An Accurate Fall Detection System for the Elderly People Using Smartphone Inertial Sensors

A A Kadhum¹, H Al-Libawy¹ and E A Hussein¹

¹Department of Electrical Engineering, University of Babylon, Iraq

abeer.alhimiery@yahoo.com

Abstract. In developed countries, the number of elderly people living alone is continuously increasing. These people are more vulnerable to serious health issues, such as falling down. A sensor-based system, augmented to mobile phones, can provide a much-needed prediction to the falls, where injuries and fracture possibilities can be significantly decreased. The purpose of this study is to develop a fall recognition system based on smartphone inertial sensors, which is a combination of accelerometer and gyroscope. The system can distinguish between falls and other activity daily livings (ADLs). The data output from the inertial sensor have been used by two different classifiers; artificial neural network (ANN) and support vector machine (SVM), where the objective is to find an accurate falling classifier using smartphone inertial sensors. Results show that SVM based classifier offers an accuracy of 99.27%, which outperforms the state of the art results that use smartphone data.

1. Introduction

Population ageing is about the most common problem worldwide. Based on a report prepared by the World Health Organization (WHO), report an elevated trend in elderly people with 60 years or more. WHO estimates that 20 billion of people in the world are elderly by 2050 compared to only 900 million in 2015. This accounts for 22% of the world's Population (World Health Organization. (2015). WHO). As the age increases, the risk of falls will increase [1], and this is due to muscle weakness and balance instability. Falls are defined as “the inadvertent settling down with a body on the ground, floor or other lower level” [2]. They are common in elderly’s life and can lead to physical and psychological consequences [3], and these consequences can be reduced by early detection of the falls. Researchers of [4], [5] have illustrated a fall event as four stages. These stages are pre-fall, critical-fall, post-fall, and recovery fall phases. Pre-fall phase means a person performs activities of daily living (ADL) such as walking, sitting down, where these events are not considered a fall. Critical phase refers to a person of body movements that begins with inclination towards a lower level and ends with impact; this phase takes an estimated time of time (300 – 500) msec. This interval of time is defined as lead detection time (LDT) [6], [7]. The post fall phase means a person lying on the floor and deceleration. The recovery phase means persons sit or stand up by themselves or helped by another.

Human Activity Recognition (HAR) grabbed many research attentions from many researchers, because of its famous applications such as smart environment health care [8]. This activity daily livings (ADLs) can be recognized either by using sensors (wearable IMU sensors, ambient sensors, and vision sensors) or smartphones, the data are collected from these sensors for more processing and analysis by using different algorithms. Many research discussed ADLs recognition system based on
sensors. For example, the authors of [9], [10] and [11] used wearable sensors. Arifoglu and Bouchachia [12] used environmental sensors. Tao et al. [13] used a combination of inertial and visual sensors, the results indicated that vision-based sensor outperforms accelerometer by an average of (14.58%) accuracy, Privacy issues making visual sensors not to be the best choice.

In the works of [14], [15], and [16] the authors have used smartphone based sensor by sensing acceleration from the accelerometer. Two types of sensors have been designated for data gathering from smartphones (triaxial accelerometers and gyroscopes) by Hassan et al. [8], while the data were collected from three types of sensors that embedded in smartphone (accelerometer, gyroscope and magnetometer) by Nurhanim et al. [17] and the results showed high accuracy fall detection.

The fall detection system detects the impact of the body to trigger an alarm [1]. Many fall detection studies used wearable sensors. Yachirema et al. [18] and Aziz et al. [19] Developed an accelerometer based system to detect falls. Gyroscope sensor can be used for fall detection as shown in [20]. The author of [21] developed fall detection system based on wearable sensors (accelerometer, gyroscope and magnetometer), and achieved accuracy, specificity, and sensitivity all above 99%. Only pressure sensor(s) were used to sense falls in [22]. Shahzad and Kim [23] Used smartphones to detect falls and achieved an accuracy more than (97%), Madansingh et al. [24] Used the embedded sensors available on a smartphone (i.e., Accelerometer, gyroscope and magnetometer), while Hakim et al. [25] Developed fall detection system by using the built inertial measurement unit (IMU) sensors of a smartphone. A smartphone based fall detection was also developed in [26] and [27], the author in the first research used acceleration data, while the author of second research used acceleration and angular velocity data to detect falls. A combined smartphone and smart watch fall detection system were proposed by Vilarinho et al. [28] to collectively form a fall detection system. Both devices have a 9-axis motion sensor combining a 3-axis accelerometer, 3-axis gyroscope, and 3-axis compass. The communication between these devices occurs via Bluetooth low energy (BLE).

Pre-impact fall detection means that falling event can be detected in advance in the pre-fall or in critical fall phase [29]. Fall prediction/or prevention systems try to predict falling early by analysing the sensor outputs [2], and can minimize the health service costs, because they may help to prevent unintentional falls that might cause serious injuries [30]. Many researches and studies have been developed to yield a fall prediction system based on smartphones or different types of sensor. For example, Shahzad et al. [31] used only accelerometer wearable sensor and achieved a high accuracy fall prevention. Nuth Otanasap [29] used the same type of sensor at different location and achieved an accuracy of (97.40%). Ahn et al. [32] developed pre-impact detection system based on data collected by a custom-made inertial measurement unit (IMU) and achieved accuracy of (100%), while Fino et al. [33] used three wireless inertial measurement units (IMUs) to predict falls. Kinect Sensor 2.0 was used by Li et al. [34], the results showed a high accuracy of fall prediction.

The smartphone based fall prediction was proposed by Shen et al. [35] and Hemmatpour et al. [30]. The data were acquired from an accelerometer sensor in the first study, while in the second study accelerometer and gyroscope were used for data collection. The advantages of wearable devices are the lower cost; installation is not complicated, easy to operate. The largest disadvantage of these devices is an intrusion and interaction with the individual [25]. Smartphones built in sensors in addition to high performance, large memory capacity and they are widely used by everyone to make them very good choice for fall detection or prediction.

In this paper, a smartphone-based fall detection system is proposed and evaluated. The data acquired in this study are from imbedded sensors in a smartphone. These data were analysed and pre-processed to be ready for classification stage. The rest of this study is organized as follows: In Section 2 fall detection algorithms are presented. Section 3 presents materials and methods of this study. Section 4 and 5 present the results and the conclusions respectively.
2. Fall Detection Approaches
The collected data are pre-processed by different methods to be ready for classification. Fall detection/prevention can be categorized based on the applied algorithms into the threshold, non-threshold and fusion-based algorithms [1], as shown in figure (1) below:

![Figure 1. Classification of Fall detection/ fall prevention according to the applied algorithm [1]](image)

These algorithms are used to distinguish ADLs from falls. Presently, most fall detection / or prevention researches apply Threshold-based algorithms to detect or prevent falls. Appropriate thresholds must be chosen because the low threshold value may lead to false alarms, while high threshold values lead to huge amounts of missing falls [1]. As mentioned before, many articles used this approach such as [36], [37] and [38] on fall detection, while Hemmatpour et al. [39], Sivaranjani et al. [40] and Nuth Otanasap [41] suggest an approach for fall prevention that is entirely based on threshold.

As shown in figure (1) non-threshold-based methods were divided into two other approaches; machine learning based or statistical based algorithms. The most commonly used algorithms in machine learning are; Support Vector Machine (SVM), Naïve Bayes, neural networks (NNs) Hidden Markov Mode (HMM), Random Forest, Fuzzy Logic, Decision Tree, etc. Min et al. [42] used one of the popular machine learning approaches called (SVM) to recognize between falls and other ADLs. Machine learning can also be used in fall prediction as published in [43], which used SVM algorithm and [44], which used neural networks to prevent falls. The statistical based approach can also be used in fall detection or prevention as proposed in [45] on fall detection, and [46] on fall prediction. Both machine learning and statistical methods can achieve higher accuracy than threshold methods while the main limitation of these methods is that they are complex computing process compared to a threshold-based method [1].

In order to increase the accuracy of the system, a fusion based method combining threshold and non-threshold method. As shown in figure (1), this method can be divided into two classes; the first class is homogeneous-based method, which is proposed by Poonsri and Chiracharti [47] to detect falls...
and Su [48] to predict falls. The second class is heterogeneous-based method as developed by Khojasteh et al. [49] on fall detection and Kutchka et al. [50] on fall prevention. The accuracy is very important in detecting falls, machine learning has higher accuracy than other approaches in fall detection or prevention and it was used in this study. Two types of this approach were used: Support Vector Machine (SVM) and Neural Networks (NNs).

2.1. Support Vector Machine (SVM)

In the last years, Support Vector Machines (SVMs) have become as powerful tools for solving classification and, regression problems [51], [52]. The SVM attempts to create either a line or hyperplane between two groups of data. As shown in figure (2) the circles in black colour represent (class A) while squares in white colour represent (class B). The SVM tries to place the boundary in such a way as to approve that the distance between the nearest data point in each class and the boundary is maximal. The margin can be defined by the nearest data points, and these data points are known as support vectors (the gray circles and square). Support vectors include all the information needed to define the classifier [53].

Mathematically, an SVM can be represented easily [53]. For any point that lies on our boundary line, we can write

\[ (w \cdot x) + b = 0. \] (1)

Where:
- \(w\): Vector that represents the boundary.
- \(x\): Input.
- \(b\): Scalar threshold value.

At the margins H1 and H2, the equations for class A and class B are:

\[ (w \cdot x) + b = 1. \] (2)

And

\[ (w \cdot x) + b = -1. \] (3)

Respectively.

For a given data, anything that belongs to class A will suit to the function

\[ (w \cdot x) + b \geq 1. \] (4)

\[ (w \cdot x) + b \leq 1. \] (5)

Figure 2. Separation of two classes by SVM [53]
For class A and B respectively. A decision function can be made by combining these two functions to decide if a given data belongs to class A or class B as shown below:

\[ f(x) = \text{sign}((w \cdot x) + b). \]  

(6)

In order to find the best boundary between data groups, we will find a solution of \( w \) that allow this. The general form of this solution will be as follows:

\[ w = \sum_{i=1}^{n} v_i x_i. \]  

(7)

\( x_i \): The support vectors that have been kept from training.

Substituting (7) in (6) we get:

\[ f(x) = \text{sign}(\sum_{i} v_i (x \cdot x_i) + b). \]  

(8)

Sometimes there will be cases where the two classes will not be separated properly by the linear boundary in two dimensions that has been constructed above, we can overcome this problem by transforming the data into higher dimensional space, a hyperplane is created which allows linear separation in the higher dimension. In SVMs, we can achieve this by using a transformation, \( \Phi(x) \), which converts the data from an N-dimensional input space into Q-dimensional feature space.

\[ s = \Phi(x). \]  

(9)

Where \( x \in \mathbb{R}^N \) and \( s \in \mathbb{R}^Q \).

Figure (3) shows the change data after transformation.

Substituting (9) in (8), we get:

\[ f(x) = \text{sign}(\sum_{i} v_i (\Phi(x) \cdot \Phi(x_i)) + b). \]  

(10)

The above transformations into higher dimensional space are computationally intensive, because of performing the dot product on the results. This transformation and dot product can be performed in one step by using a kernel. In this way we can reduce the computational load, but keep the higher dimensional transformation’s effect.

The kernel function, \( K(x,y) \) is defined as follows:

\[ K(x,y) = \Phi(x) \cdot \Phi(y). \]  

(11)

Substituting (11) in (10), we get the final basic form for SVM as follows:

\[ f(x) = \text{sign}(\sum_{i} v_i k(x, x_i) + b). \]  

(12)
2.2. Neural Networks (NNs)
Neural networks have been developed rapidly since around 1985 and are now used widely. The subject area is subjected by two areas of study; feed-forward networks, famous as multilayer perceptron, used for classification, and symmetric recurrent networks, also famous as attractor neural networks, used as associative memories[51].

In our experiment, support vector machine and multilayer perceptron (MLP) ANNs were used for the purpose of classification. The MLP used consists of three hidden layers of (11, 9, 9) nodes, and one output layer. The two types of machine learning were tested by using Waikato Environment for Knowledge Analysis (WEKA) version 3.8.3 (c) 1999 – 2018. It was developed at the University of Waikato in New Zealand. WEKA provides a collection of many tasks including data pre-processing, clustering, and classification of many algorithms, regression, visualization, and feature selection. All functions take their input in the form of a single relational table in the form of ARFF files. We can use WEKA in an easy way through a graphical user interface called Explorer [54]. Figure (4) shows the GUI explorer of WEKA.

In [51], a simplest an common form for feed forward neural network with one hidden layer as shown in figure (5).
The input units in the input layer distribute the inputs to the hidden units in the hidden layer. These units, sum their inputs, a constant (the bias) is added and the fixed function \( \phi_h \) of the results is taken. The output units are of the same form, with output function of \( \phi_o \), so we get:

\[
y_k = \phi_o(\alpha_k + \sum_j w_{jk} \phi_h(\alpha_j + \sum_l w_{lj} x_l)).
\]

(13)

The activation function \( \phi_h \) of the hidden layer is almost taken to be the logistic function:

\[
l(z) = \exp z / (1 + \exp z).
\]

(14)

The output units are linear, threshold or logistic units.

The threshold units have:

\( \phi_o(x) = 1 \) (\( x > 0 \))

The biases \( \alpha_i \) can be eliminated by introducing an input unit, which is always at +1 and feeds every other unit. The function \( f \) is then parameterized by the set of weights \( w_{ij} \).

1 for every link in the network or 0 for links which are absent.

### 3. Materials and Methods

#### 3.1. Equipment Used

A smartphone is used to develop a fall recognition system with its built in sensors. Redmi note 7 by Xiaomi was selected. It features an embedded IMU (accelerometer, gyroscope, magnetometer, gravity, linear acceleration, rotation sensor) as shown in figure (6). The accelerometer and gyroscope data are acquired from a smartphone (from sensor kinetics application) at a sampling rate of (400) sample per second, and transmitted to PC in the form of excel files which is imported to MATLAB for analysing and plotting.

#### 3.2. Experiment

Four young healthy volunteers participated in this experiment, ranging in age from 30 to 35 years as shown in figure (8) a, b. They performed four types of falls:

- Forward fall
- Backward fall
In addition, to five types of ADLs:
- Walking
- Upstairs
- Downstairs
- Seat to stand
- Stand to seat

Two trials were performed for each type of fall or ADLs. The first trial was performed on the right hand, while the second trial was performed in pocket for male or in bag for female. The acquired accelerometer and gyroscope data were plotted by MATLAB. Figure (7) a, b shows the resulted plot for forward, fall based on the data acquired from accelerometer and gyroscope.

Figure 6. Screenshots of sensor kinetics application.
Figure 7. (a) Acceleration during forward fall. (b) Angular velocity during forward fall.

Figure 8 a, b. A person performs lateral and forward falling.

3.3. Signal processing
Signals are acquired from smartphone sensors; accelerometer and gyroscope. Features are obtained by using different feature extraction methods, after that these features are labelled to different classes to
be ready for the classification by using different algorithms. Figure (9) propose a simple chart for the above processing system.

3.4. Feature Extraction
Features are acquired based on a different type of feature extraction methods such as mean, variance, median, maximum, minimum, standard deviation, skewness, kurtosis, range, etc. feature extraction process transforms high dimensional data into a significant representation of reduced dimensional data, the benefit of this process is to make classification of these higher dimensional data much easier [9]. In this study, we used two kinds of feature extraction, Mean and standard deviation.

3.5. Labelling
Four types of falls and five types of activity daily living have been carried out with two trials; the first trial was performed on the right hand, while the second trial was performed in pocket for male or in bag for female. All types of falls were labelled as class A (fall), while activity daily livings ADLs were labelled as class B (non-fall) as shown in table (1).
### Table 1. Different activities and their labels

| Description of Physical activities                        | Label | No. of trials |
|-----------------------------------------------------------|-------|---------------|
| Forward fall, backward fall, lateral fall left, lateral fall right. | A     | 2             |
| Walking, upstairs, downstairs, seat to stand, stand to seat. | B     | 2             |

#### 3.6. Classifier

The features were extracted from the data to be ready for classification as inputs. In this study, we made a comparison between two different classification algorithms by using Waikato Environment for Knowledge Analysis (WEKA):

- Multilayer perceptron: with training data of (70%) with three hidden layers of (11, 9, 9) nodes respectively, momentum 0.2 and learning rate 0.3.
- SVM classifier: with a training data set of (70%) and gamma, set to (0.15).

#### 3.7. Performance Evaluation

The most common criteria that are used for evaluating the performance of classification algorithms are accuracy, recall or sensitivity, specificity and precision.

Accuracy represents the correctly classified examples and can be determined as follows:

\[
ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)}
\]

(15)

While sensitivity, precision and specificity are defined as follows:

\[
Recall = \frac{TP}{(TP+FN)}
\]

(16)

\[
Precision = \frac{TP}{(TP+FP)}
\]

(17)

\[
Specificity = \frac{TN}{(TN+FP)}
\]

(18)

Where:

- **FP**: False positive, which means that the algorithm detects a fall when there is no fall.
- **FN**: False negative, which means that the algorithm does not detect a fall when it happens.
- **TP**: True positive, which means that a fall happens and algorithm detects it.
- **TN**: True negative, which means that there is no fall and algorithm does not detect it.

#### 4. Results and Discussion

After testing the dataset by using two types of machine-learning algorithms (neural networks NNs and support vector machine SVM) in WEKA, the results were obtained as follows:

- **NNs algorithm**

| TP Rate | FP Rate | Precision | Recall | F-Measure | ROC Area | Class |
|---------|---------|-----------|--------|-----------|----------|-------|
| 0.942   | 0.029   | 0.970     | 0.942  | 0.956     | 0.983    | A     |
| 0.971   | 0.058   | 0.944     | 0.971  | 0.957     | 0.983    | B     |

The accuracy was (95.63%) and the confusion matrix is shown in the table (3):
Table 3. Confusion matrix.

| Actual | Predicted |
|--------|-----------|
|        | A  | B  |
| A      | 5493 | 340 |
| B      | 171  | 5715 |

- SVM algorithm

Table 4. The obtained results of SVM classifier.

| TP Rate  | FP Rate | Precision | Recall | F-Measure | ROC Area | Class |
|----------|---------|-----------|--------|-----------|----------|-------|
| 0.989    | 0.003   | 0.997     | 0.989  | 0.993     | 0.993    | A     |
| 0.997    | 0.11    | 0.989     | 0.997  | 0.993     | 0.993    | B     |

The accuracy was (99.27%), Table (5) shows the confusion matrix for this type of classifier.

Table 5. Confusion matrix.

| Actual | Predicted |
|--------|-----------|
|        | A  | B  |
| A      | 5768 | 65  |
| B      | 20  | 5866 |

The results show that the accuracy of the SVM classifier to detect falls was higher than the accuracy of NNs classifier. By using the equations of performance evaluation discussed in section (3.7), the sensitivity, specificity, and precision for each algorithm is implemented in table (6).

Table 6. The performance of NNs and SVM algorithms.

| Algorithm | Accuracy% | Sensitivity% | Specificity% | Precision% |
|-----------|-----------|--------------|--------------|------------|
| NNs       | 95.63     | 94.17        | 97.09        | 96.98      |
| SVM       | 99.27     | 98.88        | 99.66        | 99.65      |

The results indicated that the performance of SVM algorithm was better than NNs algorithm. A comparison in performance of the implemented system with those from other previous studies is made in table (7):

Table 7. Sensitivity and specificity comparison with previous studies.

| Rodrigues et al. [55] | Aziz et al. [19] | Tsinganos and Skodras [26] | Proposed Algorithm |
|-----------------------|-------------------|-----------------------------|---------------------|
| Sensitivity           | 82.4%             | 96%                         | 98.88%              |
| Specificity           | 92.9%             | 96%                         | 99.66%              |

The recorded results of the proposed model show high sensitivity and specificity as compared to previous study proposed by Aziz et al. [19], Tsinganos and Skodras [26], and Rodrigues et al. [55]. These recorded results which was achieved by using smartphones are very closed to those of wearable sensors based system, smartphones are widely used by everyone at any time in addition to many
advantages that make smartphones are the best choice for elderly people as compared to wearable sensors which are intrusive, and restrict the motion of individuals.

5. Conclusion
This study has developed a fall recognition system based on inertial sensors embedded in smartphone; accelerometer and gyroscope, which is more convenient, easier to use, and more practical for elderly than wearable sensors. Falling impact was captured by smartphone inertial sensors data. Then, features were extracted from the data to be ready for classification. We applied two types of machine learning approaches: neural networks (NNs) and support vector machine (SVM) by using WEKA to distinguish falls from other activity daily livings (ADLs). The results indicated that SVM classifier was more accurate than NNs in detecting falls, and the performance of SVM was better than the state of the art classifier performance.

References
[1] Ren L and Peng Y 2019 IEEE Access vol. 7 pp. 77702–77722
[2] Ashfak Habib M, Mohktar M. S, Bahyah Kamaruzzaman S, Seang Lim K, Maw Pin T and Ibrahim F 2014 Sensors (Switzerland) vol. 14, no. 4 pp. 7181–7208
[3] Leone A, Rescio G, and Siciliano P 2018 Int. Conf. Eng. Technol. Innov. Eng. Technol. Innov. Manag. Beyond 2020 New Challenges, New Approaches, ICE/ITMC 2017 - Proc. vol. 2018-Janua pp. 1092–1095
[4] Noury N, Rumeau P, Bourke A. K, ÓLaighin G, and Lundy J. E 2008 Irbm vol. 29, no. 6 pp. 340–349
[5] Otanasap N and Boonbrahm P 2013 Proceedings of Asian Conference on Information Systems pp. 351–355
[6] Bourke A. K et al 2010 J. Biomech vol. 43, no. 15 pp. 3051–3057
[7] Bourke A. K, O’donovan K. J and G. Olaighin 2008 Med. Eng. Phys vol. 30, no. 7 pp. 937–946
[8] Hassan M. M, Uddin M. Z, Mohamed A and Almogren A 2018 Futur. Gener. Comput. Syst vol. 81 pp. 307–313
[9] Attal F, Mohammed S, Dedabishvili M, Chamroukhi F, Oukhellou L and Amirat Y 2015 Sensors (Switzerland), vol. 15, no. 12 pp. 31314–31338
[10] Hegde N, Bries M, Swibas T, Melanson E and E. Sazonov 2018 IEEE J. Biomed. Heal. Informatics vol. 22, no. 4 pp. 979–988
[11] Gjoreski H, Bizjak J, Gjoreski M and M. Gams 2016 25Th Int. Jt., no. Ml pp. 1–7
[12] Arifoglu D and Bouchachia A 2017 Procedia Comput. Sci vol. 110 pp. 86–93
[13] Tao L et al 2015 17th Int. Conf. E-Health Networking Appl. Serv. Heal pp. 644–647
[14] Ivascu T, Cinar K, Dinis A, and Negrui V, 2017 E-Health Bioeng. Conf. EHB 2017 pp. 627–630
[15] Lee K and Kwan M 2018 Comput. Environ. Urban Syst vol. 67 pp. 124–131
[16] Cruciani F, Cleland I, Nugent C, Mccullagh P and Synnes K 2018 pp. 1–20
[17] Nurhanim K, Elamvazuthi I, Izhar L. I and Ganesan T 2017.
[18] Yacchirema D, de Puga J. S, Palau C and Esteve M 2018 Procedia Comput. Sci. vol. 130 pp. 603–610
[19] Aziz O, Musngi M, Park E. J, Mori G and Robinovitch S. N 2017 Medical and Biological Engineering and Computing vol. 55, no. 1 pp. 45–55
[20] Bourke A. K and Lyons G. M 2008 Med. Eng. Phys. vol. 30, no. 1 pp. 84–90
[21] Ozdemir A. T and Barshan B 2014 Sensors (Switzerland) vol. 14, no. 6 pp. 10691–10708
[22] Lu W, Wang C, Stevens M. C, Redmond S. J and N. H. Lovell 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) pp. 2010–2013
[23] Shahzad A and Kim K 2018 IEEE Trans. Ind. Informatics vol. 15, no. 1 pp. 35–44
[24] Madansingh S, Thrasher T. A, Layne C. S, and Lee B 2015 no. Iccas pp. 370–374
[25] Hakim A, Huq M. S, Shanta S, and Ibrahim B. S. K. K 2017 Procedia Comput. Sci vol. 105 pp. 46–51
[26] Tsinganos P and Skodras A 2017 Int. Symp. Image Signal Process. Anal. ISPA pp. 53–58
[27] Chelli A and Patzold M 2019 IEEE Access vol. 7 no. c pp. 38670–38687
[28] Vilarranho T et al 2015 Proc. - 15th IEEE Int. Conf. Comput. Inf. Technol. CIT 2015, 14th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2015, 13th IEEE Int. Conf. Dependable, Auton. Se pp. 1443–1448
[29] Otanasap N 2017 Parallel Distrib. Comput. Appl. Technol. PDCAT Proc pp. 362–365
[30] Hemmatpour M, Karimshouhtari M, Ferrero R, Montrucchio B, Rebaudengo M and Novara C 2017 Proc. - Int. Comput. Softw. Appl. Conf vol. 1 pp. 973–978
[31] Shahzad A, Ko S, Lee S, Lee J. A, and Kim K 2017 IEEE Sens. J vol. 17, no. 20 pp. 6743–6751
[32] Ahn S, Kim J, Koo B and Kim Y 2019 Sensors (Switzerland) vol. 19, no. 4
[33] Fino P. C, Frames C. W, and Lockhart T. E 2015 Sensors (Switzerland) vol. 15, no. 5 pp. 10676–10685
[34] Li M, Xu G, He B, Ma X, and Xie J 2018 IEEE Sens. J. vol. 18, no. 13 pp. 5522–5531
[35] Shen R. K, Yang C. Y, Shen V. R. L and Chen W. C 2017 IEEE Sens. J. vol. 17, no. 6 pp. 1865–1871
[36] Razum D, Seketa G, Vugrin J and Lackovic I 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) pp. 1513–1516
[37] Pham N. P, Dao H. V, Phung H. N, Van Ta H, Nguyen N. H and Hoang T. T 2018 IEEE Seventh International Conference on Communications and Electronics (ICCE) pp. 316–321
[38] Ren L and Shi W 2016 IJSNet vol. 20, no. 3 pp. 163–176
[39] Hemmatpouri M, Ferrero R, Gandino F, Montrucchio B and Rebaudengo 2018 M. J. Healthc. Eng., vol. 2018
[40] Sivaranjani T, DhiviyaLakshmi L, Yogaaravinth R, Srivishnu J, Karthick M. M. S and Praveenkumar A 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI) pp. 2539–2541
[41] Otanasap N 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT) pp. 362–365
[42] Min W, Yao L, Lin Z and Liu L 2018 IET Comput. Vis. vol. 12, no. 8 pp. 1133–1140
[43] Zhen T, Mao L, Wang J and Q. Gao Q 2016 10th International Conference on Sensing Technology (ICST) pp. 1–6
[44] Steffan L, Kaul L, and Asfour T 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids) pp. 171–177
[45] Wu Y, Su Y, Hu Y, Yu N and Feng R 2019 J. Med. Biol. Eng. vol. 39, no. 3 pp. 336–351
[46] Hu X and Qu X 2014 Saf. Sci. vol. 64 pp. 13–21
[47] Poonsri A and Chiracharit W 2018 International Workshop on Advanced Image Technology (IWAIT) pp. 1–4
[48] Su Y 2016 pp. 1727–1731
[49] Khojasteh S, Villar J, Chira C, González V and De La Cal E 2018 Sensors vol. 18, no. 5 p. 1350
[50] Kutckha J, Tchuinkou D, Mandebi J, Nghonda E and Bobda C 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE) pp. 24–29
[51] Ripley B. D 1994 J. R. Stat. Soc. Ser. B vol. 56, no. 3 pp. 409–437
[52] Kaufman L 1998 Advances in Kernel Methods-Support Vector Learning. MIT Press, Cambridge, MA pp. 147–167
[53] Jack L. B and Nandi A. K 2002 Mech. Syst. Signal Process vol. 16, no. 2–3 pp. 373–390
[54] Kalmegh S 2015 Int. J. Innov. Sci. Eng. Technol vol. 2, no. 2 pp. 438–446
[55] Rodrigues F et al. 2017 vol. 10485 pp. 15–26