before and after the current interval and is calculated as follows:

$$M_i = MAX(m_{i-4.},\ m_{i-3},\ m_{i-2},\ m_{i-1},\ m_i,\ m_{i+1},\ m_{i+2},\ m_{i+3},\ m_{i+4}) \qquad (3)$$

m_i is the 97th percentile of the previously recorded inactivity durations at interval i and is used instead of the real maximum in order to eliminate extremes.

Cuddihy et al. describe UB as a buffer that is proportional to the longest inactivity in the current timeframe and makes sure that inactivities that are slightly longer than the recorded maximum are allowed:

$$UB = UBP * m_i \qquad (4)$$

Since UB should actually be different at every interval a slightly modified formula was used to reflect the values at each time interval i:

$$UB_i = UBP * m_i \qquad (5)$$

VB_i is a variable buffer that is calculated by computing the weighted sum of variable buffers of all relevant intervals. Each timeframe is assigned a weight W_i that depends on the position of the timeframe relative to the current interval.

$$VB_i = \frac{1}{\sum W_r}\left[\sum_{r=-4}^{4}(VBP * m_{i+r} * W_r)\right] \qquad (6)$$

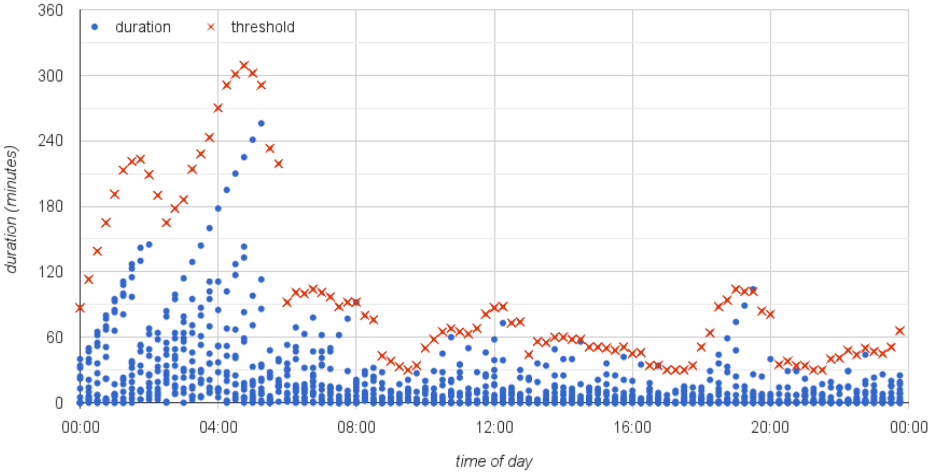

Figure 2: Durations of ongoing inactivity are represented as blue dots. The calculated thresholds are shown as red crosses.

Inactivity durations are recorded together with the number of the interval at which they occurred. The resulting data can be visualized as in Figure 2 where the calculated threshold for each time of day is visible as well. To calculate the threshold, data recorded on the 45 previous days is used. In order to make up for different behaviour of the user on different days of the week (like sleeping longer on weekends) same days of the week are given a higher chance of being used in the calculation: of 45 inactivity recordings up to 27 were recorded on the same weekday. The remaining inactivity recordings are taken from the most recent days. Thresholds are calculated once every day at midnight for the following day.

2.4. Energy efficiency

The constant stream of acceleration data from the Pebble smartwatch to the smartphone via Bluetooth resulted in the smartwatch's battery being drained after approximately 17-19 hours. This is much less than the anticipated seven days of normal operation without fall detection. An analysis showed, that the highest share of energy consumption comes from the wireless transmission of data. Thus, the application on the watch was reconfigured to not transmit a continuous stream of data but to decide about possible falls on the watch itself and restrain data transmission to these suspicious values only. For this pre-detection we retain the acceleration data of the last 6 seconds and only start transmitting the data if a certain threshold has been exceeded. The threshold was set to the Euclidean norm of 1,700 which can be calculated from the raw acceleration data. If a value higher than that is observed the smartwatch starts transmitting data for 30 seconds, sending both data that was recorded before, during and after the possible fall. This had the effect that battery runtime could be increased by at least 180%, from 17-19 to 50-60 hours, depending on the user's activity. The time of the last reception of acceleration data is also used on the smartphone to determine the currently ongoing inactivity for inactivity recognition.

3. Results

The resulting system is capable of detecting falls and suspicious inactivity, as well as providing a way of manually calling for help and issuing medication reminders using a smartwatch and a smartphone.

The assistance system consists of a Pebble smartwatch that communicates with an Android smartphone via Bluetooth. In case of emergency situations alerts are sent to configurable emergency contacts via text messages. It is essential that the smartwatch and smartphone are connected at all times as the smartwatch cannot perform fall detection on its own or send notifications because it does not have any network interfaces except Bluetooth.

Emergency detection is performed using fall detection, inactivity recognition and manual activation. A user can also manually issue an alert by pressing a dedicated button on the smartwatch. To reduce the risk of accidentally issuing a manual alert the smartwatch starts to vibrate and display a notification for 30 seconds during which the user has the chance to cancel the alert by pressing a specific button on the watch. Upon cancellation the currently ongoing inactivity duration is reset and fall alerts are inhibited for a period of one minute in case the cancelled alert originated from a misdetected fall.

If the user does not cancel the alert a predefined list of emergency contacts is notified of the situation. The alert contains the type of emergency and when an accurate GPS signal is available it also contains the user's address and a link to open Google Maps at the user's current geolocation so that the system is also of use if the user is outdoors, for example going for a walk. In case none of the contacts respond via call or text message within 5 minutes they are notified again. Emergency contacts can be configured on the smartphone by picking contacts from the contacts list or manually entering numbers. Contacts can be configured to only be notified if other contacts did not respond within 5 minutes. Hence, it is also possible to notify official help (e.g. an ambulance) if none of the primary helpers react.

To further enhance the usefulness of our application, the Carrie app on the smartphone can also be configured to issue medication reminders on different times of day. Alerts are then pushed at certain times to the smartwatch which starts to vibrate and shows a medication symbol. We decided that no information about which medication to take is included because the display can be hard to read for elderly people and it is assumed that either the user knows which pills to take or the medication has been prepared for them.

4. Discussion

The combination of fall detection, inactivity recognition and manual activation creates a smartwatch-based emergency detection system that promises high reliability because the detection mechanisms complement one another. For example if falling detection does not work the user can still manually trigger an alert or, if unable to do so, suspicious inactivity will be reported after some time has passed. A major weakness in the system's architecture is the heavy dependence on the availability of a smartphone which should therefore always be within a few meters of the user (signal range of the Bluetooth connection).

Research by Bagalà et al [14] has shown that machine learning algorithms that had been trained using staged falls tend to be less accurate when applied to real falls. Therefore, in the next phase of the project, real movement data of active elderly people will be collected in controlled environments like nursing homes to be able to better train the neural network. Additionally, the fall detection algorithm might yield better results when it is customized for the user. Therefore it might prove beneficial to walk the user through a setup process when first using the system, where he or she is instructed to perform activities of daily live which, together with pre-recorded falls, are used to train the neural network online. Furthermore, if a fall is mistakenly detected and the user cancels the alert, the signal that caused the alert could be used to further train the network so the same movement won't be classified as fall in the future. Further improvements to the fall detection could include monitoring a period of time after a suspected fall. If there are no normal activities an alert can be issued, while it can be cancelled if it is detected that the user is for example walking again. Future research should focus on evaluating and improving the emergency detection capabilities in field tests and clinical studies, as well as testing the system's acceptance and usability on elderly people.

References

[1] Höfler, Sabine; Bengough, Theresa; Winkler, Petra; Griebler, Robert (Hg.) (2015): Österreichischer Demenzbericht 2014. Bundesministerium für Gesundheit und Sozialministerium, Wien.
[2] Vervloet, M., Linn, A.J., van Weert, J.C., De Bakker, D.H., Bouvy, M.L., & Van Dijk, L. (2012). The effectiveness of interventions using electronic reminders to improve adherence to chronic medication: a systematic review of the literature. Journal of the American Med. Inf. Assoc., 19(5), 696-704.
[3] Rasmussen, J.N., Chong, A., & Alter, D.A. (2007). Relationship between adherence to evidence-based pharmacotherapy and long-term mortality after acute myocardial infarction. Jama, 297(2), 177-186.
[4] Kannus, P., Sievänen, H., Palvanen, M., Järvinen, T., & Parkkari, J. (2005). Prevention of falls and consequent injuries in elderly people. The Lancet, 366(9500), 1885-1893.
[5] Legters, K. (2002). Fear of falling. Physical therapy, 82(3), 264-272.
[6] Vellas, B.J., Wayne, S.J., Romero, L.J., Baumgartner, R.N., & Garry, P.J. (1997). Fear of falling and restriction of mobility in elderly fallers. Age and ageing, 26(3), 189-193.
[7] Statistik Austria, Lebensformen, http://www.statistik.at/web_de/statistiken/menschen_und_gesellschaft/bevoelkerung/haushalte_familien_lebensformen/lebensformen/index.html, last access: 2015-08-02.
[8] Alexander, N. B., Ulbrich, J., Raheja, A., & Channer, D. (1997). Rising from the floor in older adults. Journal of the American Geriatrics Society, 45(5), 564-569.
[9] Noury, N., Fleury, A., Rumeau, P., Bourke, A.K., Laighin, G.O., Rialle, V., & Lundy, J.E. (2007). Fall detection-principles and methods. In Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE (pp. 1663-1666). IEEE.
[10] Maglogiannis, I., Ioannou, C., Spyroglou, G., & Tsanakas, P. (2014). Fall Detection Using Commodity Smart Watch and Smart Phone. In Artificial Intelligence Appl. and Innovations (pp. 70-78). Springer.
[11] Sengto, A., & Leauhatong, T. (2012). Human falling detection algorithm using back propagation neural network. In Biomedical Engineering International Conference (BMEiCON), 2012 (pp. 1-5). IEEE.
[12] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I.H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.
[13] Cuddihy, P., Weisenberg, J., Graichen, C., & Ganesh, M. (2007). Algorithm to automatically detect abnormally long periods of inactivity in a home. In Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments (pp. 89-94). ACM.
[14] Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., Zijlstra, W. & Klenk, J. (2012). Evaluation of accelerometer-based fall detection algorithms on real-world falls. PloS one, 7(5), e37062.