Hittite Journal of Science and Engineering, 2021, 8 (3) 197-205
ISSN NUMBER: 2148-4171
DOI: 10.17350/HJSE19030000230

Elderly Fall Detection and Fall Direction Detection via Various Machine Learning Algorithms Using Wearable Sensors

Sitki Kocaoglu[1] **Yilmaz Guven**[2]
[1]Ankara Yıldırım Beyazıt University, Department of Biomedical Engineering, Ankara, Turkey
[2]Kırklareli University, Department of Electronic & Automation, Kırklareli, Turkey

ABSTRACT

The world population is aging rapidly. Some of elderly person live alone and it is observed that the elderly who live with their families frequently have to stay at home alone. Especially during the working hours of adult members of the family, elderly member stays alone. Falling while alone at home often results in fatal injuries and even death in elderly individuals. Fall detection systems can detect falls and provide emergency healthcare services in a short time. In this study, a two-step fall detection and fall direction detection system has been developed by using a public dataset with 5 different machine learning algorithms comparatively. If a fall is detected in the first stage, the second stage is started to determine the direction of the fall. In this way, the direction of fall can be determined for elderly individual to be used in future researches, and an early warning system that enables necessary measures such as opening an airbag in the direction of the fall can be developed. Thus, a gradual fall and its direction detection system has been developed by determining the best classifying algorithms. As a result, it has been determined that Ensemble Subspace k-Nearest Neighbor (ES k-NN) classifier performs a little more successful classification compared to other classifiers. The classification by using the test data which is corresponding to 30% of the total data that was never used during the training phase, has been performed 99.4% accuracy for fall detection, and 97.2% success has been achieved in determining the direction of falling.

Article History:
Received: 2021/02/02
Accepted: 2021/07/02
Online: 2021/09/29

***Correspondence to**: Sıtkı Kocaoğlu*
Department of Biomedical Engineering,
Ankara Yıldırım Beyazıt University, Ankara,
Turkey
Tel: +90 3129062309
E-Mail:sitki.kocaoglu@hotmail.com

Keywords:
Fall Detection; Falling Direction Detection; Machine Learning; Wearable Sensors

INTRODUCTION

As a result of the developments in the field of medicine, human life has been prolonged. For this reason, the ratio of elderly individuals to other individuals in the society is increasing day by day. According to the United Nations' World Population Ageing 2019 report, the world population is estimated to increase by 2 billion people and reach 9.7 billion by 2050(United Nations, 2019). According to the expectations of the World Health Organization (WHO), the number of individuals over 60 is estimated to exceed 2 billion in 2050 and the rate of deaths due to falls is estimated to increase from 28% to 42% (WHO, 2007; Chelli and Patzold, 2019). While falls of elderly people sometimes result in severe damage such as bone fracture, some falls do not cause any significant injury. The elderly person's health condition worsens as a result of the inability to get up without support after a long stay on the floor. For this reason, it is vital that the falls of elderly individuals must be detected while alone at home to provide necessary medical support. Fall detection systems are subjects of great interest for researchers since 2010. Fall detection systems can be examined in 3 main groups as; (i) Wearable Sensor Based Fall Detection Systems, (ii) Ambient Sensor and Image Processing Based Fall Detection Systems and (iii) Multimodal Fall Detection Systems (Kocaoglu, 2020). Wearable Sensor Based systems generally include accelerometers and gyroscopes in their structures. These wearables, which are connected to various parts of the elderly person's body, measure acceleration and angular velocity. Acceleration and angular velocity information collected within a certain period is processed by using various methods and it is determined whether the individual's instantaneous movement is a fall or Activity of Daily Living (ADL).

There are many studies in the literature that use the SisFall dataset to detect a fall. While some of these studies try to specifically determine the type of fall, some of them make a determination of whether there is a fall or no fall in binary. However, the important thing in daily life is to quickly detect a fall and to determine the direction of the fall and to take precautions against hitting the hard floor if possible. The main goal here is to perform fall detection and fall direction detection quickly and with high accuracy. In this study, first fall detection is performed, then the fall direction is determined if a fall is detected in first stage. This is first in the literature by using the SisFall dataset. Thus, protective measures such as deploying an airbag in the direction of fall can be developed in future studies.

The following parts of the article are structured as follows. In Chapter 2, previous studies using the SisFall dataset are reviewed. In Chapter 3, the dataset is introduced and the method used in this study is explained. In Chapter 4, the results of the study are examined. In Chapter 5, a conclusion is made.

RELATED WORK

It appears that there are many Wearable Sensor Based Fall Detection Systems in the literature (Chen et al., 2005; Kang, Yoo and Kim, 2006; Srinivasan et al., 2007; Li et al., 2009; Delgado-Escaño et al., 2020; Kerdjidj et al., 2020; Nho, Lim and Kwon, 2020). Most of the studies in this group perform fall detection with very high accuracy. Studies in the Ambient Sensor and Image Processing Based Fall Detection Systems group usually detect falls by placing a camera in the elderly individual's room (Wu, 2000; Nait-Charif and McKenna, 2004; Han et al., 2013; Mastorakis and Makris, 2014; Daga, Ghatol and Thakare, 2018; Adhikari, 2019; Gupta et al., 2020). However, it is known that systems working with this method are rejected by elderly individuals. The reason for this is the violation of privacy and the fact that these systems can only work within a certain area and cannot detect falls that occur outside the building or in another room of the house. On the other hand, Multimodal Fall Detection Systems are methods that aim to increase accuracy by using information from both wearable sensors and cameras. (Nyan et al., 2006; de Assis Neto et al., 2019; Martinez-Villasenor, Ponce and Perez-Daniel, 2019). Fig. 1 shows the working structure of fall detection systems in general.

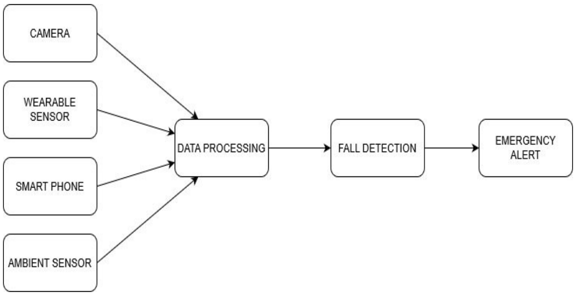

Figure 1. General block diagram of fall detection systems

To develop a fall detection system, collected data during past falls are needed. Since falls are very rare cases compared to ADLs, it is not possible to wait for real falls and record at the time of the fall. For this reason, data of artificial falls performed by the subjects are used during the development of fall detection systems. Although it is not possible to imitate real falls by the subjects, it is possible to obtain realistic results with this method. Most of the researchers present the fall data they collect during their studies for the use of other researchers as a public dataset. These public datasets differ in terms of the number of subjects, the number and variety of falling movements and ADLs, the number of repetitions of the movements and the sensor types. What is common to all of this dataset is that the fall experiments performed only by adult subjects. Since it is risky to perform a fall test with elderly subjects, there is no study that collects and shares fall data from individuals over the age of 65. Although there are differences between the fall dynamics of young people and elderly people, it is thought that this does not significantly affect the success of the system. The SisFall dataset shared publicly in 2017 is one of the most detailed datasets in the literature (Sucerquia, López and Vargas-Bonilla, 2017). This dataset was created with data from 2 accelerometers and 1 gyroscope. Most of the studies using this dataset do the fall recognition process as binary (fall/ADL) (Putra and Vesilo, 2018; Putra et al., 2018; Saleh and Le Bouquin Jeannès, 2018; Sucerquia, López and Vargas-Bonilla, 2018; Cho and Yoon, 2019; Saleh and Jeannes, 2019; Casilari, Lora- rivera and García-lagos, 2020; Han et al., 2020). Most studies used only accelerometer data. Among them, Yacchirema et al. used only 3 falling activities and 9 ADLs and made the classification in a way to determine the type of fall. In the study using Decision Tree (DT), an accuracy of about 92% was obtained (Yacchirema et al., 2018). In Liu et al.'s study, the effect of sampling rate on success was investigated and classification was made as binary fall/non-fall (Liu et al., 2018). Hussain et al. first classified the movements as binary fall/ADL, and if the result was fall, they tried to find out which type of fall it was. All fall types were included in this study, and the effects of sensor types were investigated by using them individually or together. As a result, the highest classification success was obtained using the Random Forest (RF) classifier when all sensors were used (Hussain et al., 2019). Some researchers used multiple datasets, trained using one dataset and tested by using the other one (Kruptizer et al., 2018; Delgado-Escaño et al., 2020).

In this study, 34 types of ADLs and falls are classified as binary using SisFall dataset. Then the direction of the fall is determined for the movements identified as falling. Thus, an approach suitable for obtaining information about how the elderly individual made falls and what their movements are before and after the fall is developed. Thus, a classifier is created that can be used in studies such as preventing falls, taking precautions and giving a warning in case of a fall possibility based on previous activities. The classification is performed with 5 different machine learning algorithms and the most successful classification results are obtained with ES k-NN classifier in both stages. In the classification using the 30% (testing) part of the data, which is not used in the training phase, the fall detection is performed with 99.4% accuracy, while the direction of the falls is detected correctly at a rate of 97.2%. These results coincide with the results of studies in the literature that detect falling binarily. As far as we know, this study is the first study in the literature that aims to determine the direction of falling and it does this with a high accuracy rate.

METHODS

The proposed method in this study is a two-step method. First step is determining whether the activity is a falling or not. When the real-time application is started, the information received from the sensors will be processed in short window intervals and if it indicates a fall, the direction of the rapid fall will be determined. Thus, it will be possible to take a precaution for the falling elderly person. The block diagram of the method is given in Fig. 2.

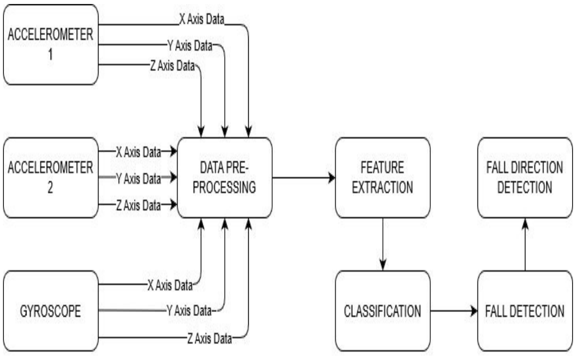

Figure 2. Block diagram of the system

Dataset

In the literature, there are public datasets containing various sensor and camera data prepared by different researchers. It is seen that almost all of these datasets consist of artificial falls and ADLs performed by young subjects (Casilari, Santoyo-Ramón, & Cano-García, 2017; Riquelme, Espinoza, Rodenas, Minonzio, & Taramasco, 2019). The body dynamics of the elderly and young individuals show great differences during the activities. However, it is not possible to perform the fall experiments by elderly subjects. So, it is a necessity to use the data obtained from the younger subjects. Likewise, there are serious dynamic differences between the real falls that occur in daily life and the artificial falls performed by the subject who knows that he fakes a falling movement. However, it seems impossible to create a dataset consisting of spontaneous falls, as the frequency of falling spontaneously in daily life is not high. In the evaluation of a dataset to be used in the detection of falls under these conditions, it is necessary to pay attention to the following.

- Possible movements should be done by elderly people. The extremity movement speeds of the elderly and young people show great differences while doing their daily living activities. If the dataset is created with young subjects and the system is used on elderly individuals, the accuracy of fall detection will decrease.

- The type of movements should be diverse in a way that matches the daily life as much as possible. while creating the dataset, if the selected movements are not frequently used in daily life, the algorithm that gives good results in the computer environment may result erroneous inferences for real-life.

- The number of repetitions for the movements in the dataset should be high. As a result, the success of the created system is directly proportional to the quality of the dataset. The more diverse and multiple trials the dataset contains, the higher the classification success of the movements will be.

- In order to increase the success in terms of classification, the dataset should be balanced both as ADL/Fall and for each ADL and fall. Because a well-balanced training set eliminates bias of the classifier toward particular class due to over-representation or under-representation of input patterns belonging to those classes (Nath & Subbiah, 2018).

- The movements should be done by using various individuals in terms of gender, height and weight. Because the physiological structure of a person significantly affects the speed and acceleration of movement.

- It should be ensured that the subject is not affected of the negative physiological effects that may occur as a result of the fall, especially when falling movements are made, but to ensure this, the measures to be taken in the experimental area must be in a way that does not disturb the characteristics of the fall.

When the public datasets are examined under these conditions, it is seen that SisFall dataset is one of the most comprehensive datasets available in the literature. Sucerquia et al. introduced SisFall dataset with their publication in 2017 (Sucerquia, López and Vargas-Bonilla, 2017).

Table 1. Activities

Code	Activity	Duration (s)
F01	Fall forward while walking, caused by a slip	15
F02	Fall backward while walking, caused by a slip	15
F03	Lateral fall while walking, caused by a slip	15
F04	Fall forward while walking, caused by a trip	15
F05	Fall forward while jogging, caused by a trip	15
F06	Vertical fall while walking, caused by fainting	15
F07	Fall while walking with damping, caused by fainting	15
F08	Fall forward when trying to get up	15
F09	Lateral fall when trying to get up	15
F10	Fall forward when trying to sit down	15
F11	Fall backward when trying to sit down	15
F12	Lateral fall when trying to sit down	15
F13	Fall forward while sitting, caused by fainting	15
F14	Fall backward while sitting, caused by fainting	15
F15	Lateral fall while sitting, caused by fainting	15
D01	Walking slowly	100
D02	Walking quickly	100
D03	Jogging slowly	100
D04	Jogging quickly	100
D05	Walking upstairs and downstairs slowly	25
D06	Walking upstairs and downstairs quickly	25
D07	Slowly sit and get up in a half-height chair	12
D08	Quickly sit and get up in a half-height chair	12
D09	Slowly sit and get up in a low-height chair	12
D10	Quickly sit and get up in a low-height chair	12
D11	Sitting, trying to get up, and collapse into a chair	12
D12	Sitting, lying slowly, wait a moment, and sit again	12
D13	Sitting, lying quickly, wait a moment, and sit again	12
D14	Changing position while lying (back-lateral-back)	12
D15	Standing, slowly bending at knees, and getting up	12
D16	Standing, slowly bending w/o knees, and getting up	12
D17	Standing, get into and get out of a car	25
D18	Stumble while walking	12
D19	Gently jump without falling (to reach a high object)	12

A large number of researchers using this dataset have developed various methods and classifiers that detect the fall. Few articles have threshold-based fall detection as in Sucerquia et al. (Sucerquia, López and Vargas-Bonilla, 2018; Jung et al., 2020). In these studies, the fall detection was made binary as fall/ADL.

This dataset contains data from two 3-axis accelerometers and one 3-axis gyroscope. The dataset, which consists of 38 participants, consisting of young and old individuals, includes 15 different falling movements and 19 types of ADLs. It is seen that almost every movement was repeated 5 times by the participants. While creating the dataset, the sensor information was recorded with frequency of 200 Hz. The dataset content is shown in Table 1.

Pre-processing

Pre-processing is an important step that directly affects the success of the classification. When the SisFall dataset is examined structurally, it is a smooth dataset that does not require much preprocessing. No noisy results were obtained thanks to low frequency data collection. In this study, only the activity duration was equalized in the preprocessing stage. All activities are divided into 12-second windows. Thus, activities such as walking and jogging were divided into parts and 8 separate samples were created from each trial.

Feature Extraction

In the next step, feature extraction is applied to extract more meaningful data from raw data. Thus, better characterization of each activity is provided. While choosing the features, the literature was examined and, the features that were used more frequently in previous studies during fall detection were determined. By examining the features in terms of computational complexity, the ones with high contribution/computational cost ratio were preferred. In Table 2, these features are given with their explanations. Here, μ present the mean value, N presents the number of elements, x presents the ith element, σ presents the standard deviation, s presents the sum, $g_2$presents the kurtosis, $g_1$presents the skewness and rms presents the root mean square. The features are extracted from each three axes of each three sensors, so that a total of 99 features for each observation are obtained.

Finally, the extracted features are brought to a common scale and normalized between -1 and 1. Thus, it was tried to take into account the small differences in the data depending on the used algorithm.

Fall Detection

In general, the movement speed of the body of the individual increases suddenly during a fall. This situation makes it possible to detect the fall by creating strong changes on the accelerometer connected to the individual's body. However, there are also movements with velocity changes close to the falling state in ADLs. This confusion is the main shortcoming of fall detection systems that are intended to be made only with accelerometer data. Therefore, in this study, gyroscope sensor is used together with accelerometer in order to detect angular change.

During the fall detection phase, binary classification is made, observations are divided into two groups Fall

Table 2. Features

Feature No	Feature	Equation/Definion	Feature No	Feature	Equation/Definion
1	Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$	7	*Minimum*	*Min. term of data*
2	Standart deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$	8	*Kurtosis*	$g_2 = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^4}{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$
3	Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$	9	*Skewness*	$g_1 = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^3}{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$
4	*Sum*	$s = \sum_{i=1}^{N} x_i$	10	*Band power*	*Average power of data*
5	*Median*	*Midterm of data*	11	*Root mean square*	$rms = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
6	Maximum	*Max. term of data*	12		

(Class 1) and ADL (Class 0). Thus, 3271 observations for Class 1 and 2440 observations for Class 0 are obtained. Classification is made using five different machine learning algorithms; Support Vector Machine (SVM), k Nearest Neighbor (k-NN), DT, RF and ES k-NN.

An SVM algorithm tries to obtain the hyperplane that best distinguishes between classes. This hyperplane is achieved by obtaining the largest margin between the classes. SVM is a computer algorithm that learns by example to assign labels to objects. At its core, an SVM is a mathematical entity and an algorithm for maximizing a particular mathematical function relative to a particular collection of data. There are 4 basic concepts at the core of the SVM classification; (i) separating hyperplane, (ii) hyperplane with maximum margin, (iii) soft margin, and (iv) kernel function. The general term for a straight line in a high-dimensional space is a hyperplane, and thus the dividing hyperplane is essentially the line that separates the differently labeled instances. If we define the distance from the separating hyperplane to the nearest expression vector as the margin of the hyperplane, the SVM chooses the maximum margin that separates the hyperplane. Choosing this particular hyperplane maximizes SVM's ability to predict the correct classification of previously unseen samples. Of course, many real data sets cannot be separated so clearly. Intuitively, it is desirable that the SVM could deal with errors in the data by allowing several anomalous expression profiles to fall on the wrong side of the separation hyperplane. To handle such cases, the SVM algorithm needs to be modified by adding a "soft margin". Essentially, this allows some data points to travel along the margin of the separation hyperplane without affecting the final result. The problem is that a single point cannot separate the two classes and adding a soft margin won't help. The kernel function provides a solution to this problem by adding an additional dimension to the data. In essence, the kernel function is a mathematical trick that allows SVM to perform a "two-dimensional" classification of a set of original one-dimensional data. In general, a kernel function projects data from a low-dimensional space to a higher-dimensional space by squaring the sample (Noble, 2006).

K nearest neighbors (kNN) is an efficient lazy learning algorithm and has successfully been developed in real applications. It is an algorithm that classifies according to both the distance measure and the number of neighbors. kNN algorithm computes the distance between each training sample and test samples in the dataset and then returns k closest samples. Its time complexity is linearly and is guaranteed to find exact k nearest neighbors (Deng et al., 2016).

By its simplest description, decision tree analysis is a divide-and-conquer approach to classification and regression. Decision trees can be used to discover features and extract patterns in large databases that are important for discrimination and predictive modeling. A decision tree is constructed by recursively partitioning the feature space of the training set. The first cell of the decision tree is called the root. Each observation is labeled yes or no according to the situation at this root. There are nodes under the roots. The complexity of the model increases as the number of nodes increases. The leaves at the bottom of the decision trees give the classification result. The objective is to find a set of decision rules that naturally partition the feature space to provide an informative and robust hierarchical classification model (Myles et al., 2004).

Random Forest is an ensemble method, which constructs many decision trees that will be used to classify a new instance by the majority vote. Each decision tree node uses a subset of attributes randomly selected from the whole original set of attributes. Additionally, each tree uses a different bootstrap sample data in the same manner as bagging. A Random Tree is a tree drawn at random from a set of possible trees, with m random attributes at each node. The term “at random” means that each tree has an equal chance of being sampled. Random Trees can be efficiently generated, and the combination of large sets of Random Trees generally lead to accurate models (Oshiro et al., 2012).

ES-KNN is also ensemble classifier like RF. Its difference from RF is that it uses KNN as a classifier model. Despite its simplicity, k-NN gives competitive results and even outperforms other complex learning algorithms in some cases. However, k-NN is affected by non-informative features in the data, which are quite common in high-dimensional data. Subspace ensembles have the advantage of using less memory than ensembles with all estimators and can handle missing values. For this reason, it is expected that the classification success of the ES-kNN algorithm will increase, especially as the dataset grows (Psathas et al., 2020).

Undoubtedly, more training data is needed to obtain a classifier model with good performance. Therefore, except for some special cases (limited data set), it is desirable that the training data set be larger than the test dataset. In fact, there is no optimum split ratio. It is generally accepted in the literature to divide the data sets as 70-80% training and 20-30% testing. It was predicted that separating the training set by 60% would reduce the classifier performance by 70%. When we separate the training data set by 80%, the performance of the classifier model will increase, but the test data will not be able to represent a part of the training data. Apart from these, considering the training and evaluation costs, the data set was divided into 70% training and 30% testing.

Standard machine learning metrics are used for evaluation. Learning outcomes are included for each observation in one of the following categories:

True Positive (TP): There is actually a fall and the classifier detect it as a fall

True Negative (TN): Actually, it is not a fall and the classifier detect it as not-fall

False Positive (FP): There is no actual fall and the classifier has a false fall alarm.

False Negative (FN): There is actually a fall and the classifier has incorrectly detected it as a not-fall.

Table 3. Model evaluation metrics for machine learning

Feature No	*Equation*
Sensitivity	$SE=\frac{TP}{TP+FN}x100$
Specificity	$SP=\frac{TN}{TN+FP}x100$
Accuracy	$ACC=\frac{TP+TN}{TP+TN+FP+FN}x100$
Precision	$PRE=\frac{TP}{TP+FP}x100$
F1-Score	$F1=2x\frac{PRE\,x\,SE}{PRE+SE}$

In order to measure the classification success of machine learning algorithms, a 2x2 confusion matrix is created and the samples are placed in the relevant section of the matrix by examining their real tags and classified tags. The components of the complexity matrix are the above-mentioned TP, TN, FP and FN. Performance parameters are calculated using these values. In this study, Sensitivity, Specificity, Accuracy, Precision and F1 score were used as performance parameters as in similar previous studies. Sensitivity (SE), which is one of the two most important metrics, is sometimes referred to as recall; It is used to calculate the rate at which the falls can be correctly detected. Another important parameter is Specifity (SP) that measures the success of the system in not giving false alarms. In addition, the value of accuracy (ACC) is the parameter that enables the classification result to be scored overall (Haq et al., 2018). The precision parameter serves to determine reliability of the model by checking whether a fall is existing when the model gives a fall alert. Real success rate of the system is obtained with the F1 score to be obtained from two parameters; precision and recall (Kocaoğlu and Akdoğan, 2019). All parameters used in performance evaluation are given in Table 3.

Falling Direction Classification

The main purpose of determining the fall of the elderly is to eliminate the negative effects of physiological and mental disorders that may occur after a fall as soon as possible. Therefore, when a fall is detected, the relevant

Table 4. Fall direction categories

Falling Direction	*Activity*
Forward	*F01-F04-F05-F07-F08-F10-F13*
Backward	*F02-F11-F14*
Lateral	*F03-F09-F12F15*
Vertical	*F06*

people are immediately informed and urgent medical support is requested. However, another area where fall detection systems can be used is to detect the falling direction of the individual and to eliminate the possibility of contact with the floor or other hard objects in the relevant direction. It is seen that researchers have used airbag systems to eliminate this impact contact in the past (Jung et al., 2020).

15 types of falling activities included in the SisFall dataset are classified as falling forward, backward and sideways (lateral) according to their directions as in Table 4. Only F06 (vertical fall while walking caused by fainting) defines the fainting and clutter drop, which can continue in any direction. Therefore, it is considered as a separate group. Again, classification is made using same machine learning algorithms. Each algorithm has been evaluated using machine learning performance parameters.

Testing Phase

At the end of the training phase, the most successful classifier model was created separately for both two stages. The system success was measured again by using 30% of the data allocated for testing and never seen by the classifiers. Testing process is carried out by using compact models of classifiers created during training phase. The success of the system is evaluated based on the results obtained at the end of this stage.

RESULTS AND DISCUSSION

The approach in this study consists of two stages: fall detection and falling direction detection. In k-folds cross validation, the relevant part of the dataset is divided into equal k parts. At each step, k-1 groups are used for training and performance is tested with the last group. This verification process is repeated k times. The performance of the classifier is calculated based on the k results (Haq et al., 2018). Choosing k at different values creates variation in classification performance. In this study, in both stages, 10 folds cross validation is used to re-check the results, since the best result was obtained in the case of k = 10.

Fall Detection Results

At this stage, each observation in the train data is classified as fall or ADL. Table 5 gives the results of all classifiers. At this stage, the classification success of all classifiers gave similar results. Although DT lags behind other classifiers by 4% in terms of accuracy, it approached the classification success of similar studies in the literature with an accurate classification rate of 94,4%. ES k-NN achieved a success rate of 99% and produced a similar F1 score. As such, binary classification in the fall detection stage has a high degree of accuracy.

Falling Direction Detection Results

At this stage, performance parameter results were obtained by testing with each classifier (Table 6). When the fall direction detection results are examined, it is seen that the ES-kNN algorithm is successful against other algorithms, although with a slight difference, as in the previous stage. Again, DT lags slightly behind other classifiers. Although the accuracy value of ES-kNN was 96,8%, the F1 score fell slightly behind this and became 93,8%. This is because the precision value is slightly lower than the accuracy value. This indicates that FPs are lower than FNs. Since the falling direction is being determined at this stage, this is due to the fact that the number of samples entering the random training set for each direction is not exactly equal.

Table 5. Performance comparison of various classifiers during fall detection phase

Classifier	*Type*	*SE*	*SP*	*ACC*	*PRE*	*F1*
DT	*Fine*	94,9	93,8	94,4	95.3	95.1
SVM	*Cubic*	98,1	98,1	98.1	98.6	98.3
KNN	*Fine*	98,9	98,6	98.7	98.9	98.9
RF		98,8	98,3	98.6	98.7	98.7
Ensemble	*Subspace k-NN*	98,9	99,0	98.9	99.2	99.1

Testing Results

It is seen that models created in previous studies are generally not tested. In order to test the reliability of the classification, the classification was made using the testing data, which was not used before, on the model. The results of this classification also coincide with the results of the training classification (Table 7).

CONCLUSION

In this study, the fall direction detection system is studied for elderly individuals. Direction detection creates a gap in the literature. It is seen that existing studies either detect falls binarily (fall/ADL) using various methods, or they try to determine exactly what falls and ADLs are. Binary fall detection success is observed to be very high in these studies, but success is low when activity detection is targeted. However, what is important in detecting

Table 6. Performance comparison of various classifiers during fall direction detection phase

Classifier	*Type*	*SE*	*SP*	*ACC*	*PRE*	*F1*
DT	*Fine*	86.4	95.0	92.7	86.1	86.1
SVM	*Cubic*	91.8	97.2	95.8	91.8	91.8
KNN	*Fine*	93.9	97.2	96.3	92.3	93.1
RF		94.2	97.2	96.4	92.0	93.1
Ensemble	*Subspace k-NN*	94.9	97.5	96.8	92.8	93.8

Table 7. The testing results of ES k-NN Model

Stage	*SE*	*SP*	*ACC*	*PRE*	*F1*
Fall Detection	*99.9*	*98.7*	*99.4*	*99.1*	*99.5*
Falling Direction Detection	*96.4*	*98.1*	*97.2*	*94.6*	*94.6*

a fall is not the reasons such as stumbling, fainting or losing balance, or head height during the fall, whether the knees touch the ground or not. The important detail in fall detection is to determine the direction of the fall and to prevent from hitting hard if it is possible. For this purpose, SisFall dataset, a very detailed dataset, was used in this study. Data belonging to two accelerometers and one gyroscope were used to create the processed dataset and a two-stage fall and fall direction detection system was developed. A total of 99 features are obtained with 11 various features extracted for three axes of each sensor and these features are used in both stages. As a result of the experiments made with 5 different machine learning algorithms, it was seen that the most successful classifier in both stages is ES k-NN. This classifier showed 98.9% accuracy in the fall detection stage and 96.8% accuracy in the fall direction detection stage. Similar results were obtained in the classification experiment using testing data corresponding to 30% of the total data, which were never used during the training on the ES k-NN model prepared.

In future studies, it can be tried to increase the classification success by using different techniques. By using different datasets together or using one of them with training and testing the other, a system can be developed that can be surer of its success when it is put into practice. While doing this study, our main goal was to develop a system that can predict the fall direction with high accuracy and take precautions against hitting the ground. We hope that in the next stages, airbags will be deployed in the determined direction and trials will be conducted with volunteers.

CONFLICT OF INTEREST

Authors approve that to the best of their knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

AUTHOR CONTRIBUTION

All sections including conceptualisation, methodology, software, analysis, writing, review and editing were equally organised and performed by Sitki Kocaoglu and Yilmaz Guven.

REFERENCES

1. Adhikari, K. (2019) 'Computer Vision Based Posture Estimation and Fall Detection', p. 131.
2. de Assis Neto, S. R. et al. (2019) 'Detecting Human Activities Based on a Multimodal Sensor Data Set Using a Bidirectional Long Short-Term Memory Model: A Case Study', in Ponce, H. et al. (eds) Challenges and Trends in Multimodal Fall Detection for Healthcare. Cham, Switzerland: Springer, pp. 31–51. doi: https://doi.org/10.1007/978-3-030-38748-8.
3. Casilari, E., Lora-rivera, R. and García-lagos, F. (2020) 'A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets', Sensors (Switzerland), 20(5). doi: 10.3390/s20051466.
4. Casilari, E., Santoyo-Ramón, J. A., & Cano-García, J. M. (2017). Analysis of public datasets for wearable fall detection systems. Sensors (Switzerland), 17(7). https://doi.org/10.3390/s17071513.
5. Chelli, A. and Patzold, M. (2019) 'A Machine Learning Approach for Fall Detection and Daily Living Activity Recognition', IEEE Access, 7, pp. 38670–38687. doi: 10.1109/ACCESS.2019.2906693.
6. Chen, J. et al. (2005) 'Wearable sensors for reliable fall detection', Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings. IEEE, 7 VOLS, pp. 3551–3554. doi: 10.1109/iembs.2005.1617246.
7. Cho, H. and Yoon, S. M. (2019) 'Applying singular value decomposition on accelerometer data for 1D convolutional neural network based fall detection', Electronics Letters, 55(6), pp. 320–322. doi: 10.1049/el.2018.6117.
8. Daga, B. S., Ghatol, A. A. and Thakare, V. M. (2018) 'Silhouette based human fall detection using multimodal classifiers for content based video retrieval systems', 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2017, 2018-Janua, pp. 1409–1416. doi: 10.1109/ICICICT1.2017.8342776.
9. Delgado-Escaño, R. et al. (2020) 'A cross-dataset deep learning-based classifier for people fall detection and identification', Computer Methods and Programs in Biomedicine, 184. doi: 10.1016/j.cmpb.2019.105265.
10. Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient kNN classification algorithm for big data. Neurocomputing, 195, 143-148.
11. Gupta, R. et al. (2020) 'Compressive sensing based privacy for fall detection', pp. 1–10. Available at: http://arxiv.org/abs/2001.03463.
12. Han, J. et al. (2013) 'Enhanced computer vision with Microsoft Kinect sensor: A review', IEEE Transactions on Cybernetics. IEEE, 43(5), pp. 1318–1334. doi: 10.1109/TCYB.2013.2265378.
13. Han, Q. et al. (2020) 'A Two-Stream Approach to Fall Detection With MobileVGG', IEEE Access, 8, pp. 17556–17566. doi: 10.1109/access.2019.2962778.
14. Haq, A. U., Li, J. P., Memon, M. H., Nazir, S., & Sun, R. (2018). A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. Mobile Information Systems, 2018.
15. 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.
16. Jung, H. et al. (2020) 'Enhanced algorithm for the detection of preimpact fall for wearable airbags', Sensors (Switzerland), 20(5).

17. Kang, J. M., Yoo, T. and Kim, H. C. (2006) 'A wrist-worn integrated health monitoring instrument with a tele-reporting device for telemedicine and telecare', IEEE Transactions on Instrumentation and Measurement, 55(5), pp. 1655–1661. doi: 10.1109/TIM.2006.881035.

18. Kerdjidj, O. et al. (2020) 'Fall detection and human activity classification using wearable sensors and compressed sensing', Journal of Ambient Intelligence and Humanized Computing. Springer Berlin Heidelberg, 11(1), pp. 349–361. doi: 10.1007/s12652-019-01214-4.

19. Kocaoğlu, S. (2020). Investigating the Impact of Activity Class Number in Fall Detection Systems, Bilecik Şeyh Edebali University Journal of Science, 7(2), 886-895, DOI: 10.35193/bseufbd.714198.

20. Kocaoğlu, S., & Akdoğan, E. (2019). Design and development of an intelligent biomechatronic tumor prosthesis. Biocybernetics and Biomedical Engineering, 39(2), 561-570.

21. Krupitzer, C., Sztyler, T., Edinger, J., Breitbach, M., Stuckenschmidt, H., & Becker, C. (2018, March). Hips do lie! A position-aware mobile fall detection system. In 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom) (pp. 1-10). IEEE.

22. Li, Q. et al. (2009) 'Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information', Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009. IEEE, pp. 138–143. doi: 10.1109/BSN.2009.46.

23. Liu, K. C., Hsieh, C. Y., Hsu, S. J. P., & Chan, C. T. (2018). Impact of sampling rate on wearable-based fall detection systems based on machine learning models. IEEE Sensors Journal, 18(23), 9882-9890.

24. Martinez-Villasenor, L., Ponce, H. and Perez-Daniel, K. (2019) 'Deep learning for multimodal fall detection', Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 2019-Octob, pp. 3422–3429. doi: 10.1109/SMC.2019.8914429.

25. Mastorakis, G. and Makris, D. (2014) 'Fall detection system using Kinect's infrared sensor', Journal of Real-Time Image Processing, 9(4), pp. 635–646. doi: 10.1007/s11554-012-0246-9.

26. Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

27. Nait-Charif, H. and McKenna, S. J. (2004) 'Activity summarisation and fall detection in a supportive home environment', Proceedings - International Conference on Pattern Recognition. IEEE, 4, pp. 323–326. doi: 10.1109/ICPR.2004.1333768.

28. Nath, A., & Subbiah, K. (2018). The role of pertinently diversified and balanced training as well as testing data sets in achieving the true performance of classifiers in predicting the antifreeze proteins. Neurocomputing, 272, 294-305.

29. Nho, Y.-H., Lim, J. G. and Kwon, D.-S. (2020) 'Cluster-Analysis-based User-Adaptive Fall Detection using Fusion of Heart Rate Sensor and Accelerometer in a Wearable Device', IEEE Access, 4, pp. 1–1. doi: 10.1109/access.2020.2969453.

30. Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

31. Nyan, M. N. et al. (2006) 'Distinguishing fall activities from normal activities by angular rate characteristics and high-speed camera characterization', Medical Engineering and Physics, 28(8), pp. 842–849. doi: 10.1016/j.medengphy.2005.11.008.

32. Oshiro, T. M., Perez, P. S., & Baranauskas, J. A. (2012). How many trees in a random forest?. In International workshop on machine learning and data mining in pattern recognition (pp. 154-168). Springer, Berlin, Heidelberg.

33. Psathas, A. P., Papaleonidas, A., Papathanassiou, G., Valkaniotis, S., & Iliadis, L. (2020). Classification of Coseismic Landslides Using Fuzzy and Machine Learning Techniques. In International Conference on Engineering Applications of Neural Networks (pp. 15-31). Springer, Cham.

34. Putra, I. P. E. S. et al. (2018) 'An event-triggered machine learning approach for accelerometer-based fall detection', Sensors (Switzerland), 18(1), pp. 1–18. doi: 10.3390/s18010020.

35. Putra, I. P. E. S. and Vesilo, R. (2018) 'Window-size impact on detection rate of wearable-sensor-based fall detection using supervised machine learning', 2017 IEEE Life Sciences Conference, LSC 2017, 2018-Janua, pp. 21–26. doi: 10.1109/LSC.2017.8268134.

36. Riquelme, F., Espinoza, C., Rodenas, T., Minonzio, J. G., & Taramasco, C. (2019). Ehomeseniors dataset: An infrared thermal sensor dataset for automatic fall detection research. Sensors (Switzerland), 19(20). https://doi.org/10.3390/s19204565

37. Saleh, M. and Le Bouquin Jeannès, R. (2018) 'An efficient machine learning-based fall detection algorithm using local binary features', European Signal Processing Conference. EURASIP, 2018-Septe, pp. 667–671. doi: 10.23919/EUSIPCO.2018.8553340.

38. Saleh, M. and Jeannes, R. L. B. (2019) 'Elderly Fall Detection Using Wearable Sensors: A Low Cost Highly Accurate Algorithm', IEEE Sensors Journal. IEEE, 19(8), pp. 3156–3164. doi: 10.1109/JSEN.2019.2891128.

39. Srinivasan, S. et al. (2007) 'Towards automatic detection of falls using wireless sensors', Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, pp. 1379–1382. doi: 10.1109/IEMBS.2007.4352555.

40. Sucerquia, A., López, J. D. and Vargas-Bonilla, J. F. (2017) 'SisFall: A fall and movement dataset', Sensors (Switzerland), 17(1). doi: 10.3390/s17010198.

41. Sucerquia, A., López, J. D. and Vargas-Bonilla, J. F. (2018) 'Real-life/real-time elderly fall detection with a triaxial accelerometer', Sensors (Switzerland), 18(4), pp. 1–18. doi: 10.3390/s18041101.

42. United Nations (2019) World Population Prospects 2019, Department of Economic and Social Affairs. World Population Prospects 2019. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12283219.

43. WHO (2007) Falls. Available at: https://www.who.int/news-room/fact-sheets/detail/falls (Accessed: 19 February 2020).

44. Wu, G. (2000) 'Distinguishing fall activities from normal activities by velocity characteristics', Journal of Biomechanics, 33(11), pp. 1497–1500. doi: 10.1016/S0021-9290(00)00117-2.

45. Yacchirema, D., de Puga, J. S., Palau, C., & Esteve, M. (2018). Fall detection system for elderly people using IoT and big data. Procedia computer science, 130, 603-610.