Comparison of fall detection methods and the prospect of model-based: pattern recognition on green computing implementation by using smartphone

Maria Seraphina Astriani¹, Raymond Bahana¹, Andreas Kurniawan¹, Lee Huey Yi²

¹Computer Science Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia 11480
²Master of Cognitive Neuroscience, University of Science Malaysia, Kuala Lumpur, Malaysia 50400

seraphina@binus.ac.id

Abstract. Portable healthcare solution to detect fall is needed for special care people and elderly, especially if they want to live independently. If this solution needs special made device by the factory, it requires huge electric consumption, lots of materials, and can contribute to the environmental pollution. Smartphone is one of the most promising device to be used as fall detection because it has lot of potentials: easy to find, compact, has sensors, and can even be accepted by elderly. There are two main approaches in fall detection: threshold-based method and model-based: pattern recognition method. Threshold-based usually been used in real life because it can be developed quite quickly and use minimal computation process to provide movement classification result (fall or non-fall). Model-based: pattern recognition method (using machine learning) has relatively better accuracy result but requires a long training time, a lot of resources to process, and more difficult to be developed. The researchers present the comparison results of threshold-based and model-based: pattern recognition; and optimizing the model in machine learning so the model has the prospect to be implemented in applications to support green computing.

1. Introduction

Everyone wants to live independently, include special care people and elderly [1]. Fall occurs commonly, especially in the elderly. If they are going alone, fall, and no one is watching them, it can cause a fatal accident if no one knows about this incident so they cannot be treated quickly [2]. Healthcare solution to detect fall is essential and require a portable device [3]. However, if it needs a special device (that must be made specifically by the factory) for example commercial watch (smartwatch), it requires huge electric consumption, lots of materials, and can contribute to the environmental pollution (waste, smoke, etc.) to produce a special device [4, 5]. The large number of electronic devices can cause other problems such as electronic waste (furans, polycyclic aromatic hydrocarbons (PAHs), etc.) which is extremely dangerous to health and can cause severe environmental pollution [5].

There have been many actions taken by the community to save this world and one of them is green computing. Green computing is an environmentally responsible way to minimizing energy consumption and electronic waste [6]. Green computing is the process of using computing resources...
efficiently. The aims are to minimize the use of hazardous materials, improve energy efficiency over the lifespan of the product, and encourage recyclability or biodegradability obsolete products and factory waste. Smartphone is one of the most promising device for this approach since they are easy to find, compact, has sensors (such as gyroscopes and accelerometers) to detect acceleration, orientation and angle of motion in daily life, and are even embraced/accepted by elderly [7, 8, 9].

The use of smartphones can provide solutions for the environment without the need to make special devices, minimize electronic waste, and maximize its potential. However, the use of smartphone has another challenge that lies in the method to detect fall. There are two main methods to detect fall, namely threshold-based and model-based: pattern recognition [10]. Threshold-based has the benefit of being able to develop quite quickly and use minimal computation process to provide classification results, but their capabilities are very limited when they face difficult dataset (example: fall pattern that is very close to non-fall pattern). Model-based: pattern recognition can give relatively better results than threshold-based, but during the process (model) and the implementation requires a large amount of resources and long computation time; this is not in line with green computing. The use of machine learning for the model-based: pattern recognition method has relatively better accuracy results than threshold-based but requires a long training time. Its implementation is also more difficult to develop, so many of applications are using threshold-based method [11, 12]. This study aims to compare threshold-based and model-based: pattern recognition methods and to see the potential/prospects of machine learning (based on the time to classify the data) to be implemented in the application.

2. Literature Review

The method to detect human fall can be divided into two parts: threshold-based and model-based: pattern recognition. Threshold-based detection focuses on the mathematical calculation of threshold implementation while model-based: pattern recognition generally uses machine learning approach [7, 12, 13, 14]. The use of algorithms in fall detection method tends to use model-based: pattern recognition (machine learning) because it can learn from input data so it can classify more complex data patterns and produce good accuracy. However, on the other hand, the use of threshold-based is still a popular method to detect human fall because computation can be done relatively quickly and tends to be easier to implement because it uses mathematical formulas.

There are several mathematical formulas to detect human fall: Accelerometer Gyroscope Vector Signal Resultant (AGVeSR), Alim, and Tilt Angle Change [11, 15, 16]. AGVeSR is the enhancement of Signal Vector Magnitude formula. AGVeSR (1) includes a gyroscope value component to calculate the resultant of the inputs based on accelerometer and gyroscope sensors [15]. If the formula detects a sudden spike in a short duration (time), the formula will classify it as fall. Ali (2) formula is extremely useful to determine linear acceleration by adding gravity vector element calculated from gyroscope sensor. To detect how much linear acceleration occurred, Ali results need to be calculated into Alim (3) to get the sum vector value [16]. Human fall can be detected by the changes in human posture. Tilt Angle Change formula has the capability to calculate the change (in degree) of human posture (4). If the change in posture that occurs is more than 60 degrees, it can be concluded as fall [11]. AX, AY, and AZ notations are X, Y, and Z axes in the accelerometer; the notations of GX, GY, and GZ represent X, Y, and Z axes in the gyroscope.

\[
AGVeSR = \sqrt{([AX] + |GX|)^2 + ([AY] + |GY|)^2 + ([AZ] + |GZ|)^2)}
\]  

\[
Ali = [(AX - GX), (AY - GY), (AZ - GZ)]
\]

\[
Alim(n) = |Ali|
\]

\[
Tilt\ Angle\ Change = \tan^{-1}\left(\frac{AY}{\sqrt{AX^2 + AZ^2}}\right)
\]
There are many models can be implemented in model-based: pattern recognition to detect human fall. Researchers picked Long Short-Term Memory (LSTM) as one of the approaches in model-based: pattern recognition. LSTM has the power to remember long-term relationship patterns where this strength is very suitable when applied to time-series data [17, 18]. LSTM architecture consists of an input, output, and forget gate that allows it to reset its own state, making it possible to learn continual tasks (figure 1). There are two most important parts in LSTM block: Forget gate and output activation functions. The learning computational complexity per time step is $O(W)$ where W can be calculated as follows:

$$W = n_c \times n_c \times 4 + n_i \times n_c \times 4 + n_c \times n_o + n_c \times 3$$

(5)

where $n_c$ is the number of memory blocks (cells), $n_i$ is the number of input units, and $n_o$ is the number of output units [19].

![Figure 1. Long Short-Term Memory (LSTM) cell](image)

3. Methods
Based on literature review, because there are two fall detection methods: threshold-based and model-based: pattern recognition, an experiment was carried out on both models to prove which model is the most optimal in order to detect falling motion especially for the prospect of implementing green computing on a model-based: pattern recognition. Threshold-based detection is based on mathematical linear thresholding calculations, while model-based: pattern recognition can be applied using machine learning.
The proposed threshold-based method is using six values from accelerometer and gyroscope sensors as the input. These values are calculated individually by AGVeSR, Ali-Alim, and Tilt Angle Change formulas. If the result calculated by the formula exceed the threshold, the movement temporary classify as fall. Voting is necessary to get the majority result based on AGVeSR, Ali-Alim, and Tilt Angle Change formulas. the voting results will be used as the classification results. For further reference, this method will be called as three fall formulas (ThreeFFs) (figure 2).

Machine learning takes time to enable algorithms to learn and develop adequately to meet their objectives with sufficient accuracy and relevance. Machine learning also requires a lot of resources to work and that means it requires more computer power. Model-based: pattern recognition (using LSTM) in this study was deliberately made with a minimum number of layers so that the model can run and process data on devices with low power and can be embedded in smartphones by transferring learning/application (figure 3).

The input of the model is the same as in ThreeFFs. It uses six inputs taken from all axes in accelerometer and gyroscope sensors. Each axis is processed by LSTM and there are six LSTM in total (layer 1). The output for each LSTM is concatenated and processed by the fully connected layer in layer 2 to help the model classify the data. As the result of the process, the model produces classification results to determine whether a movement is fall or non-fall.
Figure 3. LSTM model for fall detection

Accuracy is appropriate to use as a performance evaluation in this experiment because the existing dataset (1/3 fall and 2/3 non-fall) not categorized as imbalanced. If the dataset is imbalance, accuracy cannot be used for measurement because it is sensitive to data imbalances in the training phase.

4. Experiment

4.1. Dataset
Primary dataset collected by researchers. All movements (fall and non-fall) performed in an indoor environment. Fall movements are performed on the mattress for safety. Every volunteer needs to use a smartphone attached to the body. A total of 252 data (84 fall data and 168 non-fall data) were used during the training to determine the threshold (on the threshold-based method) and to train the model (on model-based: pattern recognition method). The data used during testing were 324 data, of which 108 were fall data and 216 were non-fall data.

Each movement has a length of 70 time-windows (1.4 seconds) because the longest duration of standing-falling, walking-falling, jumping, sitting, standing and walking movements have maximum of 70 time-windows. Each time-window consists of six values derived from the accelerometer and gyroscope sensors (AX, AY, AZ, GX, GY, and GZ).

4.2. Experimental Design
The experiment used a primary dataset. The movements were monitored by a smartphone attached to the body. The data values from the accelerometer and gyroscope sensors are monitored and stored on the smartphone and then retrieved for the purposes of this experiment. Mattress was used to simulate fall to minimize the risk of injury.

The human movement data that has been obtained are then labeled according to the type of movements being performed. Data that has been labeled are ready to be processed in the methods using ThreeFFs (threshold-based) and LSTM (model-based: pattern recognition, using supervised learning approach because the data has been labeled). Each method needs to be trained first by using training data. Parameter tuning needs to be done so the method can reach at the maximum accuracy results. After obtaining maximum accuracy, the models (based on training results) are tested on testing data (data that has never been used during training) to determine whether the model is overfitting or not. If overfitting, the model only "memorizes" or focuses on a certain training dataset, so it cannot predict correctly if given another similar dataset. If overfitting occurs, generally there will be a big difference in the accuracy values between testing and training (the testing results are not as expected
as during training). In most social science studies, the acceptable margin of error is 3-5%, so this range will be the benchmark for assessing the accuracy of the models (ThreeFFs and LSTM).

4.3. Model Implementation and Result

Each method (threshold-based and model-based: pattern recognition) trained using training data to obtain the optimal model (can achieve the highest accuracy results from the model). In fall detection case, the models need to achieve the highest True Positive (TP) value and the lowest False Positive (FP) because the risk that can occur when a person falls but is not detected as fall is very crucial compared to the non-fall activity detected by the model as fall.

Besides the models need to achieve the optimal accuracy results, time to process the data is an important element to be considered. It is not good if the model can achieve high accuracy but takes long time to process and use large resources. This condition is not ideal to be implemented in the application and is not in line with green computing.

Training data are used in ThreeFFs and LSTM to achieve optimal models. After the training process been done, the models obtained from the training are tested into testing data to find out whether the models are robust to process new data that has never been “seen” during training. The result of training and testing of fall detection methods (threshold-based using ThreeFFs model and model-based: pattern recognition using LSTM model) can be found in table 1.

| Measurements (approximately) | ThreeFFs (Threshold-based) | LSTM (Model-based: pattern recognition) |
|-----------------------------|----------------------------|------------------------------------------|
| Training accuracy           | 0.97                       | 0.91                                     |
| Testing accuracy            | 0.75                       | 0.89                                     |
| Time to process (in seconds) / movement | 0.01                  | 1.73                                     |

The training accuracy result of ThreeFFs (0.97) is higher compared to the training accuracy result of LSTM (0.91). However, when tested using testing data, the result of the accuracy of the ThreeFFs decreased quite dramatically (0.22) and exceeds the commonly used benchmark margin of error. Even though LSTM model has a training accuracy result below ThreeFFs, the testing accuracy result are still in the range of margin of error and this can prove that the LSTM model is robust.

In accordance with the advantages of the threshold-based method, ThreeFFs is very fast to process and classify the data (movement). If we have a look on the time to process using LSTM model, the time required by LSTM to classify data is still logically acceptable and does not take long time (it do not take minutes or even hours) to obtain the classification result. All training and testing process is done in low power consumption device (with low CPU power and other resources). Smartphone can be categorized as one of the low power consumption device. This time to process results are achieved using a device with the following specifications: RAM - 4.00 GB and processor - 1.60 GHz (the baseline for smartphone RAM is 4 GB, based on the PC Magazine article - April 17 , 2019). The promising result on processing time obtained by LSTM make it possible for the implementation of this model on smartphones to achieve green computing.

5. Conclusion

Threshold-based method and model-based: pattern recognition are two methods to detect human fall. Threshold-based usually been used in real life because it can be developed quite quickly and use minimal computation process. Model-based: pattern recognition method has relatively better accuracy results but requires a long training time, a lot of resources to process, and more difficult to be
developed. Based on the methods developed, researchers found the training accuracy (0.97) and time to process (0.01) of ThreeFFs (using threshold-based method) is higher than LSTM (model-based: pattern recognition) that been specially made with minimum number of layers. However, when ThreeFFs tested using testing data, the accuracy result decreased quite dramatically (0.22) and exceeds the commonly used benchmark margin of error. Even though LSTM model has a training accuracy result 0.91, the testing accuracy result (0.89) is still in the range of margin of error and this can prove that the LSTM model is robust. The time required by LSTM to classify data is still logically acceptable (1.73) and works in low power consumption device. Model-based: pattern recognition by using LSTM model do not take long time (it do not take minutes or even hours) to obtain the classification result, and has the prospect to support green computing (can be implemented on smartphone).

Acknowledgments
This work is supported by Research and Technology Transfer Office, Bina Nusantara University as a part of Bina Nusantara University’s International Research Grant entitled Robust Human Fall Detection Method For Health Monitoring with contract number: No.026/VR.RTT/IV/2020 and contract date: 6 April 2020.

References
[1] de la Concepción M Á Á, Morillo L M S, García J A Á and González-Abril L 2017 Mobile activity recognition and fall detection system for elderly people using ameva algorithm Pervasive and Mobile Computing 34 3-13
[2] Xiao M, Huang Y, Wang Y and Gao W 2020 Design of elderly fall detection based on XGBoost Artificial Intelligence in China 46-57
[3] Majumder S and Deen M J 2019 Smartphone sensors for health monitoring and diagnosis Sensors 19 9 2164 1-45
[4] Babu B R, Parande A K and Basha C A 2007 Electrical and electronic waste: a global environmental problem Waste Management & Research 25 4 307-318
[5] Robinson B H 2009 E-waste: an assessment of global production and environmental impacts Science of the total environment 408 2 183-191
[6] Agarwal S, Basu R S and Nath A 2014 Green computing and sustainable environment–introduction of E-documents and replacement of printed International Journal of Innovative Research in Information Security 1 5 46-53
[7] Luque R, Casilari E, Morón M and Redondo G 2014 Comparison and Characterization of Android-Based Fall Detection Systems Sensors 14 10 18543-18574
[8] Hsieh S L and Yang C T 2016 Detecting falls with low-end smartphones IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC) 2016 1-5
[9] Figueiredo I, Leal C, Pinto L, Bolito J and Lemos A 2016 Exploring smartphone sensors for fall detection mUX: The Journal of Mobile User Experience 5 1 1-17
[10] Xu T, Zhou Y and Zhu J 2018 New advances and challenges of fall detection systems: A survey Applied Sciences 8 3 418.
[11] Weng W X and Lo S C 2016 Fall Detection Based on Tilt Angle and Acceleration Variations. Trustcom/BigDataSE/I SPA IEEE 1712-1717
[12] Shi T, Sun X, Xia Z, Chen L and Liu J 2016 Fall detection algorithm based on triaxial accelerometer and magnetometer Engineering Letters 24 2
[13] Delahoz Y S and Labrador M A 2014 Survey on fall detection and fall prevention using wearable and external sensors Sensors 14 10 19806-19842
[14] Medrano C, Igual R, Plaza I and Castro M 2014 Detecting falls as novelties in acceleration patterns acquired with smartphones PloS one 9 4 e94811
[15] Astriani M S, Heryadi Y, Kusuma G P and Abdurachman E 2019 Human fall detection using accelerometer and gyroscope sensors in unconstrained smartphone positions International
[16] Hsu Y W, Chen K H, Yang J J and Jaw F S 2016 Smartphone-based fall detection algorithm using feature extraction *Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* 1535-1540

[17] Kalchbrenner N, Danihelka I and Graves A 2015 Grid long short-term memory *arXiv preprint* 1507.01526

[18] Lu N, Wu Y, Feng L and Song J 2018 Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data *IEEE journal of biomedical and health informatics* 23 1 314-323

[19] Sak H, Senior A and Beaufays F 2014 Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition *arXiv preprint* 1402.1128

[20] Yu Y, Si X, Hu C and Zhang J 2019 A review of recurrent neural networks: LSTM cells and network architectures *Neural computation* 31 7 1235-1270