A Novel Heuristic Fall-Detection Algorithm Based on Double Thresholding, Fuzzy Logic, and Wearable Motion Sensor Data

Billur Barshan and Mustafa Şahin Turan, *Member, IEEE*

***Abstract*—We present a novel heuristic fall-detection algorithm based on combining double thresholding of two simple features with fuzzy logic techniques. We extract the features from the acceleration and gyroscopic data recorded from a waist-worn motion sensor unit. We compare the proposed algorithm to 15 state-of-the-art heuristic fall-detection algorithms in terms of five performance metrics and runtime on a vast benchmarking fall data set that is publicly available. The data set comprises recordings from 2880 short experiments (1600 fall and 1280 non-fall trials) with 16 participants. The proposed algorithm exhibits superior average accuracy (98.45%), sensitivity (98.31%), and *F*-measure (98.59%) performance metrics with a runtime that allows real-time operation. Besides proposing a novel heuristic fall-detection algorithm, this work has comparative value in that it provides a fair comparison on the relative performances of a considerably large number of existing heuristic algorithms with the proposed one, based on the same data set. The results of this research are encouraging in the development of fall-detection systems that can function in the real world for reliable and rapid fall detection.**

***Index Terms*—Accelerometer, double thresholding, fall detection, fall-detection algorithms, fuzzy logic techniques, gyroscope, heuristic (rule-based) algorithms, inertial sensors, magnetometer, motion sensors, wearable sensors, wearables.**

I. Introduction

The UNITED Nations has projected that over the next 30 years, the population of older persons in the world will more than double, reaching 1.5 billion (22%) in 2050 [1]. This means that one out of six people worldwide will be over the age of 64. In many areas of the world, a significant percentage of elderly live alone which compounds the risks this age group already faces. As a result, ambulatory monitoring of the elderly has emerged as an important area of inter-disciplinary research.

Manuscript received 25 October 2022; revised 29 March 2023; accepted 15 May 2023. Date of publication 25 May 2023; date of current version 9 October 2023. *(Corresponding author: Billur Barshan.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Erciyes University Ethics Committee to conduct experiments with human subjects under Application No. 2011/319.

Billur Barshan is with the Department of Electrical and Electronics Engineering, Bilkent University, 06800 Ankara, Turkey (e-mail: billur@ee.bilkent.edu.tr).

Mustafa Şahin Turan is with the Data Science Department, Source.ag, 1066 VH Amsterdam, The Netherlands (e-mail: mustafasahinturan@gmail.com).

Digital Object Identifier 10.1109/JIOT.2023.3280060

Falls are unstable hazardous events where the faller unintentionally ends up on the ground or other lower level [2]. Direct injury from contact with the floor or the extended period of lying on the floor may lead to serious medical conditions, injury or even death when emergency healthcare is not provided rapidly [3]. Statistically, falls are the most common cause of injury-related deaths for people over the age of 79 [4]. Besides older people who face the most serious danger related to falls, disabled people, patients with visual, neurological, balance, gait, and orthopedic disorders, workers, mountaineers, athletes, and children also suffer from falls. Some of the long-term physical, psychological, and social consequences of falls are reduced mobility, independence, and social life. Regardless of the nature of the faller, falling is a serious, costly, and life-threatening public health problem [5], [6], [7], [8]. Developing effective and reliable fall-detection systems is vital in mitigating the severe medical and economical consequences of falls to individuals in the fall risk groups, healthcare systems, and society [9], [10], [11].

Falls often occur unexpectedly in between activities of daily living (ADLs) or during transitions between two body postures (e.g., sitting-to-standing and standing-to-lying down) [12]. Consequently, falls and ADLs are usually considered together. A fall-detection system should not fail to recognize fall events because missing a fall (missed detection) would mean that providing prompt medical attention will not be possible. On the other hand, activities that produce high acceleration (crouching, sitting down rapidly on a bed or sofa, jumping, etc.) can be easily misclassified as falls. Such non-fall activities should not be labeled as falls (false alarm) because false alarms can be disturbing and frustrating to the user. In addition, rapid detection of falls is crucial since the severity of fall-related health risks generally increases with the initial response time. Developing algorithms with lower computational complexity will yield faster response times, allowing (near) real-time operability. In short, fall-detection algorithms need to be sophisticated enough to accurately differentiate between falls and ADLs, while at the same time be sufficiently simple to be computationally efficient and fast.

Besides the essential requirements mentioned above, a functional fall-detection system should have additional qualities to fulfill its crucial role on a daily basis.

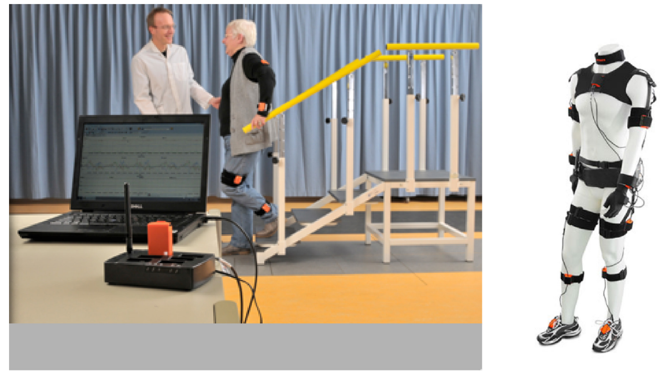
2327-4662 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

1) Since fall-detection systems aim to reduce the direct and indirect costs of falls to the people in the fall risk groups and to the national healthcare system, they need to be affordable, while sustaining high performance.
2) Considering that fall detectors are expected to monitor the user continuously, they need to have low power consumption, especially when energy is limited or costly [13], [14].
3) Fall-detection systems that require the user to wear or carry around bulky equipment inconvenience the user and become obtrusive. As the amount of equipment to be carried increases, the user comfort deteriorates. Therefore, fall detectors need to be compact and ergonomic.
4) Privacy issues are an additional concern for fall-detection systems. People may be reluctant to use systems that intrude into their personal lives through video or audio recordings. Privacy issues may be addressed by selecting a suitable sensing modality or issuing protocols that prohibit monitoring of sensory data outside the scope of the fall-detection algorithm.
5) Finally, a fall-detection system needs to be capable of handling scenarios which may involve multiple people, pets, or objects in the environment that hinder proper data acquisition. Some sensor modalities are inherently robust to these issues in that they record data only from the subject of interest (SoI) whereas others may require additional processing of the data to extract the information that belongs to the SoI, or special configurations and constraints to ensure acquisition of informative data.

One class of systems that meets most of the above criteria is based on wearable sensors (Fig. 1). Through the pervasiveness of a communicating network of interconnected devices and computing intelligence, wearables have become one of the essential elements of the Internet of Things (IoT) ecosystem. Continuous streaming of conveniently accessible signals, acquired from sensors embedded in wearable devices, provide vast amounts of data that carry valuable information about the user state and well being [15]. Proper processing of these data allows developing innovative solutions to challenging problems. Fall detection through the use of wearables has been an active research area for several decades, resulting in considerable amount of academic work and some commercial products on this class of systems [16], [17], [18]. However, existing issues and challenges still need to be addressed to develop effective and robust algorithms that can operate reliably in (near) real time [19].

Regarding the evaluation of the existing fall-detection systems, most of the works reported until now have treated the subject in an uncoordinated and piece-wise fashion. The available literature is rather fragmented and incongruent, without much common basis for comparison among the different studies. Specifically, the literature lacks standards on the choice of participants, fall types, activities, data set collection procedures, and evaluation methods [21], [22]. Most research groups process their custom data sets to evaluate their algorithms [23] to acquire which, a certain modality and configuration of sensors are selected without clear justification. Some studies report only one or two of the five well-known performance metrics which makes comparison even more difficult. Furthermore, most of the earlier work do not present the algorithm runtime or address the real-time implementation of their systems.



(a) (b)

Fig. 1. Examples of wearable systems: (a) Elderly support system. (b) Sensor units with strap set on mannequin [20].

Hauer et al. [24] identify considerable heterogeneity in existing fall definitions and the way falls are documented and analyzed. The article suggests standardizing definitions and the methods of collecting and summarizing falls data. Chaccour et al. [25] provide a comprehensive review on fall prevention and detection systems and proposes a four-level common ground classification of fall-related systems based on their sensor deployment. Noury et al. [26] propose protocols for designing and conducting fall-detection experiments, whereas Abbate et al. [27] set out the most important criteria to consider while designing fall-detection systems. Even though there is a limited number of works that follow the proposed experimental protocols [28], [29], [30], [31], [32] or provide comparison of different algorithms based on the same data set(s) [33], [34], [35], [36], the majority of the existing studies does not present comparable evaluation methods and results. The current situation, then, brings about the need to compare the existing fall-detection techniques using a sizeable and rich data set as common basis. A systematic and unified treatment of the subject is essential.

Kangas et al. [33] compare three threshold-based heuristic algorithms of low complexity. Aziz et al. [34] compare five heuristic and five machine learning (ML)-based fall-detection algorithms employing waist-worn tri-axial accelerometers on their custom data set. Quadros et al. [35] compare several threshold-based and five ML-based techniques for a wrist-worn device encapsulating an accelerometer, a gyroscope, and a magnetometer. Hussain et al. [36] compare three ML algorithms for fall detection based on accelerometer, gyroscope, and accelerometer plus gyroscope data. This article builds up on those articles by proposing a novel heuristic algorithm based on double thresholding and the use of fuzzy logic techniques to detect falls with a single waist-worn motion sensor

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

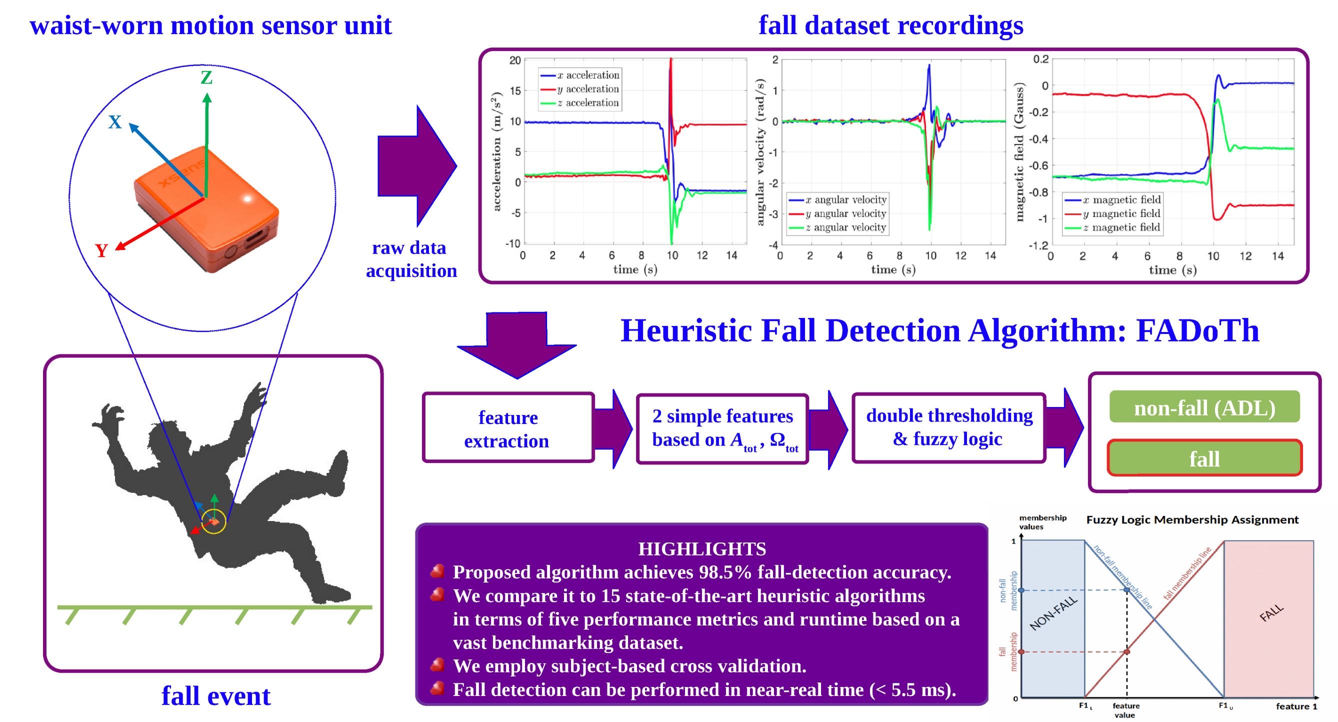


Fig. 2. Overview of the proposed fall-detection scheme.

unit that encapsulates three sensor types. We compare the proposed algorithm to 15 state-of-the-art heuristic algorithms on our publicly available extensive data set that comprises recordings from 2880 short experiments (1600 fall and 1280 non-fall trials) performed by 16 participants. Fig. 2 provides an overview of the proposed scheme. The main contributions of this article are, thus, 1) developing an effective and novel heuristic fall-detection algorithm, employing a combination of double thresholding with fuzzy logic techniques and 2) providing valuable insight on the relative performances of the state-of-the-art heuristic fall-detection algorithms through a fair comparison in terms of five performance metrics and their runtimes based on a benchmarking data set.

The organization of the remainder of this article is as follows: Section II provides the background and a summary of the related work on fall detection. We describe the proposed heuristic algorithm in Section III and give the details of the data set that we have employed in Section IV. Section V, which is on the comparative evaluation of 15 state-of-the-art heuristic algorithms, is divided into five subsections: Section V-A briefly outlines each of the heuristic algorithms. We provide the comparison methodology in Section V-B followed by the section on the results of the comparative evaluation (Section V-C). The next section (Section V-D) is dedicated to the analysis of the algorithm runtime. The final section is on the discussion of the results (Section V-E). Finally, Section VI provides a summary, draws conclusions, and identifies some future research directions.

II. Background and Related Work

With advancements in multiple enabling technologies among which are embedded and context-aware systems [37], sensor technology, wearables, and IoT, the field of fall detection has made significant progress [38], [39]. Numerous academic works have addressed accurate and reliable fall detection using a variety of sensor modalities, each tackling some aspects of the wide scope of the problem. Most of the existing works on fall detection fall under one of two categories [40], [41]: 1) ambient (external) sensor and 2) wearable sensor-based systems. Hybrid systems are a combination of the two. In the first category, smart environments are designed by installing ambient sensors in the user's environment [42]. These could be in the form of cameras, force and pressure sensors, microphones, vibration sensors, infrared proximity sensors, micro-Doppler radars, etc., to capture information-bearing signals about the activity and the state of the user. Smart floor mats, furniture, and equipment can be designed this way. Multiple sensor types can improve the performance by eliminating the drawbacks a single-sensor solution might entail. Non-vision type ambient sensors are advantageous over camera systems because of their low cost, simpler processing requirements, and elimination of privacy concerns. Once the necessary hardware is installed, more than one user (e.g., an elderly couple) in need of such systems can benefit from the smart environment at the same time. The main disadvantage of smart environments is the constraints they bring on the user's mobility because the system can function only in the restricted environment equipped with sensors.

Systems based on wearable technology employ a variety of sensor types (e.g., accelerometers, gyroscopes, magnetometers, and barometers) worn on various body parts to capture the defining characteristics of the movements [43]. With the advancement of the micro electro-mechanical systems (MEMS) technology, these devices have become lighter, smaller, more compact, embeddable, wireless and less costly, while also consuming less and less power.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

Wearable sensors have virtually limitless range; the SoI can be monitored wherever s/he might go indoors and outdoors. This is due to the small, light, and compact nature of wearable devices, which are therefore easily embeddable in clothing, daily accessories, and portable devices. Wearables directly acquire and record the data of only the SoI in 3-D without any occlusion or noise effects caused by other people, pets, or objects. Furthermore, this class of systems does not make video or audio recordings from the users, eliminating any privacy concerns. Since the acquired data typically comprise multiple 1-D time sequences, required processing is simpler and faster. One of the drawbacks of wearable systems is that the user may be reluctant or forgetful in wearing them or may not wear them properly [44], [45]. Besides, depending on their design and bulkiness, wearables can be inconvenient and obtrusive, causing discomfort to the user. Battery maintenance is another problem which can be resolved by using systems that harvest their own energy [46]. Overall, the advantages of wearable fall-detection systems outweigh their disadvantages when compared to smart environments designed for the same purpose.

A considerable number of researchers have investigated the optimal sensor configuration on the body for automatic fall detection. The work reported in [47], based on the same data set as in this article, reveals that the best location to affix motion sensors is the waist area of the subject. Ntanasis et al. [48], also using the same data set, stated that the subject's waist and thigh are the ideal positions. Pannurat et al. [49] disclose that when post-impact activity information is not available, the chest and the side of the waist are the best sensor locations, followed by the head, front of the waist, wrist, ankle, thigh, and upper arm. If such information becomes available, the optimal sensor location is the side of the waist, followed by the head, wrist, front of the waist, thigh, chest, ankle, and upper arm. Dai et al. [50] and Fang et al. [51] consider the chest, waist, and thigh of the participants to affix sensor units and report that they attain better results with the chest-worn sensor units. On the other hand, Bourke et al. [52] state that the torso (trunk) is the ideal position to carry fall-detection devices. Lindemann et al. [53] claim that a single sensor unit on the head is the most useful since rapid head motions are more likely to be caused by falls. Multiple studies consistently agree that the limbs are not suitable body parts to attach motion sensors since they are associated with relatively more chaotic acceleration patterns, generally with higher acceleration values [54]. In summary, majority of the earlier studies have reached the conclusion that the waist or the chest area are the best places to affix wearable motion sensor units for fall detection. We believe that the waist area is a suitable location to place the units due to its proximity to the center of mass of the human body. This, in turn, means that the recorded acceleration and angular velocity signals provide a better reflection of the subject's bodily motion. Thus, although the data set was originally acquired from sensor units at six different positions on the body, in this work, we only process the multi-modal data recorded by the sensor unit worn on the waist.

An issue relevant to fall-detection systems is the segmentation of the acquired sensor sequences. Several studies have conducted event-based segmentation of fall records into natural phases, such as pre-fall, fall, and post-fall, to identify some features of falls [55]. However, most existing works employ static sliding window segmentation. This technique involves taking sequences within a time window of predetermined size and the algorithm processes the windows one at a time. The successive windows may or may not overlap with each other. While using overlapping windows has the disadvantage of higher computational complexity, employing non-overlapping windows is likely to cause some falls to be missed. Using overlapping windows is the more commonly encountered approach in robust fall-detection systems.

We can group the algorithms that are used to make inferences about the captured motion and distinguish falls from ADLs under two categories: 1) heuristic (rule-based) algorithms and 2) machine/deep learning (ML/DL) techniques.

Heuristic algorithms are based on defining hand-crafted rules (e.g., if-then-else type) which often involve pre-set constant threshold levels [56], [57]. Prior to designing this type of algorithm, it is necessary to closely scrutinize the acquired sensory data from non-fall and fall scenarios. Features unique to only fall events need to be identified to develop a set of rules that lead to proper classification. Since the rules are typically not complex ones and training is not required, heuristic algorithms generally operate faster than ML/DL classifiers and are more flexible. Often, thresholding techniques are used, wherein raw sensor data or extracted features are compared with a single or multiple pre-set threshold levels to detect a fall. Heuristic algorithms are favorable especially when, due to the scarcity of fall occurrences, training data may not be sufficient or not be available. Nonetheless, this class of algorithms may need either domain knowledge or data analysis techniques to tune their parameters or set suitable threshold levels for a given user or application [58].

The most commonly used feature in heuristic algorithms is the magnitude of the total acceleration. This is followed by vertical acceleration, body posture angles, total angular velocity (angular rate), and speed of impact which are highly favored features as well. These features are capable of identifying the main characteristics of falls: high impact with the ground, a brief free fall phase prior to the impact, and sudden body posture changes. Several works have considered exploiting the distinctive acceleration profile of falls as a whole [59].

ML/DL techniques have been recently used for fall detection as well. Supervised ML/DL classifiers where the models are trained with labeled data beforehand are more common as opposed to unsupervised ones [60], [61]. Usmani et al. [62] overview the latest research trends in fall prevention and detection systems that use ML techniques. It analyzes the systems on a variety parameters, such as the age distribution of participants, sensor types and configuration for a specific task, data set, and choice of ML algorithms. The work reported in [63] attempts to identify and improve the optimal ML method. Islam et al. [64] review the most effective state-of-the-art DL techniques for fall detection, categorizing them into three. DL models are advantageous in that they can handle feature extraction naturally whereas ML models require defining hand-crafted features. Selecting/optimizing the hyperparameters of ML/DL classifiers is another important implementation issue that can be computationally intensive.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

Because ML/DL classifiers usually involve complex algorithms that require training, they are computationally more demanding compared to heuristic algorithms [65]. Majority of the ML/DL-based works have focused only on improving the accuracy of fall detection without taking into account the computational challenges in real-time applications [66]. Lowering the complexity of ML/DL-based models through data pre-processing, data dimensionality reduction, and normalization techniques is essential to make them operable in the real world.

III. Novel Heuristic Fall-Detection Algorithm

An effective fall-detection algorithm is expected to be both reliable and fast in order to detect falls in real time. Before developing such an algorithm, it is paramount to closely examine the raw signals acquired from a variety of activities. Thus, we have started our study by inspecting the acceleration and angular rate signals recorded during the non-fall and fall actions that took place while creating the vast data set that we have employed in this work. Fig. 3 depicts representative acceleration and angular velocity profiles from one fall and three non-fall type activities. In part a) of the figure, we observe a distinctive pattern that many fall signals exhibit, characterized by an initial drop in the acceleration during the free-fall phase and an acceleration peak that occurs upon impact with the ground. Most non-fall activities are also characterized by distinct acceleration profiles, such as the quasi-periodic patterns with peaks and troughs in the recordings of walking and running activities in parts b) and c) of the figure. Fig. 3(d) illustrates acceleration and angular velocity recordings from a fall-like ADL: posture transition from a standing position to lying on bed. These profiles indicate that some daily activities can also produce acceleration peaks and troughs with high acceleration magnitudes. Even though these exemplary signals of non-fall and fall actions appear to be considerably distinct from one another at first, discriminating between non-fall and fall events is not always straightforward. Non-fall activities that resemble falls, such as jumping, stumbling, and limping with high accelerations, and fall activities, such as syncope and falling out of a bed, can be easily confused with each other. When a vast range of non-fall and fall activities is considered, adequately sophisticated algorithms need to be developed to be able to capture every possible non-fall and fall scenario. Thus, the need for an extensive data set arises.

Having closely scrutinized the characteristics of raw signals produced by non-fall and fall activities and taken their similarities and differences into account, we have developed a heuristic fall-detection algorithm that employs two simple features extracted from the data recorded from a waist-worn motion sensor unit. The two features are based on the magnitude of the total acceleration A_{tot} and the magnitude of the total angular velocity Ω_{tot}, calculated based on the data acquired from the motion sensor unit affixed to the faller's waist. We first define an integer i to index the sampling instances of the data such that $i = 1, \ldots, N$, and N is the length of the data sequence (number of data samples) from a single experimental trial. We calculate the $A_{\text{tot}}[i]$ and $\Omega_{\text{tot}}[i]$ sequences as follows:

$$A_{\text{tot}}[i] = \sqrt{a_x^2[i] + a_y^2[i] + a_z^2[i]} \tag{1}$$

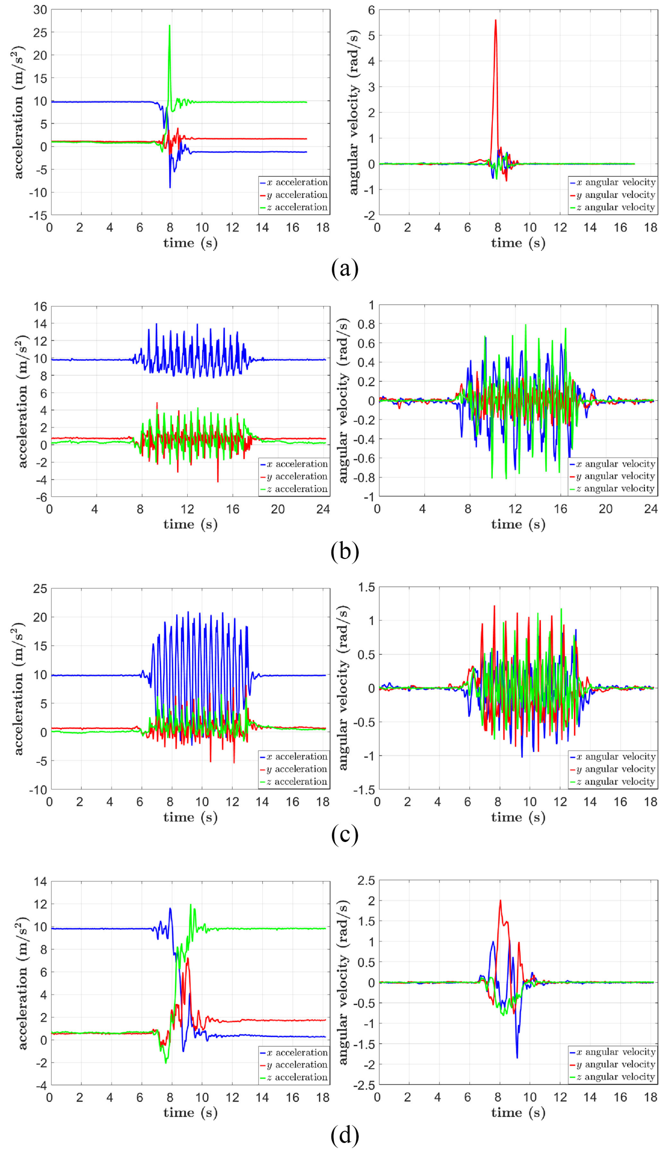


Fig. 3. Acceleration and angular velocity recordings of (a) forward fall, (b) walking (non-fall), (c) running (non-fall), and (d) posture transition from standing to lying on bed (non-fall).

$$\Omega_{\text{tot}}[i] = \sqrt{\omega_x^2[i] + \omega_y^2[i] + \omega_z^2[i]} \quad \text{where} \quad i = 1, \ldots, N.$$

Here, $a_x[i]$, $a_y[i]$, and $a_z[i]$ denote the linear acceleration *along* the x, y, and z axes, and $\omega_x[i]$, $\omega_y[i]$, and $\omega_z[i]$ correspond to the angular velocity *about* the same three axes of the sensor unit, respectively.

We employ two simple features, the maximum value of the magnitude of the total acceleration sequence ($A_{\text{tot-max}}$) and the maximum value of the time sequence resulting from the element-wise multiplication of the $A_{\text{tot}}[i]$ and $\Omega_{\text{tot}}[i]$ sequences, which we have named as $(A\Omega)_{\max}$. We extract these two features as follows:

$$A_{\text{tot-max}} = \max_{1 \le i \le N} \left\{ A_{\text{tot}}[i] \right\} \tag{2}$$

$$(A\Omega)_{\max} = \max_{1 \le i \le N} \left\{ A_{\text{tot}}[i] \cdot \Omega_{\text{tot}}[i] \right\}. \tag{3}$$

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

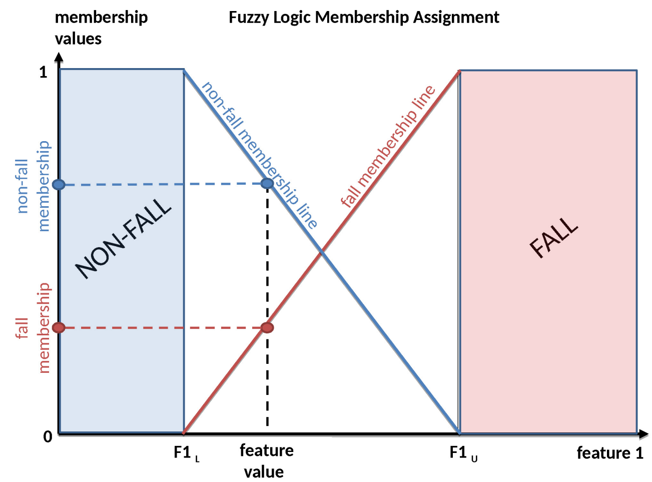


Fig. 4. Assignment of fuzzy logic membership values.

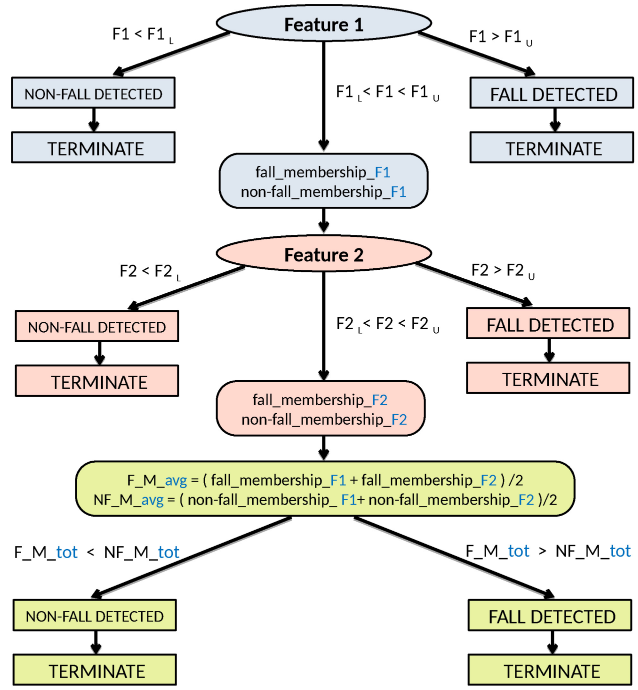


Fig. 5. Flowchart of the proposed fall-detection algorithm.

We selected the two features, $A_{\text{tot-max}}$ and $(A\Omega)_{\max}$ because falls are often associated with high acceleration and angular motion on the faller's trunk. While the first feature captures high accelerations, the second one identifies those data points associated with both high acceleration and high angular rate at the faller's waist. The results of our experiments demonstrate that these two simple features bear sufficient information to exhibit satisfactory performance without costing substantial computational power and time to make the necessary computations.

Once we obtain these two features, the algorithm employs double thresholding where we set a lower and an upper threshold level separately for each of the two features. We select the two threshold levels for the proposed algorithm, as well as those of the algorithms that it is compared to, through the use of cross-validation techniques whose details we provide in Section V-B. Once we set the individual threshold levels for each of the two features, we directly assign data instances (feature vectors) whose first or second feature element is above the corresponding upper threshold level with falls whereas identify those whose first or second feature element is below the corresponding lower threshold level as non-fall activities with complete certainty.

We use fuzzy logic techniques [67] to assign membership values of the data instances to the two classes. Since the first one of the two features ($A_{\text{tot-max}}$) is more capable of differentiating between non-fall and fall activities, the algorithm exploits this feature first, followed by the second. Fig. 4 shows the calculation of the non-fall and fall fuzzy membership values for the first feature (F1: Feature 1). In the figure, $\mathrm{F1_L}$ represents the lower threshold level, whereas $\mathrm{F1_U}$ corresponds to the upper one set for the first feature. Since we consider data instances with $A_{\text{tot-max}}$ value greater than the upper threshold as falls with complete certainty, we assign a fall membership value of one and a non-fall membership value of zero to those instances. Similarly, we assign a fall membership value of zero and non-fall membership value of one to those data instances whose $A_{\text{tot-max}}$ value remains below the lower threshold level. If either of these two extreme cases occur, the decision is made and the algorithm terminates.

For the remaining data instances whose first feature element $A_{\text{tot-max}}$ lies between the lower and upper threshold levels, we create a linear non-fall membership profile that takes the value of one at the lower threshold and the value of zero at the upper threshold, as well as another linear fall membership profile with the reversed borderline values (Fig. 4). Thus, we assign the first feature $A_{\text{tot-max}}$ non-fall and fall membership values between zero and one that add up to one. We then check the second feature $(A\Omega)_{\max}$ and follow the same procedure as we did for the first feature except that the lower and upper threshold values are set differently for the second feature. If the algorithm has not terminated after checking the second feature, in the last phase of the algorithm, we average the membership values assigned to the two features of every non-extreme data instance to obtain a single non-fall and a single fall membership value for that particular data instance. We compare these two values with each other and assign the data instance to the class whose average membership value is larger.

To provide more detail, Fig. 5 displays the flowchart of the proposed algorithm in which we use Feature 1 (and F1) to represent $A_{\text{tot-max}}$ and Feature 2 (and F2) instead of $(A\Omega)_{\max}$ for compactness, while subscripts L and U indicate respective lower and upper threshold levels. The algorithm classifies a data instance into one of the two classes upon reaching a TERMINATE node in Fig. 5. If the first feature ($A_{\text{tot-max}}$) of a data instance is below $\mathrm{F1_L}$ or above $\mathrm{F1_U}$, it classifies the data instance without even using the second feature $(A\Omega)_{\max}$. If, on the other hand, the $A_{\text{tot-max}}$ of a data instance falls between $\mathrm{F1_L}$ and $\mathrm{F1_U}$, the algorithm compares the $(A\Omega)_{\max}$ feature with $\mathrm{F2_L}$ and $\mathrm{F2_U}$. If the second feature is below $\mathrm{F2_L}$ or above $\mathrm{F2_U}$, it assigns the data instance to one of the two classes accordingly, regardless of the fuzzy membership values based on $A_{\text{tot-max}}$. Only if both features of a data instance fall between

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PROPERTIES OF THE SIMULATED FALLS AND DAILY LIVING ACTIVITIES DATA SET

Feature	Description
Areas:	Life, Daily Living
Subject Categories:	Falls, Activities of Daily Living, Wearable Sensing, Motion Sensors, Heuristic Algorithms, Thresholding-Based Algorithms, Fuzzy Logic Techniques
Associated Tasks:	Double Thresholding, Pattern Recognition, Pattern Classification
Dataset Characteristics:	Time-Series Data
Number of Sensor Units:	6 (wireless)
Sensor Types:	accelerometer, gyroscope, magnetometer (each being tri-axial)
Total Number of Sensor Axes:	54 (= 6 sensor units × 3 sensor types × 3 axes per sensor type)
Positions of Sensor Units:	head, chest, waist, right wrist, right thigh, right ankle
Number of Participants:	16
Number of Activities:	36 (= 20 fall + 16 non-fall activities)
Number of Repetitions:	5
Number of Instances:	2880 (= 16 participants × 5 repetitions × 36 movements)
Number of Attributes:	138
Attribute Characteristics:	Integer
Acquired Period:	Winter 2012
Date Donated:	2018-06-06 (UCI Machine Learning Repository [69]) 2022-12-29 (IEEE DataPort [70])
DOI:	https://doi.org/10.21227/eqmh-8m79
Missing Attribute Values:	Rare

the corresponding lower and upper threshold levels, the data instance is classified based on its average membership values. We name the proposed algorithm fuzzy-augmented double thresholding (FADoTh).

We note that not every feature exhibits linear non-fall and fall membership profiles between the lower and upper thresholds. Besides, it may not always be possible to classify extreme values in this way. For example, in some features (e.g., maximum of angular velocity magnitude) both the right and left parts of the extreme data instances may belong to one of the classes and thus the proposed linear membership profiles may not function properly. This may, then, require a different profile of membership values that is possibly of higher order, to better fit the distribution of the non-extreme data. It is, therefore, desirable to examine the distribution of the selected features thoroughly, prior to exploiting them in the proposed algorithm. We have investigated this and have observed that the two features that we employ in this study have the desired characteristics [68].

IV. Data Set and the Experiments

This section describes the publicly available data set [69], [70] that we employ in this study to evaluate the algorithms and the experimental procedure. The data set was initially collected by our research team to assess the performance of state-of-the-art ML classifiers for effective and reliable fall detection [71]. We also employed it in the research published in [47], [48], [72], [73], and [74]. Properties of the data set are summarized in Table I.

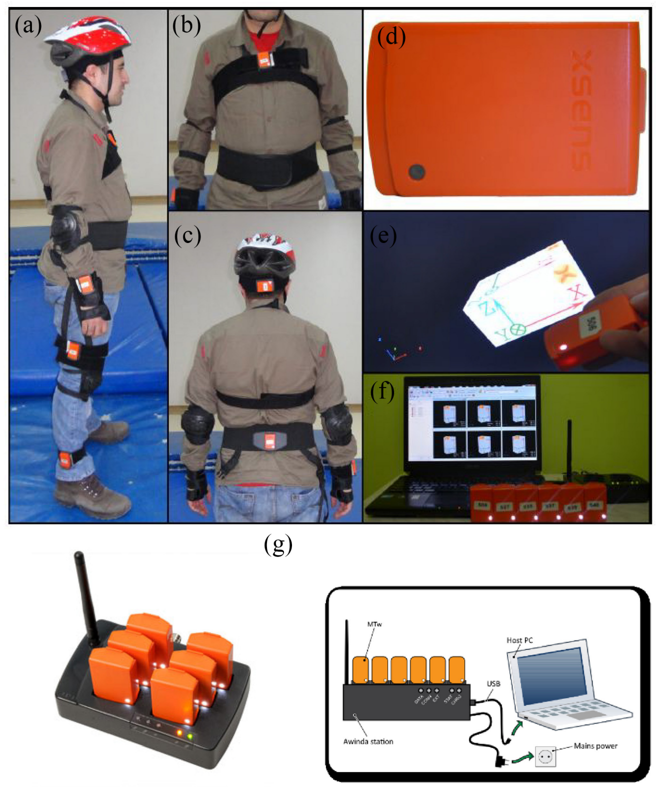


Fig. 6. (a)–(c) Sensor unit configuration on the participant's body, (d) MTw sensor unit, (e) sensor unit x, y, z axes, (f) MT manager software package, (g) Awinda station and its interface to a laptop computer [20], [71].

To conduct the experiments, we first determined the number, type, and the configuration of the sensors to be used, and designed the procedures to acquire data from non-fall and fall events. We employed the Motion Trackers (MTw) Development Kit, manufactured by Xsens Technologies [20] for this purpose. The development kit consists of hardware and software elements. The hardware components comprise six wireless MTw sensor units [Fig. 6(d)–(f)] and the Awinda Station [Fig. 6(g)]. We affixed the sensor units to the head, chest, waist, right wrist, right thigh, and right ankle of each participant using a strap set [Fig. 6(a)–(c)]. Each unit encapsulates three tri-axial sensors (an accelerometer, a gyroscope, and a magnetometer with respective operating ranges of ±120 m/s^2, ±1200°/s, and ±1.5 Gauss) and an atmospheric pressure sensor with a working range of 300–1100 hPa, the last of which we did not use in the experiments. We employed the Awinda Station for wireless acquisition of data from the sensor units as well as for charging the units. As for the software,

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

TABLE II
SHORT DESCRIPTIONS OF THE NON-FALL ACTIVITIES (ADLS) PERFORMED DURING THE EXPERIMENTS

Index	Short Description of Non-Fall Activity
1	walking forward
2	walking backward
3	running
4	squatting and then standing
5	bending at about 90°
6	bending to pick up an object
7	walking with a limp
8	stumbling with recovery
9	ankle sprain
10	coughing/sneezing
11	standing to sitting on a hard surface (chair)
12	standing to sitting on a medium surface (sofa)
13	standing to sitting on air
14	standing to sitting on a soft surface (bed)
15	standing to lying on bed
16	lying on bed to standing

TABLE III
SHORT DESCRIPTIONS OF THE FALL ACTIVITIES PERFORMED DURING THE EXPERIMENTS

Index	Short Description of Fall Activity
17	standing to falling forward to the floor
18	standing to falling forward to the floor with arm protection
19	standing to falling on knees
20	standing to falling on knees and then lying down
21	standing to falling forward with quick recovery
22	standing to falling forward with slow recovery
23	standing to falling forward, ending in right lateral position
24	standing to falling forward, ending in left lateral position
25	standing to falling down on the floor, ending sitting
26	standing to falling backward, ending lying
27	standing to falling backward, ending in right lateral position
28	standing to falling backward, ending in left lateral position
29	standing to falling on the right side, ending lying
30	standing to falling on the right side with recovery
31	standing to falling on the left side, ending lying
32	standing to falling on the left side with recovery
33	from lying, rolling out of bed and falling on the floor
34	standing on a podium to forward fall on the floor
35	syncope — standing to falling vertically
36	syncope fall, slowly slipping off a wall on the side

the MT Manager software package that is part of the MTw Development Kit handles the recording and visualization of data from the sensor units which are analyzed through a graphical interface [Fig. 6(f)]. We calibrated all six sensor units prior to initiating the experiments. It is important to select a suitable sampling rate in detecting non-fall and fall activities to avoid information loss through undersampling and to keep the power consumption of the units at an affordable level. Liu et al. [75] and Santoyo-Ramón et al. [76] state that a sampling rate of 20–22 Hz is sufficient for wearable-based fall-detection systems. Therefore, we set a sampling frequency of 25 Hz in our experiments and transmitted the acquired data via a ZigBee connection to a laptop computer for storage.

Erciyes University Ethics Committee approved the application of our research team to conduct experiments with human subjects (Approval No. 2011/319). Sixteen young and healthy participants performed the scripted activities after we acquired their informed consent in written form. The average value plus/minus one standard deviation for the age, weight, and height of the seven female participants were 21.5 ± 2.5 years, 58.5 ± 11.5 kg, and 169.5 ± 12.5 cm, respectively. Corresponding values for the nine male participants were 24 ± 3 years, 67.5 ± 13.5 kg, and 172 ± 12 cm.

Following the guidelines provided in [27], we selected a broad span of activities so that our experiments would encompass most of the real-world scenarios one would encounter in daily life. We included experimental trials on a wide spectrum of fall types as well as near-fall activities and common ADLs that one can encounter in real-life scenarios. The goal is to be able to construct a genuine representation of real-life situations so that the developed algorithm operates well in the real world and produces realistic outcomes. To acquire the data set, we followed the protocols put forth by Abbate et al. [27] on conducting fall experiments. Each participant performed five trials of each of the 16 non-fall activities (ADLs) and 20 fall activities, which we list in Tables II and III with their brief descriptions. The participants performed the activities on a soft floor mat, with protective equipment on their head, wrists, elbows, and knees to prevent injuries [Fig. 6 (a)–(c)].

We recorded and stored each experimental trial of 10–15 s duration in separate files. As a result, we acquired a vast data set comprising recordings from 2880 short experiments with 1280 non-fall and 1600 fall trials. The fall and non-fall trial recordings that constitute our data set are basically signal amplitude versus time plots similar to those provided in Fig. 3. The signal amplitude may correspond to acceleration, angular velocity (rate), or magnetic field, depending on the sensor type the signal is recorded from (accelerometer, gyroscope, and magnetometer, respectively).

As mentioned in Section II, even though the data set contains data from six motion sensor units worn by the subjects at six different positions on their body, we processed only the data acquired from the waist-worn sensor unit in this work. This results from an effort to render the proposed algorithms more feasible to embed in a hardware system, because not only a system requiring six sensor units would be considerably costly, such a system would also be obtrusive to the user. Furthermore, the larger the number of units, the more difficult it is for the user to put the units on properly [44], [45].

To import the data set, we run three loops for activities, subjects, and experimental trials to span every data instance from separate files. We filter the raw data acquired from the waist-worn motion sensor unit employing three-point median filtering to remove the high frequency noise components. We extract the two features over the whole duration of the data recording from a single experimental trial. We label a data instance as non-fall if the activity index is smaller than or equal to 16 (Table II) or as fall if it is greater than 16 (Table III). We keep a record of the subject index for each data instance to be used in subject-based cross validation, whose details we provide in the next section.

V. COMPARATIVE EVALUATION OF ALGORITHMS

As we stated in Section I, the comparative evaluation of fall-detection algorithms on an extensive benchmarking data set that is acquired according to the comprehensive protocols

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

suggested previously in [26] and [27] is highly valuable, since the current literature is short of studies that provide a fair comparison of different algorithms.

A. State-of-the-Art Heuristic Algorithms

We conducted a comparative evaluation of the proposed algorithm and 15 state-of-the-art heuristic fall-detection algorithms based on the data set described in Section IV. Before proceeding to the comparison methodology, we provide brief descriptions of the selected algorithms below:

Abbate: The work reported in [59] is based on improving the accuracy of a basic fall-detection system by filtering out the false alarms that are caused by three fall-like ADLs. The basic system applies a threshold to A_{tot} with the detection of a static interval after the impact during which A_{tot} does not exceed the threshold. Following this basic system, the researchers use the average acceleration magnitude variation (AAMV) feature to distinguish sitting/lying on a chair, sofa, or bed that causes false alarms. In addition, they employ the free fall interval (FFI) and free fall average acceleration magnitude (FFAAM) to detect the jumping activity and eliminate the false alarms that it may cause. Because the authors of this work custom created these three features and their descriptions are lengthy, the interested reader should resort to [59] for additional details.

Anania: Anania et al.'s algorithm [77] thresholds the total angular rate and the angular change of the trunk. If both quantities exceed the corresponding pre-set thresholds, it checks to see if these two peaks occur sufficiently close to each other in time before raising a fall alarm.

Baek: This algorithm proposed by Baek et al. [78] first determines the posture of the subject by thresholding the roll, pitch, and yaw angles (Euler angles) obtained by exploiting trigonometric relationships between the low pass filtered acceleration sequences in the x, y, and z directions and these angles. If the lying posture is detected, the researchers apply individual thresholding to A_{tot} and Ω_{tot} to detect a fall.

Bourke-1: The first algorithm developed by Bourke et al. [52] filters the raw data with a second-order digital Butterworth low pass filter (LPF) that has a cut-off frequency of 250 Hz. Then, it applies thresholding to A_{tot}.

Bourke-2: The second algorithm by Bourke et al. [79] applies a single threshold to vertical linear velocity. This quantity is obtained by first subtracting the gravitational acceleration from A_{tot} and then numerically integrating the result. The researchers filter the raw data with a second-order digital Butterworth LPF that has a cut-off frequency of 15 Hz. They remove the integration drift by band-pass filtering (BPF) the velocity profiles using a second-order digital Butterworth BPF with cut-off frequencies of 0.15 and 15 Hz.

Bourke-3: Bourke et al.'s third algorithm [80] employs thresholding on Ω_{tot}, $\dot{\Omega}_{\text{tot}}$, and angular change sequences about the roll and pitch axes. The researchers obtain the angular acceleration sequence ($\dot{\Omega}_{\text{tot}}$) by calculating the first difference sequence of the angular rate sequence (Ω_{tot}) and acquire the angular change by numerical integration of the angular rate sequence in time windows of 1.7 s. They filter the raw data using a second-order digital Butterworth LPF with a cut-off frequency of 100 Hz.

Chen: The algorithm suggested by Chen et al. [81] relies on detecting the impact with the floor by thresholding A_{tot} first. If this quantity exceeds the threshold, it calculates the change in the orientation of the faller's body before and after the impact. If the change is significant, the algorithm raises a fall alarm.

Jacob: Jacob et al. [82] put forth an interesting algorithm that employs seven different flags that are raised when the seven corresponding features exceed the individually pre-set thresholds. The selected features are the A_{tot}, Ω_{tot}, $\dot{\Omega}_{\text{tot}}$, the angular changes about the x, y, and z axes, and the total angular change. The algorithm calculates the angular acceleration by numerical differentiation and obtains the angular changes by numerical integration of the gyroscopic angular rate data. It detects a fall if four or more flags out of seven are raised.

Jantaraprim: The algorithm proposed by Jantaraprim et al. [83] applies double thresholding to the A_{tot} sequence. If the A_{tot} value falls below the lower threshold (free fall phase) and then exceeds the upper one (impact), the algorithm inspects the time indices of these two elements of A_{tot} to see if they are at sufficient proximity to each other in time. If so, the algorithm detects a fall. The researchers filter the raw data with a second-order digital Butterworth LPF that has a cut-off frequency of 20 Hz.

Kangas-1, Kangas-2, and Kangas-3: Kangas et al. [33] develop three different fall-detection algorithms with low complexity. The first algorithm, Kangas-1, is based on detecting the impact moment of a fall and the posture of the fallen person after the impact. The algorithm detects the impact by thresholding the dynamic total acceleration (A_{D}) and determines the posture by exploiting the low pass filtered vertical acceleration ($A_{\text{vert-LPF}}$).

The second algorithm, Kangas-2, detects the beginning of the fall, impact moment, and the posture of the faller afterward. It identifies the beginning of the fall by thresholding A_{tot}, while detecting the impact moment and posture in the same way as in Kangas-1.

The third algorithm, Kangas-3, is based on detecting the start of the fall, fall velocity, impact, and posture afterward. The difference from Kangas-2 is that, it obtains the fall velocity by numerically integrating the A_{tot} sequence from the beginning of the fall until the time of impact.

All three algorithms employ three-point median filtering on the raw data prior to extracting the necessary features. To obtain A_{D}, the researchers filter acceleration data with a second-order digital Butterworth high pass filter with a cut-off frequency of 0.25 Hz and calculate the sum vector magnitude of the resulting acceleration. They use a second-order digital Butterworth LPF with a cut-off frequency of 0.25 Hz for posture calculation.

Lindemann: Lindemann et al. [53] apply thresholding to the horizontal component of the total acceleration ($A_{\text{tot-hor}}$), A_{tot}, and the total velocity prior to the impact. The researchers obtain the quantity $A_{\text{tot-hor}}$ by simply exploiting the configuration of the sensor unit, whereas calculate the total velocity by backward integration of A_{tot} over a 1.5 s window from the

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

peak of A_{tot} backward. They filter the raw data with a LPF having a cut-off frequency of 80 Hz.

Sorvala: The algorithm proposed by Sorvala et al. [84] applies thresholding on A_{tot} and Ω_{tot} to detect impact. Upon the detection of impact, it compares the angular change of the body to a pre-set threshold level and raises a fall alarm if it is above that threshold. The authors filter raw acceleration data with a three-point median filter while they use a second-order digital Butterworth LPF with a cut-off frequency of 0.25 Hz to acquire angular change based on acceleration sequences.

Wang: The algorithm proposed by Wang et al. [85] also relies on thresholding. The algorithm raises a fall alarm if A_{tot} exceeds a pre-set threshold. If not, it compares the horizontal component of the acceleration ($A_{\text{tot-hor}}$) to another threshold. If this quantity is above the threshold, the algorithm calculates the linear velocity by backward integration and raises a fall alarm if the velocity is greater than a pre-set threshold level.

B. Comparison Methodology

In our implementation of these algorithms, we did not apply any additional filtering to the raw data other than the filters that are specified in the algorithm descriptions; however, we trimmed off 10 data samples from the beginning and the end of every trial recording and discarded them because these samples bear corrupted data caused by on–off switching of the sensor units.

Nearly every heuristic algorithm involves the use of one or more threshold levels and the selection of these levels should be done properly. Although the studies we consider here state the optimal parameter values for their algorithms, these parameters are selected based on their custom data set and do not necessarily give the best results on a different data set. To be able to make a fair and realistic comparison between the performances of the fall-detection algorithms that we have included in our study, we have employed a subject-based multi-fold cross-validation scheme [86] to identify the optimal parameter set for each algorithm. We prefer this over randomly partitioned (i.e., not according to subject) multi-fold cross validation because our aim is to make a fair evaluation of the algorithms where the data recorded from the test subjects are not employed in determining the parameters. If any algorithm we present here were to be embedded in a fall-detection device for real-world usage, it is extremely likely that the users of such a device will not have contributed to the data set. Therefore, by not including the test subjects' data in the training set, our aim is to prevent optimistic results that may be caused by the correlation between the training and test data from the same subject. This choice, in turn, generally yields lower performance metrics with more variation. As a result, the standard deviation between the performance metrics of different validation folds increases.

We partitioned the fall data set into eightfold in a subject-based manner, employing the record of the subject index for each data instance: seven data partitions from one female and one male subject each and one partition of data from two male subjects. This way, we created eightfold, each comprising the data from a pair of subjects. In a loop, we keep each of one these eightfold as the test data while combining the remaining seven to be employed as the training data. For each iteration of this loop, we conduct a grid search to obtain the set of parameters of the algorithm that gives the highest classification accuracy on the training set. During the grid search, we consider sufficiently wide intervals for the parameters to avoid obtaining resulting parameter values on the borders of the intervals; besides, we update the intervals of the parameter sweep should we encounter such a situation. After we obtain the parameter set that results in the highest classification accuracy on the training set for a given fold, we use that parameter set to evaluate the algorithm on the test set for that particular fold.

		classified		
		fall	**non-fall**	**total:**
true	**fall**	TP	FN	P
	non-fall	FP	TN	N
	total:	P′	N′	

Fig. 7. 2×2 confusion matrix.

Fall detection involves a binary decision process on whether a fall has occurred or not. In testing a given fall-detection algorithm, one may encounter one of four possible cases.

1) *True Positive (TP):* A fall occurs and the algorithm detects it;
2) *False Positive (FP):* A fall does not occur but the algorithm detects a fall;
3) *True Negative (TN):* A fall does not occur and the algorithm does not detect a fall;
4) *False Negative (FN):* A fall occurs but the algorithm does not detect it.

In radar terminology, FP and FN correspond to *false alarm* and *missed detection*, respectively. Once we identify the number of times each of these four cases occur, we can build a confusion matrix as in Fig. 7. In the figure, P represents the total number of true positives and N represents the total number of true negatives. The variables P′ and N′ denote the total numbers of estimated positives and negatives, respectively. At the end of the training and test procedures described above, we store the corresponding confusion matrices.

Using the recorded confusion matrices and the definitions given in 1)–4) above, we calculate the accuracy, precision, sensitivity (recall), specificity, and *F*-measure performance metrics of the algorithms according to the following equations:

$$\text{Accuracy} = \frac{1}{2} \cdot \left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right) \tag{4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RESULTS OF THE COMPARATIVE STUDY ON HEURISTIC ALGORITHMS

Algorithm	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F-measure (%)
Abbate [59]	96.52±2.07	98.13±2.49	95.37±2.59	97.66±3.15	96.70±1.84
Anania [77]	90.19±3.00	96.54±4.80	84.44±2.64	95.94±5.84	90.00±2.48
Baek [78]	96.02±1.38	97.90±1.41	94.63±3.03	97.42±1.75	96.20±1.43
Bourke-1 [52]	96.23±1.28	97.61±1.51	95.44±2.80	97.03±1.91	96.48±1.30
Bourke-2 [79]	94.95±1.16	95.05±2.44	96.31±3.18	93.59±3.23	95.61±1.11
Bourke-3 [80]	89.94±2.04	96.85±2.66	83.31±2.19	96.56±2.95	89.55±1.91
Chen [81]	95.20±1.86	98.00±2.35	92.81±2.07	97.58±2.90	95.32±1.65
Jacob [82]	90.28±2.17	94.70±3.34	86.81±3.42	93.75±4.25	90.51±1.97
Jantaraprim [83]	91.97±1.24	96.83±2.30	87.63±4.38	96.25±2.95	91.93±1.68
Kangas-1 [33]	97.80±1.12	98.41±1.97	97.63±1.33	97.97±2.56	98.00±0.88
Kangas-2 [33]	98.05±0.83	**98.94±1.16**	97.44±1.45	**98.67±1.47**	98.17±0.76
Kangas-3 [33]	97.13±1.45	96.92±2.20	98.25±2.36	96.02±2.91	97.55±1.26
Lindemann [53]	96.55±2.07	98.13±2.49	95.44±2.53	97.66±3.15	96.74±1.83
Sorvala [84]	95.02±1.29	96.45±2.52	94.56±4.44	95.47±3.34	95.39±1.50
Wang [85]	97.03±0.61	97.53±1.44	97.19±1.58	96.88±1.86	97.34±0.54
proposed FADoTh	**98.45±1.21**	98.89±1.51	**98.31±1.03**	98.59±1.91	**98.59±1.03**

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

$$F\text{-measure} = 2 \cdot \frac{\text{Precision} \cdot \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}}. \tag{8}$$

We note that we have employed the average of the class-based accuracies in (4) instead of the conventional definition of accuracy, given by $(\text{TP}+\text{TN})/(\text{TP}+\text{FN}+\text{TN}+\text{FP})$, since the two classes are not equal in size. It is clear that sensitivity and specificity metrics are reciprocally related. For instance, in a heuristic algorithm based on simple thresholding, as we decrease the threshold level, the rate of FN decreases and the sensitivity of the algorithm increases. Consequently, FP rate increases and specificity decreases. On the other hand, as we raise the threshold level, the opposite takes place: sensitivity decreases and specificity increases. Since the precision and sensitivity measures are inversely related, it is worthwhile to evaluate the F-measure metric that combines these two metrics into one to attain a compounded performance measure. We obtain the F-measure metric in (8) by multiplying the harmonic mean of precision and sensitivity metrics by two.

C. Comparison Results

We present the performance metrics for each algorithm in average value plus/minus one standard deviation format in Table IV. It is evident from the table that among all the algorithms considered, the proposed algorithm (FADoTh) exhibits the highest average accuracy, sensitivity, and F-measure performance metrics. Only the Kangas-2 algorithm displays slightly higher average precision and specificity values; however, this algorithm then demonstrates lower performance in the other three performance metrics. All three algorithms proposed by Kangas et al. [33] provide sensitivity metrics that are lower but within one standard deviation of FADoTh sensitivity. Among these, the only algorithm that produces an average sensitivity value that is comparable with the proposed algorithm is Kangas-3. However, the remaining four performance metrics of this algorithm are not within one standard deviation of those of the FADoTh algorithm. In fact, only the first two algorithms by Kangas (Kangas-1 and Kangas-2) are within one standard deviation of the accuracy and the F-measure metrics of the proposed algorithm. These two algorithms yield comparable results to the proposed one, although not better.

Overall, the proposed FADoTh algorithm, employing double thresholding on only two simple features and fuzzy logic techniques, proves superior to the 15 state-of-the-art heuristic fall-detection algorithms on a vast data set comprising a total of 2880 experimental trials when we employ subject-based cross validation for parameter selection and performance evaluation. We believe that the proposed algorithm can demonstrate satisfactory performance on other data sets as it exploits two fundamental characteristics of falls: data instances of high acceleration and those with both high acceleration and high angular rate.

D. Runtime Analysis

We ran all 16 algorithms on MATLAB version R2015a installed on a computer with Intel Core i5-3230M CPU running at 2.60 GHz, 4 GB RAM, and Windows 7 Home Premium 64-bit operating system while no other external application or program is being executed.

To measure algorithm runtimes, we ran each algorithm ten times on the data acquired from the waist sensor unit without using cross validation. In Table V, we provide the average runtimes for the classification of a single data instance in average value plus/minus one standard deviation format.

Even though our proposed algorithm ranks as the seventh fastest algorithm out of all 16, it operates faster than the only two algorithms (Kangas-1 and Kangas-2) with comparable performance metrics. Recall that the average accuracy and F-measure values of these two algorithms were within one standard deviation of those of the proposed algorithm.

E. Discussion

Eight of the 15 state-of-the-art algorithms produced accuracy values lower than 96% based on the comparison on the same data set. This outcome is similar for the other performance metrics although all of the algorithms we compare in this work are reported to exhibit high fall-detection

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

TABLE V
RUNTIMES OF THE COMPARED HEURISTIC ALGORITHMS

Algorithm	Runtime (ms)
Abbate [59]	**1.242±0.033**
Anania [77]	5.949±0.044
Baek [78]	6.064±0.035
Bourke-1 [52]	5.269±0.025
Bourke-2 [79]	10.731±0.074
Bourke-3 [80]	5.369±0.017
Chen [81]	5.974±0.029
Jacob [82]	5.533±0.014
Jantaraprim [83]	5.357±0.026
Kangas-1 [33]	5.955±0.034
Kangas-2 [33]	6.105±0.022
Kangas-3 [33]	6.420±0.025
Lindemann [53]	1.307±0.010
Sorvala [84]	6.111±0.024
Wang [85]	5.344±0.009
proposed FADoTh	5.461±0.017

performance with the original data sets that they are developed and tested on. The difference is most likely caused by the relatively large size of the data set employed in this work. To our knowledge, none of the data sets that are used in the publications that these 15 state-of-the-art algorithms are originally proposed in are comparable in size to the data set that we have used here. Not only the mere size of the data set we employed here is larger than those in the included publications, the variety of non-fall and fall activities it includes spans far more of real-life scenarios as well. However, the fall-detection algorithms developed by Kangas et al. [33] perform considerably well on our sizeable data set, despite that they were originally demonstrated over a limited amount of data acquired from three subjects performing nine fall types. Their success lies in their capability to capture the most general defining characteristics of falls.

The 16 ADLs included in our data set are a subset of real-world daily activities that can easily be mistaken as falls. Falls/ADLs that take place in a laboratory-like setting with protective safety gear (e.g., helmets, knee supports, and mats) and those that occur out in the real world may differ. In particular, intentional (voluntary or simulated) falls performed with such protective equipment may not reflect the sudden and chaotic nature of real-world falls. Second, we must note that we acquired the data set mostly from healthy and young participants, performing intentional and scripted fall scenarios in a structured environment. These factors may hinder the performance of fall-detection systems in the real world. Fall risk groups include the elderly and patients with movement disorders who may have impairments and abnormalities in their gait cycles that may not necessarily match with the subjects who took part in the experimental trials. However, fall data from the real world involving individuals in the fall risk groups are quite scarce and difficult to acquire because of their fragility as well as the long waiting times. There exists only a limited number of works that consider real-world falls [87], [88] and involve elderly participants [52], [53], [83], [89], [90], [91] or patients with certain movement disorders or conditions (e.g., Parkinson's disease, multiple sclerosis, and stroke). For example, Bourke et al. [91] include unscripted ADL data from older people to evaluate the false alarm rate of their algorithm in a real-world setting. Mosquera-Lopez et al. [88] exploit real-world data from patients with multiple sclerosis, monitored over a period of eight weeks during free-living conditions, to develop their ML-based fall-detection algorithm.

Although we would have preferred to process real fall data from individuals in the actual fall risk group, we have made the choices for the non-fall/fall activity types and the current subject profiles with the goal of gathering a rich and extensive data set [69]. These choices should not impede the results obtained in this article because Kangas et al. compare real fall data acquired from the elderly with those of simulated falls in [92] and reach the conclusion that the former bears similar characteristics to those of intentional falls, despite that some parameters may differ between the two. We have compared the average and peak acceleration values of the intentional falls that we have recorded, with those in [92], where a limited number of natural falls by the elderly are available. We presented this comparison in our earlier study [71] where we observed that for a given fall type, features of the signals recorded from involuntary and voluntary falls show a good resemblance. Thus, we have demonstrated that the experimental records acquired in our work are coherent with the involuntary falls recorded in an independently conducted study [92]. Consequently, we would expect the proposed FADoTh algorithm to perform well in real-world scenarios if embedded in fall-detection hardware.

We finally provide examples and performance metrics of some of the relevant works that we could not include in our comparative study to keep the number of algorithms at a reasonable level. A threshold-based pre-impact fall-detection algorithm is developed in [93] for wearable airbags that minimize the impact of falls on the faller's body. The algorithm achieves 92.4% accuracy, 90.5% specificity, and 96.1% sensitivity with the publicly available SisFall data set [94]. De Sousa et al. [95] also use the same data set to test their threshold-based algorithm and reported 97.7% specificity and 92.6% sensitivity values. In the comparative study reported in [35] that uses a wrist-worn device with motion sensors, threshold-based methods resulted in a maximum accuracy of 89.1% with 82.3% specificity and 95.8% sensitivity. When thresholding was supplemented with Madgwick's decomposition to estimate the spatial orientation of the body, the respective performance metrics improved to 91.1%, 95.8%, and 86.5%. Qian et al. [96] present a wearable fall-detection system based on multi-level thresholding. The algorithm combines MEMS with narrowband IoT (NB-IoT) and achieves 94.88% accuracy, 94.5% specificity, and 95.25% sensitivity. Patel et al. [97] describe the VitaFALL device which employs multiple thresholds on the tri-axial acceleration data together with vital signs and reports that accuracy levels up to 94% are achieved. Another recent study that uses threshold levels on wearable accelerometer data is by Amir et al. [98] which reports 83.0% accuracy, 69.4% specificity, and 97.4% sensitivity. Lee and Tseng [99] identify suitable threshold levels for fall and non-fall events and propose an enhanced threshold-based fall-detection methodology which, in addition, classifies falls into four basic directions (forward, backward,

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

right lateral, and left lateral). Accuracy and fall-detection rates of 99.38% and 96% are obtained. In [100], a dynamic threshold model is employed for pre-impact fall detection where 97.40% accuracy, 95.31% specificity, and 99.48% sensitivity figures are reported. Razum et al. [101] elaborate on optimal threshold selection for fall-detection algorithms that process multiple features. Qu et al. [102] propose a fall-detection algorithm that combines quadratic threshold decision with human posture and support vector machines (SVMs) where a smart bracelet recognizes falls at a rate of 92.2%. The multi-threshold-based algorithm evaluated in [103] yields 86% accuracy, 89.4% specificity, and 95.7% sensitivity. The results are improved by supplementing thresholding with an extreme learning machine (EML). The study reported in [104] employs a two-step algorithm based on thresholding and multiple kernel learning SVMs that process accelerometer signals. Respective accuracy figures of 97.8% and 91.7%, specificity of 95.2% and 88.0%, and sensitivity values of 99.5% and 95.8% are reported when the accelerometer is worn on the user's waist and thigh. Xu et al. [105], [106] propose another two-stage fall-detection algorithm that combines thresholding with DL. The algorithm first pre-screens an event using a threshold-based method. If a fall event is suspected, a convolutional neural network (CNN) processes the data. The fusion algorithm exhibits 97.02% accuracy, 96.64% specificity, and 97.83% sensitivity. Zhang et al. [107] also present a two-step algorithm that combines thresholding with fuzzy logic techniques to process the motion data acquired from a wrist-worn fall detector. A small data set with four fall types, five types of bodily activities, and four types of hand activities is created with the participation of only two subjects. The authors report 98.36% accuracy, 100% specificity, and 95.10% sensitivity.

Although the authors are well aware of the fact that the performance metrics reported in different studies are not directly comparable for the reasons explained in the introductory section, we still note that all five performance metrics of the proposed FADoTh algorithm are above 98.3%, exceeding those of most of the existing threshold-based heuristic algorithms reviewed here.

VI. Conclusion

We proposed a novel heuristic fall-detection algorithm based on processing data acquired from a waist-worn motion sensor unit. The algorithm applies double thresholding on two simple features to first identify and then classify extreme data. We calculate fuzzy membership values of the non-extreme data instances to the non-fall or fall classes by using fuzzy logic techniques and assign them to the two classes accordingly. Besides the proposed one, we have implemented 15 existing heuristic fall-detection algorithms and evaluated their performances based on a sizeable benchmarking data set. We employ subject-based cross validation for the parameter selection and evaluation of all algorithms, which is more appropriate than alternative cross-validation techniques. The proposed algorithm is superior to all of the heuristic algorithms in terms of the accuracy, sensitivity, and F-measure performance metrics. Only one of the considered algorithms (Kangas-2) yields slightly higher specificity and precision metrics than the proposed one. Analysis of the algorithm runtimes reveals that the runtime of the proposed algorithm is shorter than the two alternative algorithms that yield comparable performance measures. Besides proposing a novel heuristic fall-detection algorithm, this work has comparative value in that it compares the relative performances of a considerably large number (15) of existing heuristic algorithms with the proposed one in terms of five performance metrics and their runtimes. We aimed to make this comparison a fair one by implementing each of the 15 algorithms from scratch and identifying their optimal parameter values using subject-based multi-fold cross-validation over the same data set.

Future work may consider a similar comparative evaluation among the state-of-the-art ML/DL algorithms. Developing fall-detection algorithms that are robust to misalignment of the sensor unit(s) is a potential research direction that can be pursued as well. With the pervasiveness of wearables and advancements in IoT, there is an ever growing need for systems that can harvest their own energy. Sensor units that can harvest the required energy from bodily motion can be developed to tackle the battery maintenance issue in wearables. In addition, the algorithms can be embedded in hardware for real-world usage. Such an implementation would also enable the evaluation of the algorithms based on real-world data acquired from real non-fall and fall activities of those in the actual fall risk group.

References

[1] United Nations, Department of Economic and Social Affairs, Population Division. "World population ageing 2019 (ST/ESA/SER.A/444)." 2020. [Online]. Available: https://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2019-Report.pdf

[2] "Falls." World Health Organization. Apr. 2021. Accessed: Jun. 27, 2023. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs344/en/

[3] D. Wild, U. S. L. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?" *Brit. Med. J.*, vol. 282, no. 6260, pp. 266–268, Jan. 1981. [Online]. Available: https://doi.org/10.1136/bmj.282.6260.266

[4] C. Griffiths, C. Rooney, and A. Brock, "Leading causes of death in England and Wales—How should we group causes?" *Health Stat. Quart.*, vol. 28, no. 9, pp. 6–17, 2005, PMID: 16315552.

[5] A. Ishtiaq, S. Saeed, M. U. Khan, A. Samer, M. Shabbir, and W. Ahmad, "Fall detection, wearable sensors & artificial intelligence: A short review," *J. Adv. Res. Elect. Eng.*, vol. 6, no. 2, pp. 122–129, Oct. 2022. [Online]. Available: https://doi.org/10.12962/jaree.v6i2.323

[6] S. Nooruddin, M. M. Islam, F. A. Sharna, H. Alhetari, and M. N. Kabir, "Sensor-based fall detection systems: A review," *J. Ambient Intell. Humanized Comput.*, vol. 13, no. 5, pp. 2735–2751, May 2022. [Online]. Available: https://doi.org/10.1007/s12652-021-03248-z

[7] L. Ren and Y. Peng, "Research of fall detection and fall prevention technologies: A systematic review," *IEEE Access*, vol. 7, pp. 77702–77722, Jun. 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2922708

[8] P. Vallabh and R. Malekian, "Fall detection monitoring systems: A comprehensive review," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 7, pp. 1809–1833, Nov. 2018. [Online]. Available: https://doi.org/10.1007/s12652-017-0592-3

[9] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: A survey," *Appl. Sci.*, vol. 8, no. 3, Mar. 2018, Art. no. 418. [Online]. Available: https://doi.org/10.3390/app8030418

[10] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, Jul. 2014. [Online]. Available: https://doi.org/10.3390/s140712900

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

[11] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, Jan. 2013. [Online]. Available: https://doi.org/10.1016/j.neucom.2011.09.037

[12] B. Najafi, K. Aminian, F. Loew, Y. Blanc, and P. A. Robert, "Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its application in fall risk evaluation in the elderly," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 843–851, Aug. 2002. [Online]. Available: https://doi.org/10.1109/tbme.2002.800763

[13] S. B. Kwon, J.-H. Park, C. Kwon, H. J. Kong, J. Y. Hwang, and H. C. Kim, "An energy-efficient algorithm for classification of fall types using a wearable sensor," *IEEE Access*, vol. 7, pp. 31321–31329, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2902718

[14] T. N. Gia et al., "Energy efficient wearable sensor node for IoT-based fall detection systems," *Microprocess. Microsyst.*, vol. 56, pp. 34–46, Feb. 2018. [Online]. Available: https://doi.org/10.1016/j.micpro.2017.10.014

[15] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using IoT and Big Data," *Procedia Comput. Sci.*, vol. 130, pp. 603–610, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.04.110

[16] D. Yacchirema, J. S. de Puga, C. Palau, and M. Esteve, "Fall detection system for elderly people using IoT and ensemble machine learning algorithm," *Pers. Ubiquitous Comput.*, vol. 23, pp. 801–817, Jan. 2019. [Online]. Available: https://doi.org/10.1007/s00779-018-01196-8

[17] S. Nooruddin, M. M. Islam, and F. A. Sharna, "An IoT based device-type invariant fall detection system," *Internet Things*, vol. 9, Mar. 2020, Art. no. 100130. [Online]. Available: https://doi.org/10.1016/j.iot.2019.100130

[18] S. Moulik and S. Majumdar, "FallSense: An automatic fall detection and alarm generation system in IoT-enabled environment," *IEEE Sensors J.*, vol. 19, no. 19, pp. 8452–8459, Oct. 2019. [Online]. Available: https://doi.org/10.1109/jsen.2018.2880739

[19] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. Online*, vol. 12, Jul. 2013, Art. no. 66. [Online]. Available: https://doi.org/10.1186/1475-925X-12-66

[20] "MTw Awinda user manual: MTw hardware, MT manager, Awinda protocol," Xsens Technol. B.V., Enschede, The Netherlands, Document MW0502P, Revision L, May 2018. [Online]. Available: https://www.xsens.com/hubfs/Downloads/Manuals/MTw_Awinda_User_Manual.pdf

[21] L. Schwickert et al. for the FARSEEING Consortium and the FARSEEING Meta Database Consensus Group, "Fall detection with body-worn sensors: A systematic review," *Zeitschrift Für Gerontologie und Geriatrie*, vol. 46, no. 8, pp. 706–719, Nov. 2013. [Online]. Available: https://doi.org/10.1007/s00391-013-0559-8

[22] N. Noury, P. Rumeau, A. K. Bourke, G. ÓLaighin, and J. E. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340–349, Dec. 2008. [Online]. Available: https://doi.org/10.1016/j.irbm.2008.08.002

[23] E. Casilari, J.-A. Santoyo-Ramón, and J.-M. Cano-García, "Analysis of public datasets for wearable fall detection systems," *Sensors*, vol. 17, no. 7, Jun. 2017, Art. no. 1513. [Online]. Available: https://doi.org/10.3390/s17071513

[24] K. Hauer, S. E. Lamb, E. C. Jorstad, C. Todd, and C. Becker, "Systematic review of definitions and methods of measuring falls in randomised controlled fall prevention trials," *Age Ageing*, vol. 35, no. 1, pp. 5–10, Jan. 2006. [Online]. Available: https://doi.org/10.1093/ageing/afi218

[25] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andrès, "From fall detection to fall prevention: A generic classification of fall-related systems," *IEEE Sensors J.*, vol. 17, no. 3, pp. 812–822, Feb. 2017. [Online]. Available: https://doi.org/10.1109/jsen.2016.2628099

[26] N. Noury et al., "Fall detection—Principles and methods," in *Proc. 29th Annu. Int. IEEE EMBS Conf.*, Lyon, France, Aug. 2007, pp. 1663–1666. [Online]. Available: https://doi.org/10.1109/iembs.2007.4352627

[27] S. Abbate, M. Avvenuti, P. Corsini, A. Vecchio, and J. Light, "Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: A survey," in *Wireless Sensor Networks: Application-Centric Design*, G. V. Merrett and Y. K. Tan, Eds. Rijeka, Croatia: InTech, 2010, pp. 147–166, ch. 9.

[28] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4544–4553, Aug. 2015. [Online]. Available: https://doi.org/10.1109/jsen.2015.2423562

[29] B. Aguiar, T. Rocha, J. Silva, and I. Sousa, "Accelerometer-based fall detection for smartphones," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Lisbon, Portugal, Jun. 2014, pp. 480–485. [Online]. Available: https://doi.org/10.1109/MeMeA.2014.6860110

[30] G. Rescio, A. Leone, and P. Siciliano, "Supervised expert system for wearable MEMS accelerometer-based fall detector," *J. Sensors*, vol. 2013, Jul. 2013, Art. no. 254629. [Online]. Available: https://doi.org/10.1155/2013/254629

[31] D. Chen, W. Feng, Y. Zhang, X. Li, and T. Wang, "A wearable wireless fall detection system with accelerators," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Phuket, Thailand, Dec. 2011, pp. 2259–2263. [Online]. Available: https://doi.org/10.1109/robio.2011.6181634

[32] H. Rimminen, J. Lindström, M. Linnavuo, and R. Sepponen, "Detection of falls among the elderly by a floor sensor using the electric near field," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 6, pp. 1475–1476, Nov. 2010. [Online]. Available: https://doi.org/10.1109/titb.2010.2051956

[33] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait Posture*, vol. 28, no. 2, pp. 285–291, Aug. 2008. [Online]. Available: https://doi.org/10.1016/j.gaitpost.2008.01.003

[34] O. Aziz, M. Musngi, E. J. Park, G. Mori, and S. N. Robinovitch, "A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials," *Med. Biol. Eng. Comput.*, vol. 55, no. 1, pp. 45–55, Jan. 2017. [Online]. Available: https://doi.org/10.1007/s11517-016-1504-y

[35] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, "A movement decomposition and machine learning-based fall detection system using wrist wearable device," *IEEE Sensors J.*, vol. 18, no. 12, pp. 5082–5089, Jun. 2018. [Online]. Available: https://doi.org/10.1109/jsen.2018.2829815

[36] F. Hussain, F. Hussain, M. E.-U. Haq, and M. A. Azam, "Activity-aware fall detection and recognition based on wearable sensors," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019. [Online]. Available: https://doi.org/10.1109/jsen.2019.2898891

[37] A. Patel and J. Shah, "Sensor-based activity recognition in the context of ambient assisted living systems: A review," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 4, pp. 301–322, Jul. 2019. [Online]. Available: https://doi.org/10.3233/ais-190529

[38] A. Ramachandran and A. Karuppiah, "A survey on recent advances in wearable fall detection systems," *Biomed. Res. Int.*, vol. 2020, Jan. 2020, Art. no. 2167160. [Online]. Available: https://doi.org/10.1155/2020/2167160

[39] X. Wang, J. Ellul, and G. Azzopardi, "Elderly fall detection systems: A literature survey," *Front. Robot. AI*, vol. 7, Jun. 2020, Art. no. 71. [Online]. Available: https://doi.org/10.3389/frobt.2020.00071

[40] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, "Sensor technologies for fall detection systems: A review," *IEEE Sensors J.*, vol. 20, no. 13, pp. 6889–6919, Jul. 2020. [Online]. Available: https://doi.org/10.1109/jsen.2020.2976554

[41] L. Chen and C. D. Nugent, Eds., *Human Activity Recognition and Behaviour Analysis for Cyber-Physical Systems in Smart Environments*. Cham, Switzerland: Springer, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-19408-6

[42] J. Clemente, F. Li, M. Valero, and W. Song, "Smart seismic sensing for indoor fall detection, location, and notification," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 524–532, Feb. 2020. [Online]. Available: https://doi.org/10.1109/jbhi.2019.2907498

[43] K.-H. Chen, Y.-W. Hsu, J.-J. Yang, and F.-S. Jaw, "Evaluating the specifications of built-in accelerometers in smartphones on fall detection performance," *Instrum. Sci. Technol.*, vol. 46, no. 2, pp. 194–206, 2018. [Online]. Available: https://doi.org/10.1080/10739149.2017.1363054

[44] B. Barshan and A. Yurtman, "Classifying daily and sports activities invariantly to the positioning of wearable motion sensor units," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4801–4815, Jun. 2020. [Online]. Available: https://doi.org/10.1109/jiot.2020.2969840

[45] A. Yurtman, B. Barshan, and S. Redif, "Position invariance for wearables: Interchangeability and single-unit usage via machine learning," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8328–8342, May 2021. [Online]. Available: https://doi.org/10.1109/jiot.2020.3044754

[46] W. Y. Toh, Y. K. Tan, W. S. Koh, and L. Siek, "Autonomous wearable sensor nodes with flexible energy harvesting," *IEEE Sensors J.*, vol. 14, no. 7, pp. 2299–2306, Jul. 2014. [Online]. Available: https://doi.org/10.1109/jsen.2014.2309900

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

[47] A. T. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," *Sensors*, vol. 16, no. 8, Jul. 2016, Art. no. 1161. [Online]. Available: https://doi.org/10.3390/s16081161

[48] P. Ntanasis, E. Pippa, A. T. Özdemir, B. Barshan, and V. Megalooikonomou, "Investigation of sensor placement for accurate fall detection," in *Wireless Mobile Communication and Healthcare* (Lecture Notes of the Institute for Computer Sciences, Social Informatics, and Telecommunications Engineering (LNICST)), vol. 192, P. Perego, G. Andreoni, and G. Rizzo, Eds. Cham, Switzerland: Springer Int., Jun. 2017, pp. 225–232. [Online]. Available: https://doi.org/10.1007/978-3-319-58877-3_30

[49] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "A hybrid temporal reasoning framework for fall monitoring," *IEEE Sensors J.*, vol. 17, no. 6, pp. 1749–1759, Mar. 2017. [Online]. Available: https://doi.org/10.1109/jsen.2017.2649542

[50] J. Dai, X. Bai, Z. Yang, Z. Shen, and D. Xuan, "PerFallD: A pervasive fall detection system using mobile phones," in *Proc. 8th IEEE Int. Conf. Pervasive Comput. Commun. (PERCOM) Workshops*, Mannheim, Germany, Mar./Apr. 2010, pp. 292–297. [Online]. Available: https://doi.org/10.1109/percomw.2010.5470652

[51] S.-H. Fang, Y.-C. Liang, and K.-M. Chiu, "Developing a mobile phone-based fall detection system on Android platform," in *Proc. IEEE Comput. Commun. Appl. Conf. (ComComAp)*, Hong Kong, Jan. 2012, pp. 143–146. [Online]. Available: https://doi.org/10.1109/ComComAp.2012.6154019

[52] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, Jul. 2007. [Online]. Available: https://doi.org/10.1016/j.gaitpost.2006.09.012

[53] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker, "Evaluation of a fall detector based on accelerometers: A pilot study," *Med. Biol. Eng. Comput.*, vol. 43, no. 5, pp. 548–551, Oct. 2005. [Online]. Available: https://doi.org/10.1007/bf02351026

[54] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, pp. 619–627, Dec. 2010. [Online]. Available: https://doi.org/10.1109/tnsre.2010.2070807

[55] W. Saadeh, S. A. Butt, and M. A. Bin Altaf, "A patient-specific single sensor IoT-based wearable fall prediction and detection system," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 5, pp. 995–1003, May 2019. [Online]. Available: https://doi.org/10.1109/tnsre.2019.2911602

[56] F.-T. Wang, H.-L. Chan, M.-H. Hsu, C.-K. Lin, P.-K. Chao, and Y.-J. Chang, "Threshold-based fall detection using a hybrid of tri-axial accelerometer and gyroscope," *Physiol. Meas.*, vol. 39, no. 10, Oct. 2018, Art. no. 105002. [Online]. Available: https://doi.org/10.1088/1361-6579/aae0eb

[57] R. A. Torres-Guzman et al., "Smartphones and threshold-based monitoring methods effectively detect falls remotely: A systematic review," *Sensors*, vol. 23, no. 3, Jan. 2023, Art. no. 1323. [Online]. Available: https://doi.org/10.3390/s23031323

[58] S. S. Khan and J. Hoey, "Review of fall detection techniques: A data availability perspective," *Med. Eng. Phys.*, vol. 39, pp. 12–22, Jan. 2017. [Online]. Available: https://doi.org/10.1016/j.medengphy.2016.10.014

[59] S. Abbate, M. Avvenuti, G. Cola, P. Corsini, J. Light, and A. Vecchio, "Recognition of false alarms in fall detection systems," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2011, pp. 23–28. [Online]. Available: https://doi.org/10.1109/ccnc.2011.5766464

[60] A. Chelli and M. Pätzold, "A machine learning approach for fall detection and daily living activity recognition," *IEEE Access*, vol. 7, pp. 38670–38687, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2906693

[61] M. Saleh and R. L. B. Jeannès, "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 2019. [Online]. Available: https://doi.org/10.1109/jsen.2019.2891128

[62] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, "Latest research trends in fall detection and prevention using machine learning: A systematic review," *Sensors*, vol. 21, no. 15, Jul. 2021, Art. no. 5134. [Online]. Available: https://doi.org/10.3390/s21155134

[63] N. Thakur and C. Y. Han, "A study of fall detection in assisted living: Identifying and improving the optimal machine learning method," *J. Sensor Actuator Netw.*, vol. 10, no. 3, Jun. 2021, Art. no. 39. [Online]. Available: https://doi.org/10.3390/jsan10030039

[64] M. M. Islam et al., "Deep learning based systems developed for fall detection: A review," *IEEE Access*, vol. 8, pp. 166117–166137, 2020. [Online]. Available: https://doi.org/10.1109/access.2020.3021943

[65] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1276–1289, Jul./Sep. 2021. [Online]. Available: https://doi.org/10.1109/tetc.2020.3027454

[66] S. Rastogi and J. Singh, "A systematic review on machine learning for fall detection system," *Comput. Intell.*, vol. 37, no. 2, pp. 951–974, May 2021. [Online]. Available: https://doi.org/10.1111/coin.12441

[67] L. A. Zadeh, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh* (Advances in Fuzzy Systems—Applications and Theory), vol. 6, G. J. Klir and B. Yuan, Eds. Singapore: World Sci., May 1996. [Online]. Available: https://doi.org/10.1142/2895

[68] M. Ş. Turan, "Fall detection and classification using wearable motion sensors," M.S. thesis, Dept. Elect. Electron. Eng., Bilkent Univ., Ankara, Turkey, Sep. 2017. [Online]. Available: https://repository.bilkent.edu.tr/items/ea5cb6b1-675b-47df-8e44-c8d8a2167d99

[69] A. T. Özdemir and B. Barshan, "Simulated Falls and Daily Living Activities Data Set," *UCI Machine Learning Repository*, School Inf. Comput. Sci., University of California at Irvine, Irvine, CA, USA, donated Jun. 2018. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Simulated+Falls+and+Daily+Living+Activities+Data+Set

[70] A. T. Özdemir and B. Barshan, "Simulated Falls and Daily Living Activities Data Set," *IEEE DataPort*, donated Dec. 2022. [Online]. Available: https://doi.org/10.21227/eqmh-8m79

[71] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, Jun. 2014. [Online]. Available: https://doi.org/10.3390/s140610691

[72] M. Ş. Turan and B. Barshan, "Classification of fall directions via wearable motion sensors," *Digit. Signal Process.*, vol. 125, Jun. 2022, Art. no. 103129. [Online]. Available: https://doi.org/10.1016/j.dsp.2021.103129

[73] E. Kavuncuoğlu, E. Uzunhisarcıklı, B. Barshan, and A. T. Özdemir, "Investigating the performance of wearable motion sensors on recognizing falls and daily activities via machine learning," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103365. [Online]. Available: https://doi.org/10.1016/j.dsp.2021.103365

[74] E. Pippa, E. I. Zacharaki, A. T. Özdemir, B. Barshan, and V. Megalooikonomou, "Global vs local classification models for multi-sensor data fusion," in *Proc. 10th Hellenic Conf. Artif. Intell. (SETN)*, Patras, Greece, Jul. 2018, pp. 9–12. [Online]. Available: https://doi.org/10.1145/3200947.3201034

[75] K.-C. Liu, C.-Y. Hsieh, S. J.-P. Hsu, and C.-T. Chan, "Impact of sampling rate on wearable-based fall detection systems based on machine learning models," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9882–9890, Dec. 2018. [Online]. Available: https://doi.org/10.1109/jsen.2018.2872835

[76] J. A. Santoyo-Ramón, E. Casilari, and J. M. Cano-García, "A study of the influence of the sensor sampling frequency on the performance of wearable fall detectors," *Measurement*, vol. 193, Apr. 2022, Art. no. 110945. [Online]. Available: https://doi.org/10.1016/j.measurement.2022.110945

[77] G. Anania et al., "Development of a novel algorithm for human fall detection using wearable sensors," in *Proc. 7th IEEE Sensors Conf.*, Lecce, Italy, Oct. 2008, pp. 1336–1339. [Online]. Available: https://doi.org/10.1109/icsens.2008.4716692

[78] W.-S. Baek, D.-M. Kim, F. Bashir, and J.-Y. Pyun, "Real life applicable fall detection system based on wireless body area network," in *Proc. IEEE 10th Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2013, pp. 62–67. [Online]. Available: https://doi.org/10.1109/ccnc.2013.6488426

[79] A. K. Bourke, K. J. O'Donovan, J. Nelson, and G. ÓLaighin, "Fall-detection through vertical velocity thresholding using a tri-axial accelerometer characterized using an optical motion-capture system," in *Proc. 30th Annu. Int. IEEE EMBS Conf.*, Vancouver, BC, Canada, Aug. 2008, pp. 2832–2835. [Online]. Available: https:/ /doi.org/10.1109/iembs.2008.4649792

[80] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Med. Eng. Phys.*, vol. 30, no. 1, pp. 84–90, Jan. 2008. [Online]. Available: https://doi.org/10.1016/j.medengphy.2006.12.001

[81] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Proc. 27th Annu. Int. IEEE EMBS Conf.*, Shanghai, China, Sep. 2005, pp. 3551–3554. [Online]. Available: https://doi.org/10.1109/iembs.2005.1617246

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.

[82] J. Jacob et al., "A fall detection study on the sensors placement location and a rule-based multi-thresholds algorithm using both accelerometer and gyroscopes," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ)*, Taipei, Taiwan, Jun. 2011, pp. 666–671. [Online]. Available: https://doi.org/10.1109/fuzzy.2011.6007744

[83] P. Jantaraprim, P. Phukpattaranont, C. Limsakul, and B. Wongkittisuksa, "Improving the accuracy of a fall detection algorithm using free fall characteristics," in *Proc. IEEE Int. Conf. Elect. Eng. Electron. Comput. Telecommun. Inf. Technol. (ECTI-CON)*, Chiang Mai, Thailand, May 2010, pp. 501–504.

[84] A. Sorvala, E. Alasaarela, H. Sorvoja, and R. Myllylä, "A two-threshold fall detection algorithm for reducing false alarms," in *Proc. IEEE 6th Int. Symp. Med. Inf. Commun. Technol (ISMICT)*, La Jolla, CA, USA, Mar. 2012. [Online]. Available: https://doi.org/10.1109/ismict.2012.6203028

[85] C.-C. Wang et al., "Development of a fall detecting system for the elderly residents," in *Proc. IEEE 2nd Int. Conf. Bioinformatics Biomed. Eng. (ICBBE)*, Shanghai, China, May 2008, pp. 1359–1362. [Online]. Available: https://doi.org/10.1109/icbbe.2008.669

[86] L. Wang, L. Cheng, and G. Zhao, *Machine Learning for Human Motion Analysis: Theory and Practice*. Hershey, PA, USA: IGI Global, 2010.

[87] F. Bagalà et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS ONE*, vol. 7, no. 5, May 2012, Art. no. e37062. [Online]. Available: https://doi.org/10.1371/journal.pone.0037062

[88] C. Mosquera-Lopez et al., "Automated detection of real-world falls: Modeled from people with multiple sclerosis," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 6, pp. 1975–1984, Jun. 2021. [Online]. Available: https://doi.org/10.1109/jbhi.2020.3041035

[89] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jämsä, "Sensitivity and specificity of fall detection in people aged 40 years and over," *Gait Posture*, vol. 29, no. 4, pp. 571–574, Jun. 2009. [Online]. Available: https://doi.org/10.1016/j.gaitpost.2008.12.008

[90] G. Wu and S. Xue, "Portable preimpact fall detector with inertial sensors," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 178–183, Apr. 2008. [Online]. Available: https://doi.org/10.1109/tnsre.2007.916282

[91] A. K. Bourke et al., "Evaluation of waist-mounted tri-axial accelerometer based fall-detection algorithms during scripted and continuous unscripted activities," *J. Biomechan.*, vol. 43, no. 15, pp. 3051–3057, Nov. 2010. [Online]. Available: https://doi.org/10.1016/j.jbiomech.2010.07.005

[92] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jämsä, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait Posture*, vol. 35, no. 3, pp. 500–505, Mar. 2012. [Online]. Available: https://doi.org/10.1016/j.gaitpost.2011.11.016

[93] H. Jung, B. Koo, J. Kim, T. Kim, Y. Nam, and Y. Kim, "Enhanced algorithm for the detection of preimpact fall for wearable airbags," *Sensors*, vol. 20, no. 5, Feb. 2020, Art. no. 1277. [Online]. Available: https://doi.org/10.3390/s20051277

[94] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, Jan. 2017, Art. no. 198. [Online]. Available: https://doi.org/10.3390/s17010198

[95] F. A. S. F. de Sousa, C. Escriba, E. G. A. Bravo, V. Brossa, J.-Y. Fourniols, and C. Rossi, "Wearable pre-impact fall detection system based on 3D accelerometer and subject's height," *IEEE Sensors J.*, vol. 22, no. 2, pp. 1738–1745, Jan. 2022. [Online]. Available: https://doi.org/10.1109/jsen.2021.3131037

[96] Z. Qian et al., "Development of a real-time wearable fall detection system in the context of Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21999–22007, Nov. 2022. [Online]. Available: https://doi.org/10.1109/jiot.2022.3181701

[97] W. D. Patel, C. Patel, and M. Patel, "VitaFALL: Advanced multi-threshold based reliable fall detection system," *Recent Adv. Comput. Sci. Commun.*, vol. 15, no. 1, pp. 32–39, Sep. 2022. [Online]. Available: https://doi.org/10.2174/2666255813999200904132939

[98] N. I. M. Amir, R. A. Dziyauddin, N. Mohamed, N. S. N. Ismail, N. S. A. Zulkifli, and N. M. Din, "Real-time threshold-based fall detection system using wearable IoT," in *Proc. 4th Int. Conf. Smart Sensors Appl. (ICSSA)*, Kuala Lumpur, Malaysia, Jul. 2022, pp. 173–178. [Online]. Available: https://doi.org/10.1109/icssa54161.2022.9870974

[99] J.-S. Lee and H.-H. Tseng, "Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8293–8302, Sep. 2019. [Online]. Available: https://doi.org/10.1109/jsen.2019.2918690

[100] N. Otanasap, "Pre-impact fall detection based on wearable device using dynamic threshold model," in *Proc. 17th Int. Conf. Parallel Distrib. Comput. Appl. Technol. (PDCAT)*, Guangzhou, China, Dec. 2016, pp. 362–365. [Online]. Available: https://doi.org/10.1109/pdcat.2016.083

[101] D. Razum, G. Šeketa, J. Vugrin, and I. Lacković, "Optimal threshold selection for threshold-based fall detection algorithms with multiple features," in *Proc. 41st Int. Convention Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, Optaija, Croatia, May 2018, pp. 1513–1516. [Online]. Available: https://doi.org/10.23919/mipro.2018.8400272

[102] J. Qu, C. Wu, Q. Li, T. Wang, and A. H. Soliman, "Human fall detection algorithm design based on sensor fusion and multi-threshold comprehensive judgment," *Sensors Mater.*, vol. 32, no. 4, pp. 1209–1221, Apr. 2020. [Online]. Available: https://doi.org/10.18494/sam.2020.2527

[103] J. Yang, T. Gao, and K. Huang, "A fall detection method based on multi-threshold value and EML learning machine," *AIP Conf. Proc.*, vol. 2073, no. 1, Feb. 2019, Art. no. 020051. [Online]. Available: https://doi.org/10.1063/1.5090705

[104] A. Shahzad and K. Kim, "FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 35–44, Jan. 2019. [Online]. Available: https://doi.org/10.1109/tii.2018.2839749

[105] T. Xu, H. Se, and J. Liu, "A two-step fall detection algorithm combining threshold-based method and convolutional neural network," *Metrol. Meas. Syst.*, vol. 28, no. 1, pp. 23–40, 2021. [Online]. Available: https://doi.org/10.24425/mms.2021.135999

[106] T. Xu, H. Se, and J. Liu, "A fusion fall detection algorithm combining threshold-based method and convolutional neural network," *Microprocess. Microsyst.*, vol. 82, Apr. 2021, Art. no. 103828. [Online]. Available: https://doi.org/10.1016/j.micpro.2021.103828

[107] H. Zhang, M. Alrifaai, K. Zhou, and H. Hu, "A novel fuzzy logic algorithm for accurate fall detection of smart wristband," *Trans. Inst. Meas. Control*, vol. 42, no. 4, pp. 786–794, Feb. 2020. [Online]. Available: https://doi.org/10.1177/0142331219881578

Billur Barshan received the B.Sc. degrees in electrical engineering and in physics from Boğaziçi University, Istanbul, Turkey, in 1986, and the M.S. and M.Phil. degrees in 1988 and the Ph.D. degree in 1991 in electrical engineering from Yale University, New Haven, CT, USA.

After working as a Post-Doctoral Researcher with the Robotics Research Group, University of Oxford, Oxford, U.K., she joined the Faculty of Bilkent University, Ankara, Turkey, where she is currently a Professor with the Department of Electrical and Electronics Engineering. Her current research interests include wearable sensing, wearable robots and mechanisms, intelligent sensing, motion capture and analysis, detection and classification of falls, machine/deep learning, pattern recognition, and multi-sensor data fusion.

Dr. Barshan received the TÜBİTAK Incentive Award in 1998, the METU Mustafa Parlar Foundation Research Award in 1999, and two best paper awards. She served on the Management Committee of the COST-IC0903 Action MOVE from 2010 to 2013. She currently serves on the Editorial Board of *Digital Signal Processing* (Elsevier) journal and as a Guest Editor for IEEE SELECTED TOPICS IN SIGNAL PROCESSING.

Mustafa Şahin Turan (Member, IEEE) received the B.S. degree in mechatronics engineering from Sabancı University, Istanbul, Turkey, in 2015, the M.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2017, and the Ph.D. degree in mechanical engineering from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2021.

He was a Research and Teaching Assistant with Bilkent University from 2015 to 2017 and with EPFL from 2017 to 2021. He is currently a Data Scientist with Source.ag, Amsterdam, Netherlands. His research interests include cyber-attack detection and distributed control in large scale systems, data-driven control, networked control systems, and microgrids.

Authorized licensed use limited to: ULAKBIM UASL - Bilkent University. Downloaded on October 10,2023 at 13:51:09 UTC from IEEE Xplore. Restrictions apply.