International Journal of E-Health and Medical Communications
Volume 12 • Issue 1 • January-March 2021

Human Fall Detection Using Efficient Kernel and Eccentric Approach

Rashmi Shrivastava, NIT, Raipur, India
Manju Pandey, NIT, Raipur, India

ABSTRACT

Unintentional human falls are a very crucial problem in elderly people. If the fall goes unnoticed or undetected, it can lead to severe injuries and can even lead to death. Detecting falls as early as possible is very important to avoid severe physical injurious and mental trauma. The objective of this paper is to design the fall detection model using data of daily living activities only. In the proposed fall detection model, an eccentric approach with SVM based one-class classification is used. For the pre-processing step, fast fourier transformation has been applied to the data and seven features have been calculated using the preprocessed ADL dataset that has been calculated from the dataset of ADL (activities of daily living) activities acquired from the smartphones. An enhancement of the chi-square kernel-based support vector machine has been proposed here for classifying ADL activities from fall activities. Using the proposed algorithm, 98.81% sensitivity and 98.65% specificity have been achieved. This fall detection model achieved 100% accuracy on the FARSEEING dataset.

KEYWORDS

ADL, Anomaly, Eccentric, Fall Detection, Kernel, One-Class, Smartphone, SVM

INTRODUCTION

In recent years, the trend of living alone has been increased which exposed elderly persons of families to be unaccompanied. This phenomenon increased the risk of falling especially in elderly people or people suffering from diseases like Alzheimer's, dementia, diabetes, etc. Among all these Alzheimer is a major cause of falls in elderly people. According to Global QYResearch via COMTEX report "the global fall detection systems market was valued at USD 365million in 2018 and is expected to reach a market valuation of approximately USD 544 billion by 2026 growing at a CAGR of 4.2% during the forecast period" (Global QYResearch 2019). Human fall can be stated as suddenly going down towards the ground unintentionally due to any disease, unconsciousness, and weakness, etc. or activity as a result of an accident. Falls can cause severe physical and psychological injuries to the person including death. In addition to the physical disturbance people also suffer from fear, anxiety, physical injuries, expensive medication, long hospital stay, social burden, etc. If immediate help does not reach the older people then it can result in fracture and long lie and further delay may even cause the casualty. Human falls detection is becoming a big challenge to encounter effectively.

Sensors are an essential component for the detection of human falls. Many different types of sensors are available like accelerometer, gyroscope, camera, vibration, etc. These sensors can be categorized into three different types (1) wearable sensors (2) ambient sensor and (3) vision-based sensors. The wearable sensor is those sensors that can be worn around the waist, neck, and wrist. Some authors have proposed wearable based fall detection methods(Sztyler, Stuckenschmidt, and

DOI: 10.4018/IJEHMC.2021010105

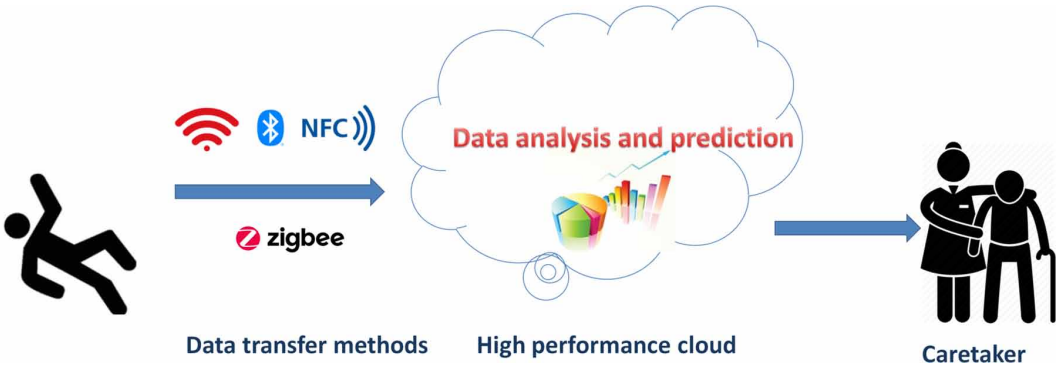
This article, published as an Open Access article on January 7, 2021 in the gold Open Access journal, International Journal of E-Health and Medical Communications (converted to gold Open Access January 1, 2021), is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Petrich 2017; Zhang, Wang, et al. 2006). Ambient sensors imply vibration sensors, acoustics sensor or a combination of them(Noor, Salcic, and Wang 2017). Vision-based sensors are considered most non-obtrusive sensors but they are not easily portable whereas the wearable sensor can be easily worn and easily portable. Since the last few decades, the induction of smartphones among the daily living of people and advanced technologies have created a favorable environment for the growth of the fall detection system. The working of fall detection systems can be understood as shown in Figure 1. IoT technology provides a simple platform to capture human fall detection efficiently. As the algorithms are very complex to run in smartphones, therefore, data is collected via smartphone and processed at cloud environment.

Falls can be detected using two ways: using threshold values(Abbate et al. 2012; Wang et al. 2014) and by using machine learning techniques(Albert et al. 2012; Zhang, Wang, et al. 2006). In threshold-based approaches, calculated values of resultant are checked, if it is between lower and upper threshold values then it is daily living activity otherwise it is fall. Although machine learning algorithms are more complex than threshold-based methods but they provide better results than threshold-based methods. Machine learning-based fall detection algorithm uses various techniques like neural networks (Musci et al. 2018), classification algorithms, deep learning (Shin et al. 2016) (Ma et al. 2014), etc.

Different types of machine learning-based classification algorithms were used for fall detection. Some most popular algorithms are support vector machine (SVM), K-nearest neighbor (KNN), Naive Bayes algorithm, etc. There are two types of classification techniques (1) Multiclass classification (2) Binary classification. The Multiclass classifier can classify the dataset into more than two classes. Binary classification is a method, which classifies the dataset into the two classes. From the binary classification, one more classification strategy was generated, which is one-class classification. In binary class classification, the classifier is trained with positive as well as negative examples and tested on both whereas one-class classification is used in the scenario where examples of negative class either not present or poorly present. One-class classification is trained using only positive class examples and predicts only two values: 1 if data is presented in class which was used for training and 0 if data is outside of the class which was used for training. One class classification is more complex than binary or multiclass classification because it is trained using only positive class (TAX and D 2001). One class classification can be categorized based on the availability of data, type of algorithm used and application domain. In 1975 minter used the term one class classifier in the context of learning Bayes classifier that requires only labeled data from the "class of interest". Then different researchers presented similar concepts such as Outlier Detection (Ritter and Gallegos 1997), Novelty Detection (Bishop 1994) or Concept Learning(Japkowicz 1999).

Figure 1. Fall detection scenario



Contribution of Paper

- The proposed fall detection model is built using only ADL activity data so data of falls is not required to build a fall detection model which avoids building of imbalance fall detection model.
- The eccentric kernel and support vector machine-based classifier is proposed here to build a fall detection model. The proposed fall detection model performed better than two discussed methods which were tested on the same dataset using one-class classification.
- This proposed method is also tested on real fall data and performed very well.

In this paper, a one-class classification based fall detection model has been proposed. For this purpose, authors have used two publically available datasets of ADL and Fall activities. From these datasets, only Smartphone's accelerometer data in x,y,z direction is used. At first, resultant is calculated then seven features are calculated from resultants. Using the kernel trick, the kernel matrix is calculated using training data. The support vector machine-based one-class classification is used here to build the fall detection model. This model is tested on simulated ADL data, fall data, and real-world fall data. Section II describes related work in the area of fall detection using one-class classification. Section III describes the methodology of proposed research work, Section IV describes results and discussion and section V describes the conclusion of the proposed research work.

RELATED WORK

Many researchers have explored the human fall detection area by employing many algorithms, tools, and techniques. Non-obstructive fall detection methods were implemented using vision-based techniques. In 2018, (Min, Zou, and Li 2019) presented an approach of fall detection using the shape aspect ratio. In 2019, (Fan, Wang, and Zhuang 2019) proposed slow feature analysis to detect vision-based falls using the SDUFall dataset. In 2019, (Chen et al. 2019) analyze the effect of camera height on vision-based fall detection. In 2019, (Tsai, Wang, and Hsu 2019) proposed fall detection which was based on computer vision techniques. For analysis purposes, they used only the hue image. In 2019 (Adhikari, Bouchachia, and Nait-Charif 2019) presented a vision-based method to detect falls. They used Skelton joint position from Microsoft Kinect. They grouped the all joint position in three clusters: head, torso, and leg. They trained the fall detection model using CNN and found that their approach of dividing joint position into three clusters was very promising. Vision-based fall detection was also proposed by (Ding et al. 2019). They proposed a "real-time appropriate dynamic image (RT-ADI) " method based on fall detection using multiple cameras taking videos from different angles. In 2020(Dhiraj et al. 2020) proposed Indoor fall detection by using 360^0 video cameras with 90% accuracy. In 2020, (Subramanyan KAVYA et al. 2020) proposed fall detection using texture segmentation. In 2020, (Qiu et al. 2020) proposed vision-based fall detection using deep sort and YOLOv3 algorithms.

Many Researchers (Principi et al. 2016) (Droghini et al. 2017)(Irtaza et al. 2017) (Adnan et al. 2018)proposed fall detection using acoustic sensors. Some fall detection methods were also proposed using multisensor fusion. (Su, Liu, and Wu 2017) used angular velocity and angle to data for fall detection, (Zhou et al. 2018) used wave radar and optical camera to detect fall, (Khan et al. 2019) used camera, gyroscope, and accelerometer, (Pham et al. 2019) proposed multisensor data fusion using accelerometer, gyroscope, magnetometer, barometer, and MQ7 sensor. Some fall detection methods (Tsai, Wang, and Hsu 2019)(Narui et al. 2020)(Yang et al. 2018)(Palipana and Rojas 2017)(Gu et al. 2018) used wifi device to detect fall. Many other methods (Ahmad et al. 2016; Amin et al. 2016; Erol and Amin 2018; Q. Wan et al. 2014) have used radar signals for fall detection. Wearable based fall detection methods were proposed by (Lopes, Vaidya, and Rodrigues 2009) (Gjoreski et al. 2020) (Kerdjidj et al. 2020)(Giuffrida et al. 2019)(Saleh and Jeannes 2019)(Boutellaa, Kerdjidj, and Ghanem 2019)(Hussain et al. 2019)(Silva de Lima et al. 2019) in recent years. Some IoT based fall detection

systems(Hsieh and Jeng 2018)(Greene, Thapliyal, and Carpenter 2017)(Yacchirema et al. 2018)(Hsu et al. 2017)(Chandra et al. 2018)(Santos et al. 2016) were also proposed by many researchers. In 2013, a fall prevention method (Horta et al. 2013) was proposed, which used accelerometer signals along with physiological signals electrocardiogram sensors, electromyography sensors, and EDA sensors.

Much diversified research work has been done in the fall detection area. Various classification algorithms like SVM classifier, KNN classifier, decision tree classifier, etc. have been used to detect falls but very less research has been done in the area of fall detection using one-class classification. One class classification has been applied in various application areas like smart home, security(M. Wan, Shang, and Zeng 2017)(Guerbai, Chibani, and Hadjadji 2015), image processing(Krawczyk and Filipczuk 2014), Gaussian process(Kemmler, Rodner, and Denzler 2011), machine learning(Leng et al. 2015).

One class classification is suitable for fall detection because data of real-world fall is either presented in fewer amounts or not presented and the available data is simulated data. Initially, one class classification based fall detection method using accelerometer data was proposed in(Zhang, Wang, et al. 2006). They used axis a_{xz} as a strong feature to detect falls. Their one-class SVM model was trained using positive samples from the fall of younger volunteers and dummy and they tested outliers using non-fall activities of younger and elderly persons. They found that the correct ratio to classify non-fall activities was 96.7%. In 2006 a fall detection method was proposed by (Zhang, Zhang, et al. 2006) using the Kernel Fisher Discriminate algorithm. Preprocessing of data was done using one-class SVM. KFD and KNN were used for prediction. They collected the data from 32 volunteers. In their first method in (Zhang, Wang, et al. 2006), they used one-step computation but in their second method (Zhang, Zhang, et al. 2006) they have used two-step verification method which was more precise than the earlier method. In this paper, they also included fall data for testing. The correctness ratio was 89.1% using this method. After that many video sensor and acoustic sensor-based fall detection based methods(Popescu and Mahnot 2009)(Miao Yu, Naqvi, et al. 2011)(Miao Yu, Rhuma, et al. 2011)(M. Yu et al. 2012) were also proposed. Some fall detection methods (Zhang, Wang, et al. 2006)(Zhang, Zhang, et al. 2006)(Medrano et al. 2014)(Micucci et al. 2017)(Medrano et al. 2017) have used one-class classifier to detect falls.

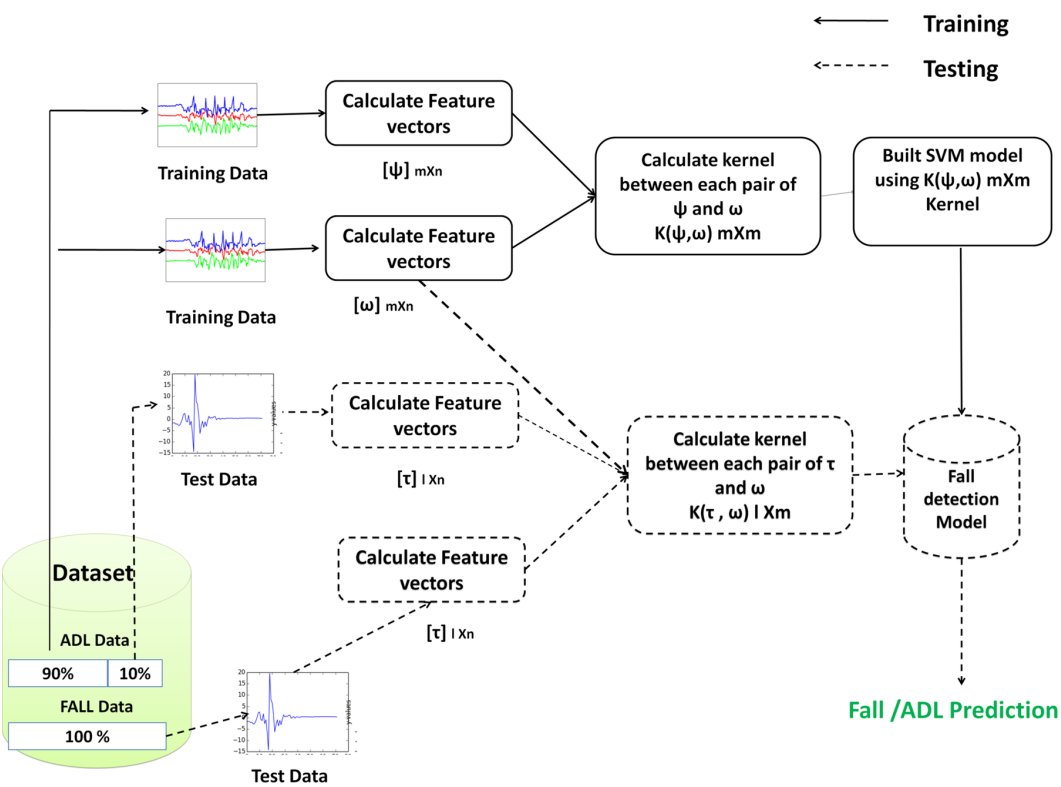
The use of one class classifier to detect falls was also proposed by (Medrano et al. 2014) in their paper "Detecting Falls as Novelties in Acceleration Patterns Acquired with Smartphone" in 2014. They trained the novelty based fall detection model using SVM and KNN classifiers. In 2015 (Micucci et al. 2017) presented an approach to detect falls as anomalies. They also used a one-class classification on publically available datasets (Medrano et al. 2014). They compared the KNN and SVM classifiers based on one- class classification and two-class classification. They concluded that the SVM method was better than KNN classifiers. Here, however, results of the two-class classifier were better than one class classifier but in real-life situations, the performance of two-class classification drops significantly because in two-class classification simulated fall data is used for training the model. After some years, (Medrano et al. 2017) revised their approach by combining novelty detectors to improve fall detection in 2017. They proposed two combination rules: first combination rule was based on probabilities and the second combination rule was based on output labels. They used a blend of novelty detectors rather than single novelty detector. They got 0.979 sensitivity, 0.967 specificity while using a combination of novelty detectors and 0.964 sensitivity, 0.955 specificity while using SVM based method. A method proposed by (Droghini et al. 2017), which is based on using an acoustic sensor and used a combination of one class SVM and template matching classifier to detect fall. Their method was divided into two steps. One class SVM was used for abnormal event detection and template matching was used for fall detection. They found that their method was more efficient as compared to OCSVM only approaches. A method to detect real-time falls for Korean farmers was proposed by (Kim et al. 2019). In this method, the accelerometer sensor and pressure sensor is fitted insole of shoes. They achieved accuracy 98.4%, sensitivity 98.9%, and specificity 91.3%. This method was based on threshold-based fall detection. In an IoT environment, a method was proposed

by (Nguyen, Nguyen, and Ngo 2020). They used the SVM algorithm for fall detection and the PCA method for feature selection. They found that the method is 82.6% accurate to detect falls and 98.4% accurate to detect ADL activities. Their method did not use one-class SVM.

METHODOLOGY

In this section, at first, the dataset and feature extraction process are explained then SVM based one-class classification using an improved kernel is explained. The fall detection model which has been developed is shown in Figure 2. At first ADL activities (Medrano et al. 2014) are divided into training data and test data in the ratio of 90:10. 90% of ADL data is used to build the fall detection model and 10% of ADL data along with fall data and real-world fall data are used to test the model. After applying fast Fourier transformation, seven features namely standard deviation, root mean square, post-impact, mean frequency, median frequency, peak to peak distance and peak to RMS have been calculated. Then eccentric kernel-based one-class classification using the support vector machine is used to build a fall detection model. The same process of data pre-processing along with feature calculation is also applied to test data. The proposed fall detection model is developed in MATLAB 2015a using the LIBSVM library for SVM based one-class classification.

Figure 2. Fall Detection model



Dataset used -Authors have used two publically available datasets for the experiment. The first dataset is (Medrano et al. 2014). In the process of collecting the dataset, there was 10 volunteer (7

males and 3 females). Their weight ranges between 54 to 98 kg, age 20 to 42 years and height from 1.61 to 1.81 meters. Data was collected using the Samsung Galaxy mini phone having android operating system version 2.2. The sampling rate during data collection was 45±12Hz. On average, 50 Hz sampling rate was considered. During data collection, volunteers carried smartphones in any of their pockets whether left pocket or right pocket for one week and did all daily living activities like walking, running, sitting, etc. in their sequence. On average, 800 records of ADL activities were obtained from each subject. Whenever acceleration magnitude was greater than 1.5g, then a new record entry of 301 values having 150 values before the peak and 150 values after the peak was recorded in a file. These 301 values comprise accelerator values in 6 seconds. This dataset also contains various fall activities. This dataset contains 8 different types of falls: forward falls, backward falls, left and right-lateral falls, syncope, sitting on the empty chair, falls using compensation strategies to prevent the impact and falls with contact to an obstacle before hitting the ground. There are total 7816 records for ADL activities and 503 records when the Smartphone is in the pocket of the volunteer. This dataset contains a wide range of ADL and fall activities more than other datasets (Igual, Medrano, and Plaza 2015).

Another dataset, which has been used is the FARSEEING dataset (Jochen *et al.*, 2016). This FARSEEING dataset contains real-world fall. Authors have used this dataset to test the proposed fall detection model because only real-world fall data can verify the fall detection model efficiently. The FARSEEING dataset was collected by 94 fallers with a mean age of 76.1 years and out of 94 fallers there were 51 females and 43 males. This dataset was collected using three sensors made by Samsung which were AcitvePal3, Mini mode and Hybrid containing accelerometer, gyroscope, and magnetometer. These sensors were placed in lower back and thighs. In this experiment, authors have used accelerometer data only. Out of 208 real fall events, the FARSEEING consortium shares only 22 real-world fall events for research purposes. Out of 22 real fall events, 10 fallers were male and 12 fallers were female with mean age 69.5 years, mean height 167 c.m. and mean weight 75.59 kg. All signal files were available in .mat format. This real-world fall dataset is collected at two different sampling frequencies i.e. 20Hz and 100Hz. Authors have used central data of the only 6s around the peak for analysis purposes.

In (Medrano et al. 2014) dataset, Falls were recorded on a soft mattress in the laboratory environment and to record ADL events. Smartphone devices were given to participants for one week to do natural daily activities. The second dataset (Klenk et al. 2016a) was collected at Azienda Sanitaria di Firenze and Robert Bosch Gesellschaft fuer Medizinische Forschung. Both of these were health institutes.

Data Pre-processing - Authors have used ADL activities of the dataset(Medrano et al. 2014) for building the model. ADL activities are divided into the ratio of 90:10. 90% of data is used for building the model and 10% of data is used for testing purposes. The data from Smartphone's accelerometer has acceleration in three directions i.e. x, y, and z. Suppose acceleration values in all x, y and z directions are Ag_x, Ag_y, Ag_z. Then resultant will be

$$A_{G\text{-}xyx} = \sqrt{(Ag_x^2 + Ag_y^2 + Ag_z^2)}$$

Thus, authors got a vector of 301 values i.e. 150 values before peak and 150 values after peak value from accelerometer resultant for ADL activities. Then Authors have used this data of 6s around the peak for further analysis. After that Fast Fourier Transformation is applied to the dataset.

Features Calculation - The dataset contains 301 values per instance so it is necessary to reduce the amount of data to be input to the classifier. For this purpose, it is required to calculate some features as part of data processing.

Let the training instance is A_{G1}, A_{G2}, A_{G3}..... A_{G301}

Standard Deviation (std) - It is a statistical measure that measures the deviation of data from the mean. It has been calculated using the following formula.

$$std = \sqrt{\frac{1}{N-1}\sum_{i=126}^{176}\left(A_{Gi}-\mu\right)^2}$$

where N is the total number of observations in the range.

Root mean square (RMS) - It is the square root of the mean of squares of values. It has been calculated using the following formula.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=126}^{176}\left(A_{Gi}\right)^2}$$

where N is the total number of observations in the range

Post Impact- Authors have considered post impact as the most discriminative feature of activities of daily living (ADL) and fall activities, because in case of fall, after fall activity the people are static on the floor for some time so the accelerometer values are mostly uniform after fall. Post Impact is calculated using the following formula.

$$Post\,impact = \frac{\sum_{i=201}^{301}\left(A_{Gi}-\mu\right)^2}{N}$$

where A_{Gi}: the individual values of $A_{g\text{-}xyz}$ from 201th value to 301th value of the instances

μ: mean of $A_{g\text{-}xyz}$ from 201th value to 301th value of the instances
N: total no of data points from 201th value to 301th value of the instances

Mean Frequency - To calculate mean frequency, at first product of accelerometer values and frequencies in the given range are calculated then it' s sum is divided from a sum of accelerometer values.

$$Mean\,frequency = \frac{\sum_{i=126}^{176} Ii.fi}{\sum_{i=126}^{176} Ii}$$

where:

n = number of frequency bins in the spectrum

fi = frequency of spectrum at bin i of n
Ii = Intensity (dB scale) of spectrum at bin i of n

Median frequency(M_dF)- Median frequency is the frequency at which the signal spectrum is divided into two equal parts with equal amplitude.

$$\sum_{i=1}^{MdF} Si = \sum_{i=MdF}^{N} Si = \frac{1}{2}\sum_{i=1}^{N} Si$$

where Si is a signal spectrum where i =1 to M_dF

Peak To Peak Distance - Peak to peak distance is the maximum negative amplitude of signal subtracted by the maximum positive amplitude of the signal.

Peak to RMS - Peak to Root mean Square (RMS) feature is the ratio of the largest absolute value in data to the RMS value of data.

Here standard deviation, Root mean square, mean frequency, median frequency, peak to peak distance and peak to RMS features are calculated using central 1-second values around the peak whereas post impact is calculated using last 2-second values of 6-second data instance. This vector containing seven features will be used for building the fall detection model.

One class Classifier- The support vector machine (SVM) is a classifier, which classifies given data in different classes according to their feature vectors. SVM classifier can be supervised or unsupervised. One class SVM is one of the variants of the SVM classifier. One-class SVM classifies the target data point from an anomaly data point in feature space F. The one class classification returns +1 in the area where training data exist and 0 elsewhere. One-class SVM was proposed by (Schölkopf et al. 2000) for estimating the support of a high-dimensional distribution.

Let $x_1, x_2, x_3, x_4,...x_n \in X$ is training data where n is the total number of observations. Suppose Φ is a function that maps X into F where F is high dimensional feature space. Kernel k(x,y) can be calculated by the dot product:

$$k(x,y) = (\Phi(x).\Phi(y))$$

Given training vectors $x_i \in R^n$,i=1,...,l without any class information, the primal problem of one-class SVM is

$$min \frac{1}{2}\left\|w\right\|^2 - \rho + \frac{1}{vl}\sum_{i=1}^{l}\xi i$$

$w \epsilon F, \xi \epsilon R^l, \rho \epsilon R$

$$subject\ to \left(W.\Phi\left(xi\right)\right) \geq \rho - \xi i\ \xi i \geq 0,\ i = 1..... l$$

The decision function is

$$f(x) = sgn\ (\sum_{i=1}^{l}\alpha i K\left(xi, x\right) - \rho\ )$$

The decision function is positive for most of the instances of the training set and negative for others. This method creates hyper-plane using w and p which has maximal distance from the origin in feature space F.

Proposed Kernel Matrix - Kernel function maps low dimensional nonlinear data to high dimensional feature map. It converts linearly non-separable data to linearly separable data. The kernel function calculates the inner product between each couple of data. This method is called kernel trick (Theodoridis, Networks, and 2008 n.d.).

Suppose there is a training matrix

$$\chi = [<a_1, b_1, c_1, d_1, e_1, f_1, g_1> <a_2, b_2, c_2, d_2, e_2, f_2, g_2> \ldots\ldots\ldots\ldots <a_m, b_m, c_m, d_m, e_m, f_m, g_m>]$$

where a, b, c, d, e, f, g are calculated features and m is the total number of instances in χ

Let K is a function that maps χ into M where M is a high dimensional feature map. The method to calculate the kernel matrix to build a fall detection model is as follows.

Suppose ψ and ω are two alias of training matrix χ then

$$\psi_{m\times n} = \begin{bmatrix} \psi 11 & \psi 12 & \psi 13 & \cdot & \psi 1n \\ \psi 21 & \psi 22 & \psi 23 & \cdot & \psi 2n \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \psi m1 & \psi m2 & \psi m3 & \cdot & \psi mn \end{bmatrix}$$

$$\omega_{m\times n} = \begin{bmatrix} \omega 11 & \omega 12 & \omega 13 & \cdot & \omega 1n \\ \omega 21 & \omega 22 & \omega 23 & \cdot & \omega 2n \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \omega m1 & \omega m2 & \omega m3 & \cdot & \omega mn \end{bmatrix}$$

Take a matrix

$$G_{m\times m} = \begin{bmatrix} 0 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & 0 \end{bmatrix}$$

For every row in ψ calculate W_i and U_i:

$$Wi_{m\times n} =$$

$$Wi_{m\times n} = \begin{bmatrix} (\psi i1^2 + \omega 11^2) & (\psi i2^2 + \omega 12^2) & (\psi i3^2 + \omega 13^2) & \cdot & (\psi in^2 + \omega 1n^2) \\ (\psi i1^2 + \omega 21^2) & (\psi i2^2 + \omega 22^2) & (\psi i3^2 + \omega 23^2) & \cdot & (\psi in^2 + \omega 2n^2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ (\psi i1^2 + \omega m1^2) & (\psi i2^2 + \omega m2^2) & (\psi i3^2 + \omega m3^2) & \cdot & (\psi in^2 + \omega mn^2) \end{bmatrix}$$

where i= 1.....m

where i= 1.....m

$$Ui_{m\times n} = \begin{bmatrix} (\psi i1 + \omega 11) & (\psi i2 + \omega 12) & (\psi i3 + \omega 13) & \cdot & (\psi in + \omega 1n) \\ (\psi i1 + \omega 21) & (\psi i2 + \omega 22) & (\psi i3 + \omega 23) & \cdot & (\psi in + \omega 2n) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ (\psi i1 + \omega m1) & (\psi i2 + \omega m2) & (\psi i3 + \omega m3) & \cdot & (\psi in + \omega mn) \end{bmatrix}$$

where i= 1.....m

Now for every corresponding element of W_i and U_i calculate

$$Hi_{m\times n} = \begin{bmatrix} \frac{2*(\psi i1^2 + \omega 11^2)}{(\psi i1 + \omega 11)} & \frac{2*(\psi i2^2 + \omega 12^2)}{(\psi i2 + \omega 12)} & \frac{2*(\psi i3^2 + \omega 13^2)}{(\psi i3 + \omega 13)} & \cdot & \frac{2*(\psi in^2 + \omega 1n^2)}{(\psi in + \omega 1n)} \\ \frac{2*(\psi i1^2 + \omega 21^2)}{(\psi i1 + \omega 21)} & \frac{2*(\psi i2^2 + \omega 22^2)}{(\psi i2 + \omega 22)} & \frac{2*(\psi i3^2 + \omega 23^2)}{(\psi i3 + \omega 23)} & \cdot & \frac{2*(\psi in^2 + \omega 2n^2)}{(\psi in + \omega 2n)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{2*(\psi i1^2 + \omega m1^2)}{(\psi i1 + \omega m1)} & \frac{2*(\psi i2^2 + \omega m2^2)}{(\psi i2 + \omega m2)} & \frac{2*(\psi i3^2 + \omega m3^2)}{(\psi i3 + \omega m3)} & \cdot & \frac{2*(\psi in^2 + \omega mn^2)}{(\psi in + \omega mn)} \end{bmatrix}$$

where i= 1.....m

Now calculate G

$$G_{m\times m} =$$

$$\begin{bmatrix} H1(1.1) + H1(1.2) + \ldots H1(1.n) & H2(1.1) + H2(1.2) + \ldots H2(1.n) & \cdot & Hm(1.1) + Hm(1.2) + \ldots Hm(1.n) \\ H1(2.1) + H1(2.2) + \ldots H1(2.n) & H2(2.1) + H2(2.2) + \ldots H2(2.n) & \cdot & Hm(2.1) + Hm(2.2) + \ldots Hm(2.n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ H1(m.1) + H1(m.2) + \ldots H1(m.n) & H2(m.1) + H2(m.2) + \ldots H2(m.n) & \cdot & Hm(m.1) + Hm(m.2) + \ldots Hm(m.n) \end{bmatrix}$$

Order of G matrix is mxm:

Kernel matrix $K(\psi,\omega) = 1 - G$

In the proposed method, Authors have used kernel matrix $[\rho, K(\psi,\omega)]$ where $\rho = 1.....m$.

The authors have used the kernel matrix $[\rho, K(\psi,\omega)]$ to build the fall detection model using a one-class support vector machine. One-class SVM has one parameter 'nu' with $0 < nu <= 1$ which needs to be chosen in advance while performing one-class classification. In the experiment, the nu parameter is tuned to 0.01. Keeping the value of nu other than 0.01 gave the worst results and keeping the value of nu is 0.01 gave optimum results.

Testing - For testing, at first authors have calculated the kernel matrix between each pair of training matrix ψ and testing matrix τ. Suppose the testing matrix is:

$$\tau = [<a_1, b_1, c_1, d_1, e_1, f_1, g_1> <a_2, b_2, c_2, d_2, e_2, f_2, g_2> .....<a_l, b_l, c_l, d_l, e_l, f_l, g_l >]$$

where a, b, c, d, e, f, g are calculated features and m is the number of instances in τ. Then kernel matrix between the test matrix and train matrix will be $[\alpha, K(\tau, \psi)]$ where $\alpha=1.....m$. Order of kernel matrix for testing will be l x m where l is the number of rows in the test matrix and m is the number of rows in the training matrix. Prediction is done using this kernel matrix.

RESULTS AND DISCUSSION

Authors have classified ADL activities and fall activities using SVM based one-class classification (Schölkopf et al. 2000) with the proposed kernel. Authors have compared the performance of widely used kernels which are the polynomial kernel, Linear Kernel and RBF kernels with the proposed kernel. Authors have also compared the performance of the proposed method with two existing fall detection methods i.e. (Micucci et al. 2017) and (Medrano et al. 2017) which were developed using one-class classification methods on the same dataset.

Performance measures have been calculated as follows

True positive rate (TPR): Total Fall activities classified as fall
True Negative rate (TNR): Total ADL activities classified as ADL
False positive Rate (FPR): Total ADL activities classified as fall
False Negative rate (FNR): Total Fall activities classified as ADL

$$Sensitivity = \frac{TPR}{(TPR + FNR)}$$

$$Specificity = \frac{TNR}{(TNR + FPR)}$$

$$Accuracy = \frac{TPR + TNR}{(TPR + TNR + FPR + FNR)}$$

$$Precision = \frac{TPR}{(TPR + FPR)}$$

$$F1\ score = \frac{2TPR}{\left(2TPR + FPR + FNR\right)}$$

$$Geometric\ Mean\ (GM) = \sqrt{Sensitivity\ X\ Specificity}$$

$$Critical\ Success\ Index = \frac{TPR}{\left(TPR + FNR + FPR\right)}$$

$$Matthews\ correlation\ coefficient\ (MCC) = \frac{\left(TPR\ X\ TNR\right) - \left(FPR\ X\ FNR\right)}{\sqrt{\left(TPR + FPR\right)\left(TPR + FNR\right)\left(TNR + FPR\right)\left(TNR + FNR\right)}}$$

Bookmaker Informedness (BM) = Sensitivity + Specificity -1

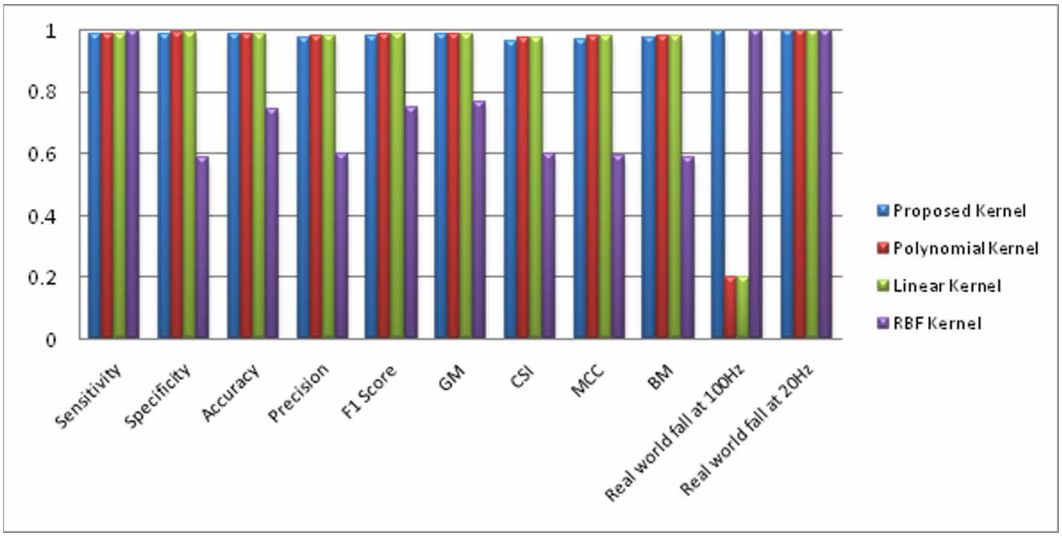
In the proposed method the ADL activities and fall activities were classified using SVM based one-class classification. The performance of widely used kernels such as the polynomial kernel, linear kernel, RBF kernel was compared along with the proposed kernel based one-class classification. The proposed method was also compared with the two existing fall detection methods (Micucci et al. 2017) and (Medrano et al. 2014) were developed using one-class classification methods on the same data set.

The performance of SVM based one-class classification using different kernels is shown in Table 1 and Figure 3. Different performance measures like sensitivity, specificity, accuracy, precision, F1 score. Geometric mean, critical success index(CSI), Matthews correlation coefficient (MCC), Bookmarker Informedness (BM) have been considered. Few performance measures like sensitivity, specificity, accuracy precision, F1 score, Geometric mean are common for both one-class classification (imbalance data) and multiclass classification(balanced data) but for imbalance data the performance

Table 1. Performance of SVM based one-class classifier to detect falls using different kernels

Performance measure	Proposed Kernel	Polynomial Kernel	Linear Kernel	RBF Kernel
Sensitivity	0.9881	0.9901	0.9901	1.0000
Specificity	0.9865	0.9902	0.9902	0.5858
Accuracy	0.9871	0.9901	0.9901	0.7437
Precision	0.9783	0.9842	0.9842	0.5981
F1 Score	0.9832	0.9871	0.9871	0.7485
Geometric Mean(GM)	0.9873	0.9901	0.9901	0.7654
Critical Success Index(CSI)	0.9669	0.9746	0.9746	0.5981
Matthews correlation coefficient (MCC)	0.9728	0.9791	0.9791	0.5919
Bookmaker Informedness (BM)	0.9746	0.9803	0.9803	0.5858
Falls correctly classified as fall in real-world fall dataset collect at 100Hz	100%	20%	20%	100%
Falls correctly classified as fall in world fall dataset collect at 20Hz	100%	100%	100%	100%

Figure 3. Performance of SVM based one-class classifier to detect fall using different kernels



measures like Critical success index(CSI), Matthews correlation coefficient (MCC), Bookmaker Informedness (BM) are used.

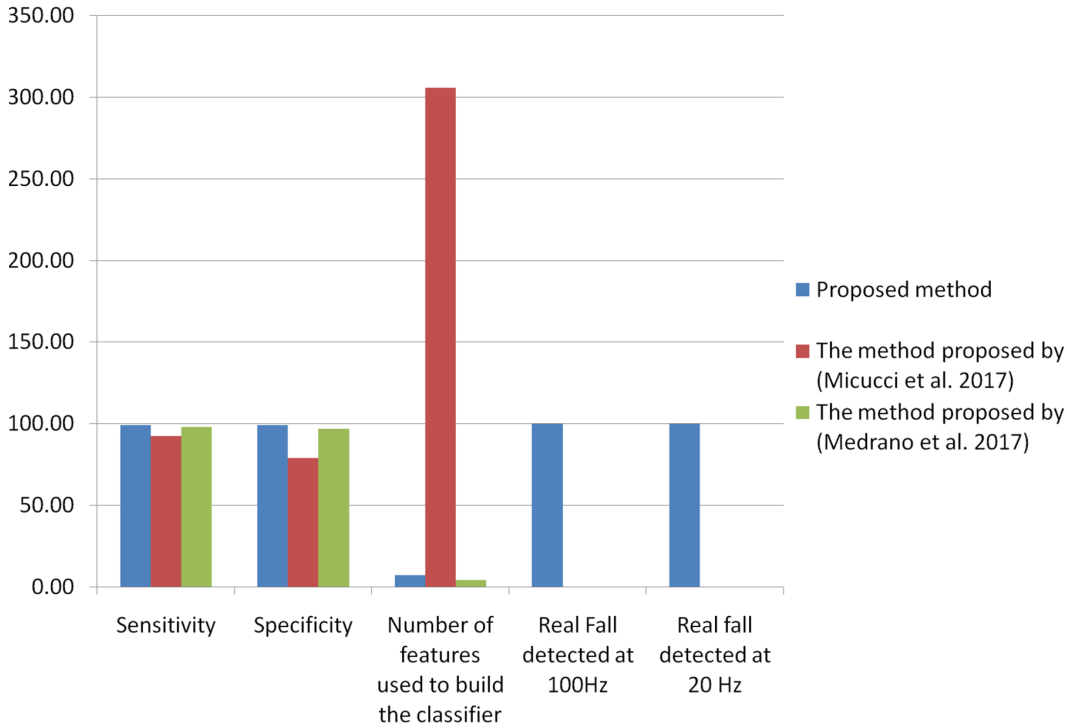
It can be seen that in terms of sensitivity RBF kernel performed better than other methods but when using RBF kernel other performance measures are poor. When comparing the performance of the proposed method with polynomial or linear kernel, there is no great difference between their performance but when liner or polynomial kernel is used to detect real fall events then their performance degraded drastically whereas proposed method performs exceptionally well on both simulated data and real-world fall data.

It can also be seen from Table 2 and Figure 4, that the proposed method performs better than two existing methods (Micucci et al. 2017) (Medrano et al. 2017). In method (Micucci et al. 2017) sensitivity is 92.2% (one class KNN) 92.5%(one-class SVM) and in method (Medrano et al. 2017) sensitivity is 97.9% whereas in proposed method authors got 98.81% sensitivity. In method (Micucci et al. 2017) specificity is 82.2%(one class KNN) 79.0%(one-class SVM) and in method (Medrano

Table 2. Comparisons of the proposed method with other fall detection methods on the same dataset

Performance measures	Proposed method	The method proposed by (Micucci et al. 2017)	The method proposed by (Medrano et al. 2017)
Sensitivity	98.81%	92.2%(one class KNN) 92.5%(one class SVM)	97.9%
Specificity	98.65%	82.2%(one class KNN) 79.0%(one class SVM)	96.7%
Number of features used to build the classifier	7	306	4
Falls correctly classified as fall in real-world fall dataset collected at 100Hz	100%	Not tested	Not tested
Falls correctly classified as fall in world fall dataset collected at 20Hz	100%	Not tested	Not tested

Figure 4. Comparison of the proposed method with other methods on the same dataset



et al. 2017) specificity is 96.7% whereas in proposed method authors got 98.65% specificity. It is a great improvement that authors have achieved this performance using only seven features whereas methods (Micucci et al. 2017) (Medrano et al. 2017) have used 306 and 4 features respectively.

Micucci et.al. (Micucci et al. 2017) have used two publically available datasets and used a 10-fold cross-validation approach for model building and testing. They built the model using ADL activities. Medrano et.al. (Medrano et al. 2017) have used only ADL activities to train the classifier and used fall activities to test classifier but authors have divided ADL activities dataset into 90:10 ratio and used 90% data for model building and testing was performed on rest 10% simulated activity of daily living data, simulated fall dataset and real-world fall dataset. It can be seen that the proposed method classifies 98.81% fall activities as fall activity which is better than other classifiers. In existing methods (Micucci et al. 2017) (Medrano et al. 2017), the authors tested their fall detection methods only on simulated fall data. Their methods were not verified on real fall data whereas the proposed method is tested on that and performed very well. The proposed fall detection model can be used for adult and elderly fall detection because it is trained on the (Medrano et al. 2014) dataset in which volunteer weight ranges between 54 to 98 kg, age 20 to 42 years and height from 1.61 to 1.81 meters and tested on both (Klenk et al. 2016a; Medrano et al. 2014) datasets. In (Klenk et al. 2016) dataset mean age was 69.5 years, mean height was 167 c.m. and mean weight was 75.59 kg.

CONCLUSION

Fall among elders is a major obstacle for independently living elderly people. It can cause serious health problems. Fall detection is a very complex process, for which many fall detection algorithms have been proposed. Most of the fall detection algorithms were designed using both ADL and fall data which were simulated in the lab. These models are less reliable to detect fall because actual

fall events are very much different from simulated fall. Inspired by eccentric learning, Support vector machine-based one-class classification to classify ADL activities and fall activities have been proposed here. This model was built using ADL data. For testing, both ADL data and fall data were used. In SVM based one-class classification, instead of using widely used kernel like linear kernel, polynomial kernel, RBF kernel, etc. Authors have proposed a new method to calculate kernel matrix and it is found that the proposed method can score 98.81% sensitivity and 98.65% specificity which is better than existing methods. The proposed model is also tested on actual falls instance and can detect 100% fall events. Existing methods, which were based on one class classification, were not tested on real-world fall events but it is necessary to validate the fall detection model on real-world fall data which is done in the proposed method.

ACKNOWLEDGMENT

We extend gratitude to all volunteers, scientists, managers, and people who are directly or indirectly involved in the FARSEEING project for making real fall data available to us to do real research on human fall detection to benefit society.

REFERENCES

Abbate, S., Avvenuti, M., Bonatesta, F., Cola, G., Corsini, P., & Vecchio, A. (2012). A Smartphone-Based Fall Detection System. *Pervasive and Mobile Computing*, *8*(6), 883–899. doi:10.1016/j.pmcj.2012.08.003

Adhikari, Bouchachia, & Nait-Charif. (2019). *Deep Learning Based Fall Detection Using Simplified Human Posture*. Academic Press.

Adnan, S. M., Irtaza, A., Aziz, S., Ullah, M. O., Javed, A., & Mahmood, M. T. (2018). Fall Detection through Acoustic Local Ternary Patterns. *Applied Acoustics*, *140*, 296–300. doi:10.1016/j.apacoust.2018.06.013

Ahmad, F., Boashash, B., Jokanovic, B., & Amin, M. (2016). *Radar Fall Detection Using Principal Component Analysis Structural Health Monitoring View Project Special Issue 'Radar Imaging in Challenging Scenarios from Smart and Flexible Platforms' on Remote Sensing View Project Branka Jokanovic Delphi Automotive Ra.* https://www.researchgate.net/publication/303032243.

Albert, M. V., Kording, K., Herrmann, M., & Jayaraman, A. (2018, November 30). Fall Classification by Machine Learning Using Mobile Phones. *PLoS One*, *7*(5), e36556. doi:10.1371/journal.pone.0036556 PMID:22586477

Amin, M. G., Zhang, Y. D., Ahmad, F., & Dominic Ho, K. C. (2016). Radar Signal Processing for Elderly Fall Detection: The Future for in-Home Monitoring. *IEEE Signal Processing Magazine*, *33*(2), 71–80. doi:10.1109/MSP.2015.2502784

Bishop, C. M. (1994). Novelty Detection and Neural Network Validation. *IEE Proceedings. Vision Image and Signal Processing*, *141*(4), 217–222. doi:10.1049/ip-vis:19941330

Boutellaa, E., Kerdjidj, O., & Ghanem, K. (2019). Covariance Matrix Based Fall Detection from Multiple Wearable Sensors. *Journal of Biomedical Informatics*, *94*, 103189. doi:10.1016/j.jbi.2019.103189 PMID:31029654

Chandra, Sivakumar, Gokulnath, & Parthasarathy. (2018). IoT Based Fall Detection and Ambient Assisted System for the Elderly. *Cluster Computing*. https://link.springer.com/10.1007/s10586-018-2329-2.

Chen, Y. (2019). A Dynamic Height Analysis on Vision Based Fall Detection System. In *International Conference on Advanced Mechatronic Systems, ICAMechS*. IEEE Computer Society. doi:10.1109/ICAMechS.2019.8861676

de Lima, S., & Lígia, A. (2019). Home-Based Monitoring of Falls Using Wearable Sensors in Parkinson's Disease. *Movement Disorders*. PMID:31449705

Dhiraj, . (2020). Activity Recognition for Indoor Fall Detection in 360-Degree Videos Using Deep Learning Techniques. In *Advances in Intelligent Systems and Computing* (pp. 417–429). Springer.

Ding, Q. (2019). RT-ADI: Fast Real-Time Video Representation for Multi-View Human Fall Detection. In *Proceedings - 2019 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2019*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICMEW.2019.00010

Droghini, D., Ferretti, D., Principi, E., Squartini, S., & Piazza, F. (2017). A Combined One-Class SVM and Template-Matching Approach for User-Aided Human Fall Detection by Means of Floor Acoustic Features. *Computational Intelligence and Neuroscience*, *2017*, 2017. doi:10.1155/2017/1512670 PMID:28638405

Erol, Baris, & Amin. (2018). *Radar Data Cube Analysis for Fall Detection Advanced Signal Processing and Emerging Sensing Technologies for Assisted Living View Project Special Issue 'Radar Imaging in Challenging Scenarios from Smart and Flexible Platforms' on Remote Sensing View Proje*. https://www.researchgate.net/publication/327805468

Fan, K., Wang, P., & Zhuang, S. (2019). Human Fall Detection Using Slow Feature Analysis. *Multimedia Tools and Applications*, *78*(7), 9101–9128. doi:10.1007/s11042-018-5638-9

Giuffrida, D., Benetti, G., De Martini, D., & Facchinetti, T. (2019). Fall Detection with Supervised Machine Learning Using Wearable Sensors. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. IEEE. https://ieeexplore.ieee.org/document/8972246/

Gjoreski, H. (2020). *Wearable Sensors Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition.* https://link.springer.com/10.1007/978-3-030-38748-8_4

Global QYResearch. (n.d.). *Fall Detection Systems Market Analysis – Global Industry Size, Share, Growth Opportunity, Trends and Forecast 2026 - MarketWatch*. https://www.marketwatch.com/press-release/fall-detection-systems-market-analysis-global-industry-size-share-growth-opportunity-trends-and-forecast-2026-2019-04-05

Greene, S., Thapliyal, H., & Carpenter, D. (2017). IoT-Based Fall Detection for Smart Home Environments. In *Proceedings - 2016 IEEE International Symposium on Nanoelectronic and Information Systems, INIS 2016*. Institute of Electrical and Electronics Engineers Inc.

Gu, Y. (2018). *Your WiFi Knows You Fall: A Channel Data-Driven Device-Free Fall Sensing System*. https://ieeexplore.ieee.org/abstract/document/8691181/

Guerbai, Y., Chibani, Y., & Hadjadji, B. (2015). The Effective Use of the One-Class SVM Classifier for Handwritten Signature Verification Based on Writer-Independent Parameters. *Pattern Recognition*, *48*(1), 103–113. doi:10.1016/j.patcog.2014.07.016

Horta, E. T., Lopes, I. C., Rodrigues, J. J. P. C., & Misra, S. (2013). Real Time Falls Prevention and Detection with Biofeedback Monitoring Solution for Mobile Environments. *2013 IEEE 15th International Conference on E-Health Networking, Applications and Services, Healthcom 2013*, 594–600. doi:10.1109/HealthCom.2013.6720746

Hsieh, Y.-Z., & Jeng, Y.-L. (2019, January 4). Development of Home Intelligent Fall Detection IoT System Based on Feedback Optical Flow Convolutional Neural Network. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 6048–6057. https://ieeexplore.ieee.org/document/8101471/. doi:10.1109/ACCESS.2017.2771389

Hsu, C. C. H. (2017). FallCare+: An IoT Surveillance System for Fall Detection. In *Proceedings of the 2017 IEEE International Conference on Applied System Innovation: Applied System Innovation for Modern Technology, ICASI 2017*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICASI.2017.7988590

Hussain, F., Hussain, F., Ehatisham-Ul-Haq, M., & Azam, M. A. (2019). Activity-Aware Fall Detection and Recognition Based on Wearable Sensors. *IEEE Sensors Journal*, *19*(12), 4528–4536. doi:10.1109/JSEN.2019.2898891

Igual, R., Medrano, C., & Plaza, I. (2015). A Comparison of Public Datasets for Acceleration-Based Fall Detection. *Medical Engineering & Physics*, *37*(9), 870–878. doi:10.1016/j.medengphy.2015.06.009 PMID:26233258

Irtaza, A. (2017). A Framework for Fall Detection of Elderly People by Analyzing Environmental Sounds through Acoustic Local Ternary Patterns. In *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/SMC.2017.8122836

Japkowicz, N. (1999). *Concept-Learning in the Absence of Counter-Examples: An Autoassociation-Based Approach to Classification*. https://www.researchgate.net/profile/Stephen_Hanson3/publication/2293736_Concept-Learning_In_The_Absence_Of_Counter-Examples_An_Autoassociation-Based_Approach_To_Classification/links/02e7e51b889e587d59000000.pdf

Kemmler, M., Rodner, E., & Denzler, J. (2011). One-Class Classification with Gaussian Processes. In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer. doi:10.1007/978-3-642-19309-5_38

Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., & Chouireb, F. (2020). Fall Detection and Human Activity Classification Using Wearable Sensors and Compressed Sensing. *Journal of Ambient Intelligence and Humanized Computing*, *11*(1), 349–361. doi:10.1007/s12652-019-01214-4

Khan, S., Qamar, R., Zaheen, R., Al-Ali, A. R., Al Nabulsi, A., & Al-Nashash, H. (2019). Internet of Things Based Multi-Sensor Patient Fall Detection System. *Healthcare Technology Letters*, *6*(5), 132–137. doi:10.1049/htl.2018.5121 PMID:31839969

Kim, I. et al.. (2019). Implementation of a Real-Time Fall Detection System for Elderly Korean Farmers Using an Insole-Integrated Sensing Device. *Instrumentation Science & Technology*.

Klenk, J., Schwickert, L., Palmerini, L., Mellone, S., Bourke, A., Ihlen, E. A. F., Kerse, N., Hauer, K., Pijnappels, M., Synofzik, M., Srulijes, K., Maetzler, W., Helbostad, J. L., Zijlstra, W., Aminian, K., Todd, C., Chiari, L., & Becker, C. (2018, November 30). 2016. "The FARSEEING Real-World Fall Repository: A Large-Scale Collaborative Database to Collect and Share Sensor Signals from Real-World Falls. *European Review of Aging and Physical Activity*, *13*(1), 8. doi:10.1186/s11556-016-0168-9 PMID:27807468

Krawczyk, B., & Filipczuk, P. (2014). Cytological Image Analysis with Firefly Nuclei Detection and Hybrid One-Class Classification Decomposition. *Engineering Applications of Artificial Intelligence*, *31*, 126–135. doi:10.1016/j.engappai.2013.09.017

Leng, Q. (2015). One-Class Classification with Extreme Learning Machine. *Mathematical Problems in Engineering*.

Lopes, I. C., Vaidya, B., & Rodrigues, J. J. P. C. (2009). SensorFall - An Accelerometer Based Mobile Application. *Proceedings of the 2009 2nd International Conference on Computer Science and Its Applications, CSA 2009.*

Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., & Li, Y. (2014). Depth-Based Human Fall Detection via Shape Features and Improved Extreme Learning Machine. *IEEE Journal of Biomedical and Health Informatics*, *18*(6), 1915–1922. doi:10.1109/JBHI.2014.2304357 PMID:25375688

Medrano, C., Igual, R., García-Magariño, I., Plaza, I., & Azuara, G. (2018, November 30). "Combining Novelty Detectors to Improve Accelerometer-Based Fall Detection. *Medical & Biological Engineering & Computing*, *55*(10), 1849–1858. doi:10.1007/s11517-017-1632-z PMID:28251444

Medrano, C., Igual, R., Plaza, I., & Castro, M. (2018, November 30). Detecting Falls as Novelties in Acceleration Patterns Acquired with Smartphones. *PLoS One*, *9*(4), e94811. doi:10.1371/journal.pone.0094811 PMID:24736626

Micucci, D., Mobilio, M., Napoletano, P., & Tisato, F. (2018, November 30). Falls as Anomalies? An Experimental Evaluation Using Smartphone Accelerometer Data. *Journal of Ambient Intelligence and Humanized Computing*, *8*(1), 87–99. doi:10.1007/s12652-015-0337-0

Min, W., Zou, S., & Li, J. (2019). Human Fall Detection Using Normalized Shape Aspect Ratio. *Multimedia Tools and Applications*, *78*(11), 14331–14353. doi:10.1007/s11042-018-6794-7

Musci, M. (2018). *Online Fall Detection Using Recurrent Neural Networks*. https://arxiv.org/abs/1804.04976

Narui, H., Shu, R., Gonzalez-Navarro, F. F., & Ermon, S. (2020). Domain Adaptation for Human Fall Detection Using WiFi Channel State Information. In *Studies in Computational Intelligence* (pp. 177–181). Springer Verlag.

Nguyen, T. H., Nguyen, T. T., & Ngo, B. V. (2020). A SVM Algorithm for Falling Detection in an IoTs-Based System. In *Intelligent Systems Reference Library* (pp. 139–170). Springer Science and Business Media Deutschland GmbH.

Noor, M. H. M., Salcic, Z., & Kai Wang, K. I. (2017). Adaptive Sliding Window Segmentation for Physical Activity Recognition Using a Single Tri-Axial Accelerometer. *Pervasive and Mobile Computing*, *38*, 41–59. doi:10.1016/j.pmcj.2016.09.009

Palipana & Rojas. (2017). *FallDeFi: Ubiquitous Fall Detection Using Commodity Wi-Fi Devices*. 10.1145/3161183

Pham, V. (2019). Multi-Sensor Data Fusion in A Real-Time Support System for On-Duty Firefighters. *Sensors, 19*(21), 4746. https://www.mdpi.com/1424-8220/19/21/4746

Popescu, M., & Mahnot, A. (2009). Acoustic Fall Detection Using One-Class Classifiers. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*. IEEE Computer Society.

Principi, E., Droghini, D., Squartini, S., Olivetti, P., & Piazza, F. (2016). Acoustic Cues from the Floor: A New Approach for Fall Classification. *Expert Systems with Applications*, *60*, 51–61. doi:10.1016/j.eswa.2016.04.007

Qiu, Z. (2020). *Old Man Fall Detection Based on Surveillance Video Object Tracking*. https://link.springer.com/10.1007/978-981-15-2767-8_15

Ritter, G., & Gallegos, M. T. (1997). Outliers in Statistical Pattern Recognition and an Application to Automatic Chromosome Classification. *Pattern Recognition Letters*, *18*(6), 525–539. doi:10.1016/S0167-8655(97)00049-4

Saleh, M., & Regine, L. B. J. (2019). Elderly Fall Detection Using Wearable Sensors: A Low Cost Highly Accurate Algorithm. *IEEE Sensors Journal*, *19*(8), 3156–3164. doi:10.1109/JSEN.2019.2891128

Santos, J., Rodrigues, J. J. P. C., Silva, B. M. C., Casal, J., Saleem, K., & Denisov, V. (2016). An IoT-Based Mobile Gateway for Intelligent Personal Assistants on Mobile Health Environments. *Journal of Network and Computer Applications*, *71*, 194–204. doi:10.1016/j.jnca.2016.03.014

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2018, November 30). New Support Vector Algorithms. *Neural Computation*, *12*(5), 1207–1245. doi:10.1162/089976600300015565 PMID:10905814

Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., & Summers, R. M. (2016). Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*, *35*(5), 1285–1298. doi:10.1109/TMI.2016.2528162 PMID:26886976

Su, Y., Liu, D., & Wu, Y. (2017). A Multi-Sensor Based Pre-Impact Fall Detection System with a Hierarchical Classifier. In *Proceedings - 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2016*. Institute of Electrical and Electronics Engineers Inc.

Subramanyan, Thathupara, Jang, Tsogtbaatar, & Cho. (2020). Fall Detection System for Elderly People Using Vision-Based Analysis. *Romanian Journal of Information Science and Technology, 23*.

Sztyler, T., Stuckenschmidt, H., & Petrich, W. (2017). Position-Aware Activity Recognition with Wearable Devices. *Pervasive and Mobile Computing*, *38*, 281–295. doi:10.1016/j.pmcj.2017.01.008

Tax, D. (2001). *One-Class Classification.* PhD thesis, Delft University of Technology. https://ci.nii.ac.jp/naid/10010287825/

Theodoridis, S., & Koutroumbas, K. (2018). Pattern recognition. *IEEE Transactions on Neural Networks*.

Tsai, T. H., Rui, Z. W., & Chin, W. H. (2019). Design of Fall Detection System Using Computer Vision Technique. *ACM International Conference Proceeding Series*, 33–37. doi:10.1145/3351180.3351191

Wan, M., Shang, W., & Zeng, P. (2017). Double Behavior Characteristics for One-Class Classification Anomaly Detection in Networked Control Systems. *IEEE Transactions on Information Forensics and Security*, *12*(12), 3011–3023. doi:10.1109/TIFS.2017.2730581

Wan, Q., Li, Y., Li, C., & Pal, R. (2014). Gesture Recognition for Smart Home Applications Using Portable Radar Sensors. *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, 6414–17.

Wang, J. (2014). An Enhanced Fall Detection System for Elderly Person Monitoring Using Consumer Home Networks. *IEEE Transactions on Consumer Electronics*, *60*(1), 23–29. doi:10.1109/TCE.2014.6780921

Yacchirema, D., De Puga, J. S., Palau, C., & Esteve, M. (2018). Fall Detection System for Elderly People Using IoT and Big Data. In *Procedia Computer Science* (pp. 603–610). Elsevier B.V. doi:10.1016/j.procs.2018.04.110

Yang, X., Xiong, F., Shao, Y., & Niu, Q. (2018). WmFall: WiFi-Based Multistage Fall Detection with Channel State Information. *International Journal of Distributed Sensor Networks*, *14*(10). Advance online publication. doi:10.1177/1550147718805718

Yu, M., Naqvi, S. M., Rhuma, A., & Chambers, J. (2011). Fall Detection in a Smart Room by Using a Fuzzy One Class Support Vector Machine and Imperfect Training Data. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 1833–36. doi:10.1109/ICASSP.2011.5946861

Yu, M., Naqvi, S. M., Rhuma, A., & Chambers, J. (2020, January 22). One Class Boundary Method Classifiers for Application in a Video-Based Fall Detection System. *IET Computer Vision*, *6*(2), 90. doi:10.1049/iet-cvi.2011.0046

Yu, M., Rhuma, A., Naqvi, S. M., & Chambers, J. (2011). Fall Detection for the Elderly in a Smart Room by Using an Enhanced One Class Support Vector Machine. *17th DSP 2011 International Conference on Digital Signal Processing, Proceedings.* doi:10.1109/ICDSP.2011.6004881

Zhang, T., Wang, J., Xu, L., & Liu, P. (2006). Fall Detection by Wearable Sensor and One-Class SVM Algorithm. In *Intelligent Computing in Signal Processing and Pattern Recognition*. Springer. https://link.springer.com/10.1007/978-3-540-37258-5_104

Zhang, T., & Zhang, T. (2006). Fall Detection by Embedding an Accelerometer in Cellphone and Using Kfd Algorithm. *International Journal of Computer Science and Network Security*. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.557.7053

Zhou, X. (2018). Fall Detection Using Convolutional Neural Network with Multi-Sensor Fusion. In *2018 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2018*. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICMEW.2018.8551564

Rashmi Shrivastava received her master's degree in Computer Applications from Govt. Engineering college, Jabalpur(M.P.) She is a research scholar at the Department of Computer Applications, National Institute of Technology Raipur, Chhattisgarh, India and assistant Professor at Dept of CSA, Atal Bihari Vajpayee University, Bilaspur, Chhattisgarh, India. Her research interests include human activity recognition and machine learning.

Manju Pandey received her PhD in Computer Science & Engineering from MANIT Bhopal (MP) India. Her research interests include Fuzzy Multi-criteria decision making and Internet of Things (IoT) and Cloud Computing . Currently, she is an Assistant Professor at the Department of Computer Applications, National Institute of Technology Raipur, Chhattisgarh, India.