International Journal of Engineering

Journal Homepage: www.ije.ir

Fall Detection using Deep Learning Algorithms and Analysis of Wearable Sensor Data by Presenting a New Sampling Method

R. Keramati Hatkeposhti, M. Yadollahzadeh Tabari*, M. GolsorkhtabariAmiri

Department of Computer Engineering, Islamic Azad University, Babol Branch, Babol, Iran

PAPER INFO

Paper history:
Received 01 May 2022
Received in revised form 01 June 2022
Accepted 14 June 2022

Keywords:
Deep Learning
Fall Detection
Health Monitoring
Sisfall
Wearable Sensors

ABSTRACT

Fall is one of the most critical health challenges in the community, which can cause severe injuries and even death. The primary purpose of this study is to develop a deep neural network using wearable sensor data to detect falls. Most datasets in this field suffer from the problem of data imbalance so that the instances belonging to the Fall classes are significantly less than the data of the normal class. This study offers a dynamic sampling technique for increasing the balance rate between the samples belonging to fall and normal classes to improve the accuracy of the learning algorithms. The Sisfall dataset was used in which human activity is divided into three categories: normal activity (BKG), moments before the fall (Alert), and role on the ground (Fall). Three deep learning models, CNN, LSTM, and a hybrid model called Conv-LSTM, were implemented on this dataset, and their performance was evaluated. Accordingly, the Conv-LSTM hybrid model presents 96.23%, 98.59%, and 99.38% in the Sensitivity parameter for the BKG, Alert, and Fall classes, respectively. For the accuracy parameter, we have managed to reach 97.12%. In addition, by using noise smoothing and removal techniques, we can hit a 97.83% accuracy rate. The results indicate the proposed model's superiority compared to other similar studies.

***doi**: 10.5829/ije.2022.35.10a.13*

1. INTRODUCTION

Increasing life expectancy and social change have increased the population of the elderly living alone in their homes. Falling is a significant danger to the lives of these people. According to reliable sources, a sudden fall is the leading cause of fatal injuries and the most common cause of hospitalization. After road traffic injuries, falling is the second leading cause of death due to unintentional injuries[1]. On the other hand, the medical expenses for the fall accident are also increasing [1].

In addition to physical injury, falls can cause much psychological damage, especially to the elderly. They can cause other side effects such as decreased physical activity, fear of falling, depression, anxiety, loneliness, and loss of confidence in independent living [2].

Of course, the elderly are not the only group affected by the fall; anyone with any disability or hospitalized patients will frequently experience the fall. The incidence of falls is higher in people suffering from chronic diseases such as Parkinson's, osteoarthritis, and osteoporosis [3].

Response and relief time are critical in preventing the most severe possible consequences of fall-related side effects and injuries. In the meantime, fall detection systems (FDS) can provide rapid services and assistance to the individual to reduce the consequences of the fall and ensure the well-being of the elderly at home and better patient management. Despite fall detection systems, a person can be saved by timely warning or calling for help.

FDSs can be classified into two general groups. The first group is systems based on sensors located around the user and are mainly based on machine vision (Context-Aware System). Using these tools may have high accuracy in fall detection, but there are also

*Corresponding Author Institutional Email: *m_tabari@baboliau.ac.ir* (M. Yadollahzadeh Tabari)

[1] http://www.who.int/mediacentre/factsheets/fs344/en/

Please cite this article as: R. Keramati Hatkeposhti, M. Yadollahzadeh Tabari, M. GolsorkhtabariAmiri, Fall Detection using Deep Learning Algorithms and Analysis of Wearable Sensor Data by Presenting a New Sampling Method, ***International Journal of Engineering, Transactions A: Basics***, Vol. 35 No. 10, (2022), 1941-1958

challenges. These systems are effective for use in a fixed area. In this particular range, changes in factors such as lighting, furniture arrangement, or the presence of unexpected elements and various noises may have a negative impact on the performance of these systems. It also violates people's privacy [4].

The existence of these challenges, on the one hand, and the advancement of new technologies and wearable sensors, such as accelerometers, gyroscopes, and magnetometers, on the other, led to falling detection systems moving toward the second group, namely wearable and inertial sensors (Inertial Measurement Units: IMUs). By analyzing the data of these sensors, they can detect the fall in real-time. Wearable fall detection systems are portable and do not have privacy issues. They are very accurate and can identify people's activities and send warning messages to third parties for help before or after the fall to reduce the negative consequences of the fall [5, 6].

A typical sensor in wearable devices is the three-axis accelerometer, which is widely used due to its low cost, small size, and installation in all smartphones.

Developers of fall detection systems are currently facing many challenges. One of the most important is the lack of access to real data. Obtaining actual data from people's behavior in everyday life is not an easy task. Falling is a sudden and unpredictable behavior. Therefore, we are required to use datasets with simulated movements in laboratories that, due to the high risk, most public data available to the target population of the elderly are not tested. This reduces the accuracy of the results of these systems in real life [7].

Another challenge is the unbalanced data used in deep learning algorithms. In this case, the number of samples belonging to one class is very different from the number of samples belonging to another class. In such cases, the system is usually biased for the majority class, and the probability of incorrectly detecting the fall increases.

Accordingly, this study aims to design an effective algorithm based on deep learning using wearable sensor data for fall detection. With relatively simple architecture, while increasing the balance rate between the data, can detect the fall directly on the sensor current with high accuracy.

The article is categorized as follows:

After the introduction in the first section, which discusses related issues and challenges, the second part refers to the classification of fall detection systems and a review of studies conducted in this field. The third section describes the conditions for selecting a database. Then in the fourth section, the proposed method of how to sample the data with the proposed models is mentioned. In the fifth section, the experiments and evaluations resulting from the implementation of the models are shown and compared with other studies. In the sixth section, a general discussion and conclusion are made.

2. RELATED WORKS

The review of past studies can be divided into two parts: datasets and algorithms used. The trend of using sensors used in FDSs before and after 2014 has been reported in literature [8]. The report shows that the number of FDSs using cameras has greatly decreased since 2014, and the use of accelerometers is expanding.

Chen et al. [9] demonstrated that accelerometer data are mainly crucial for fall detection, and about 85% of the selected features are related to accelerometers, about 10% are related to gyroscopes, and the rest are related to other sensors.

Özdemir [10] has conducted a study on the effect of sensor location on fall detection accuracy. They performed experiments with six different traditional machine learning algorithms. They showed that the sensors, which are located close to the center of gravity of the human body (like the chest and waist), are the most effective place.

The results of studies showed that when a system combines several types of sensors, its performance is significantly improved because each of the sources and sensors can independently provide excellent and sufficient information about different aspects of human activities and balance characteristics of individuals [11].

Since 2014, the trend of using algorithms has changed from threshold algorithms to machine learning algorithms and, in these recent years, has shifted to deep learning algorithms [12, 13].

In 2017, Aziz et al. [14] made a detailed comparison of ten fall detection algorithms that use accelerometer-based sensor data. Five of them used threshold-based methods, and the remaining methods were based on machine learning algorithms. The final comparison showed that machine learning-based fall detection methods perform better than threshold-based algorithms [14].

Ozdemir and Barshan [15] used 2520 experiments to create a large dataset of 14 volunteers and performed a set of standard movements that included 20 fall behaviors and 16 Activities of Daily Living (ADL). Using multilayer perceptron (MLP) for binary classification between normal behavior and fall, their fall detection system provided 95% accuracy [15].

Of course, traditional machine learning algorithms need to extract appropriate features from the data. Feature extraction must be done before any learning and must be manually determined that their effectiveness depends on the researcher's knowledge and genius. This can make feature extraction and selection very complex and significantly affect the efficiency and performance of the machine learning model [16].

But deep learning methods can automatically perform the representations needed to detect and classify raw data without human intervention and the need for specialized knowledge and automatically select the appropriate features [17].

Along with the rapid advancement of deep learning, data enhancement, and the promotion of computing hardware, deep learning models to identify human movements and activities, including fall behavior, have grown significantly [18].

Chen et al. [19] designed a Convolutional Neural Network (CNN) model consisting of three layers of Convolutional and three layers of dense and used it to detect falls. They used a dataset consisting of 31,688 samples and eight types of activities for analysis. This study compared Support Vector Machine (SVM) and Deep Belief Network (DBN) methods with the CNN method, which CNN model provided the best accuracy with 93.8%.

Tao and Yun [20] proposed the Long Short Term Memory (LSTM) model and Skeleton Data, recorded by Kinect, to predict the fall. This model reports a value of 91.7% for the sensitivity parameter and 75% for the specificity parameter, which indicates that this model can detect most pre-impact falls but has a high false alarm rate [30].

Torti et al. [21] used the sliding windowing technique and the Sisfall dataset to detect a fall, classifying the fall process into three stages: "non-fall, Alert, and fall." They reported a high sensitivity value for the fall class (98.73%). But they obtained lower accuracy regarding non-fall and Alert (88.39% and 91.08%). Only the LSTM model has been used in this study, and no comparison has been made with other deep learning structures [21].

Musci et al. [22] proposed a Recurrent Neural Network (RNN) model for fall detection using accelerometer data. The core of their neural network architecture is a fully connected layer that processes raw data, followed by two LSTM layers. They trained and tested their model with the Sisfall dataset. Their model achieved 97.16% accuracy in fall behavior and 94.14% in ADL behavior.

3. DATASET SELECTION

Choosing the suitable dataset for training and validating models has always been an important and influential issue. Due to the emergence and development of intelligent sensors, it was decided to use the data of laboratory wearable hybrid sensors for this study and finally selected the Sisfall dataset for this study.

In 2017, Sucerquia et al. [23], instead of using smartphones, introduced a handheld device to perform various experiments to identify normal and falling behaviors; thus providing the Sisfall dataset. The device consisted of two 3D accelerometers and a gyroscope, and sampled sensor data were at a frequency of 200 Hz. As a result, it also takes advantage of hybrid sensors, which can effectively deliver results.

The dataset was produced with the help of 38 volunteers, including 19 men and 19 women in the age range of 19 to 75 years, divided into two groups the elderly and adults. The elderly group consisted of 15 participants (eight males and seven females between 60 and 75 years old). The adult group had 23 participants (11 males and 12 females between 19 and 30 years old) who recorded more than 4,500 experiments. The data included 19 normal behaviors and 15 types of falls. One of the advantages of this data is the simultaneous use of young and older adults to test and prepare data, which is close to reality. Table 1 presents the types of fall behaviors defined in the Sisfall dataset and their descriptions.

In 2018, Musci and colleagues [22] did more labeling on the Sisfall dataset, which includes the following three classes:

BKG (Back-ground)**:** The class is the default, and the person behaves normally and has control over their situation.

Alert: The interval is the time when a person loses his balance and goes from normal to falling.

Fall: Specify the position in which the person is completely lying on the ground.

Unlike other datasets, Sisfall has a third class called "Alert, " a normal behavior close to falling. This class, which has also been considered in this study, can be used to modify the assessments of fall detection systems and identify fall behavior before impact.

Figure 1 shows an example of a data sequence with temporary annotations from Sisfall. This figure shows the status of the first three-axis accelerometer signals in the three regions. In the first few seconds, the person behaves normally and is moving, labeled BKG (blue color range). The initial imbalance is marked as Alert in yellow and the Fall range in red. The person then gets up and continues to move. These intervals are marked with the Alert and BKG labels, respectively.

Figure 2a shows an example signal of a fall behavior where a red circle indicates the moment of fall. Figure 2b also shows the falling behavior discretely in three classes, BKG, Alert, and Fall.

4. THE PROPOSED METHOD

4. 1. Structure of the Models Used

In this study, a deep learning approach was used in which two models of CNN, LSTM, and a combined model of the two called Conv-LSTM, were examined.

Numerous experiments were performed to determine the best structure for each model. In these experiments, the performance of the models was evaluated with

TABLE 1. Fall behaviors defined in the Sisfall database

Code	Activity	Trials	Duration
F01	Fall forward while walking caused by a slip	5	15s
F02	Fall backward while walking caused by a slip	5	15s
F03	Lateral fall while walking caused by a slip	5	15s
F04	Fall forward while walking caused by a trip	5	15s
F05	Fall forward while jogging caused by a trip	5	15s
F06	Vertical fall while walking caused by fainting	5	15s
F07	Fall while walking, with use of hands in a table to dampen fall, caused by fainting	5	15s
F08	Fall forward when trying to get up	5	15s
F09	Lateral fall when trying to get up	5	15s
F10	Fall forward when trying to sit down	5	15s
F11	Fall backward when trying to sit down	5	15s
F12	Lateral fall when trying to sit down	5	15s
F13	Fall forward while sitting, caused by fainting or falling asleep	5	15s
F14	Fall backward while sitting, caused by fainting or falling asleep	5	15s
F15	Lateral fall while sitting, caused by fainting or falling asleep	5	15s

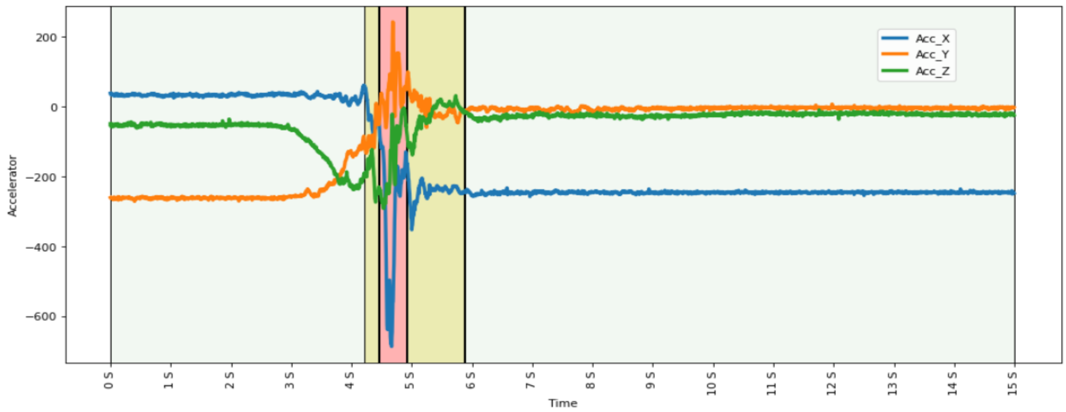
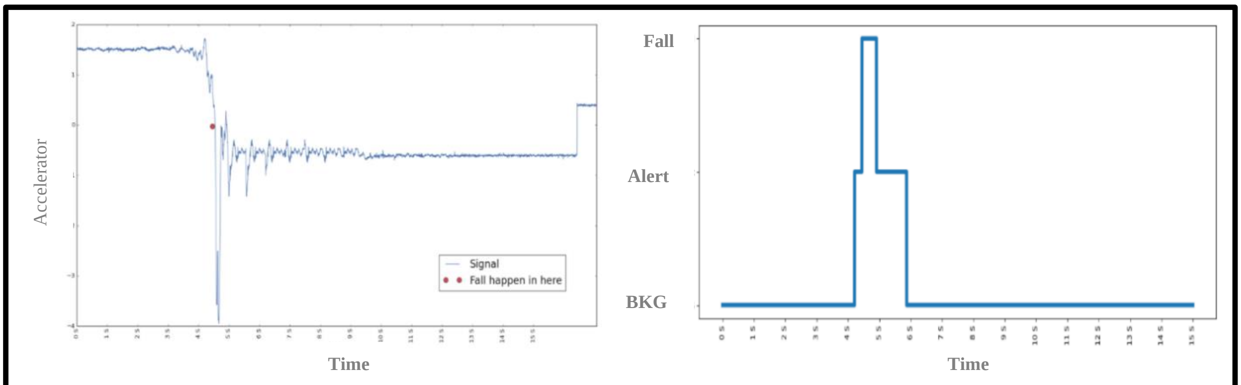


Figure 1. Sample labeling SA01/F10_R01 showing the ranges of the three classes BKG, Alert, and Fall in blue, yellow, and red, respectively



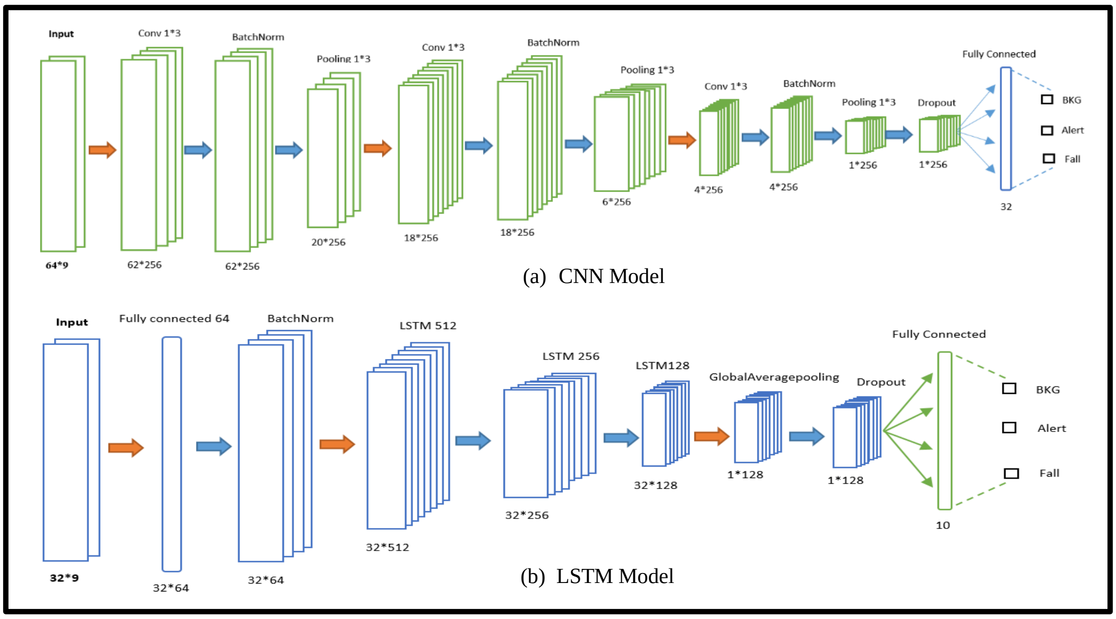
(a) Sample motion signal display SA01/F10_R01

(b) Display the falling status

Figure 2. An example of a fall status signal

different numbers and sizes of layers such as Convolutional, LSTM, Dropout, and Fully Connected. Finally, the best structure was obtained for each model shown in Figures 3 and 4.

Models with different parameters were trained, and the stage (Epoch) with the lowest "Loss Validation" error was considered the best mode. Based on the nature of the data and the experience of working with deep networks, various hyperparameters were adjusted and examined in the model construction and how to train them. These parameters include the number of layers, the size of each layer filter, the number and size of the dropout layer, the number and size of the Fully Connected Layer, the Activation Function, the Optimizer Function, the Kernel Size, and the Learning Rate, BatchSize value, Decay parameters, and Epoch number.



(a) CNN Model

(b) LSTM Model

Figure 3. The structure of CNN and LSTM networks with the layers used

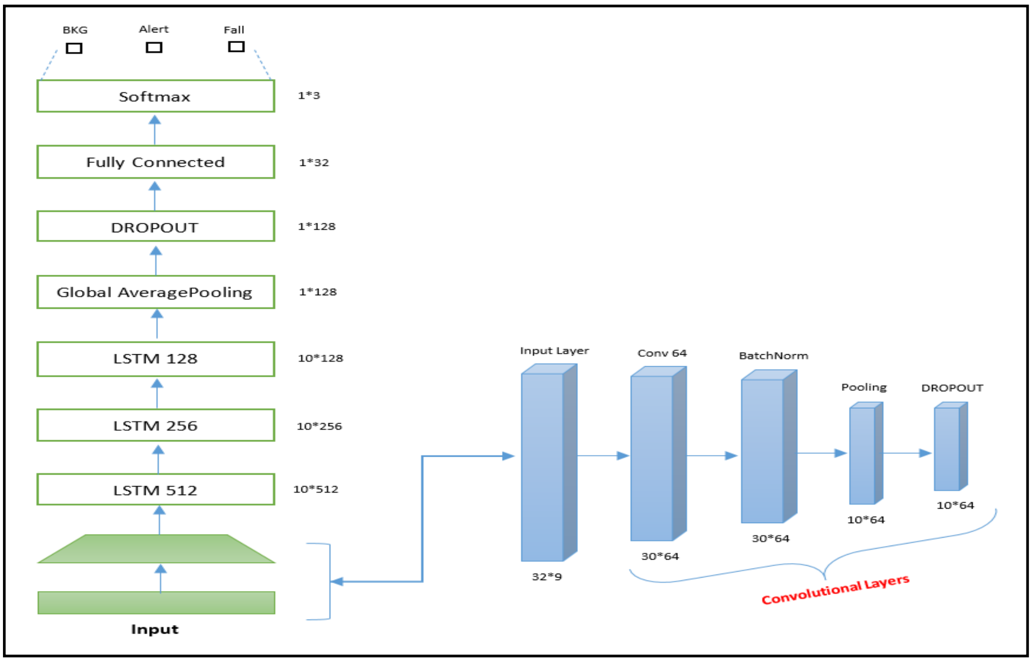


Figure 4. Conv-LSTM hybrid network structure with the number and status of its conv and LSTM layers

As shown in Figure 3, the best CNN model is when three Convolutional layers with sizes 256 are used. The LSTM Network also provided the best results when three layers of LSTM were used, with sizes 512, 256, and 128, respectively.

In the hybrid model, an attempt has been made to design a combination of the best modes of the CNN and LSTM models to achieve better results and incorporate the advantages of using CNN and LSTM. Using the CNN algorithm, appropriate features are extracted automatically, but it ignores long-term time relationships in the time series that are important for identifying behaviors and movements. LSTM, on the other hand, uses memory cells to learn long-term dependencies on time series data and dependencies between extracted properties. Of course, due to its complex structure, it has a long and very time-consuming execution time.

Therefore, in the Conv-LSTM hybrid model, CNN layers first extract the raw data properties and send them to the LSTM layers to identify temporal relationships. Compared to the single LSTM method, this technique saves a lot of time and execution calculations.

As shown in Figure 4, a combination grid of a one-dimensional convolutional layer of size 64 starts, which performs the input processing. After identifying essential and practical features, the three layers of LSTM with sizes 512, 256, and 128 are sent.

In addition, the Batchnormalize layer is used to help normalize data and adjust input data after the convolutional layer. Due to the network structure, the adopted solution also includes two Dropout layers with a rate of 50%. Of course, in the meantime, the MaxPooling technique has also been used to adjust and reduce the data and select the best option for the next round. The activation function used in layers is also a modified Rectified Linear Unit (ReLU). However, functions such as Scaled exponential linear unit (SeLU) can also be used, and repeated experiments have shown that the two functions have almost the Similar performance.

4. 2. Preparation and Preprocessing of Data

- **Standardization**

All machine learning architectures have one thing in common, and that is the issue of normalization and standardization of data input to the network. Using standardization, data is scaled, and data distribution is normalized. Standardizing is to obtain values with a mean of zero and a standard variance or deviation of one. If the mean of the original data is equal to μ and their standard deviation is also σ, the value of Z is obtained based on Equation (1), where xi is a data point in between (x1, x2,…, xn)[1].

$$Z = \frac{X_i - \mu}{\sigma} \tag{1}$$

and based on this conversion, we will have:

$$\bar{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma} \right) = \frac{\bar{x} - \mu}{\sigma} = 0 \tag{2}$$

On the other hand, we have to calculate the variance:

$$\sigma_Z^2 = \frac{1}{n} \sum_{i=1}^{n} (Z_i - \bar{Z})^2 = \frac{1}{n} \sum_{i=1}^{n} Z_i^2 = \sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma} \right)^2 = \frac{1}{\sigma^2} \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2 = \frac{\sigma^2}{\sigma^2} = 1 \tag{3}$$

Therefore, in standardization, the conversion $\sim (\mu . \sigma^2) \rightarrow \sim (0 . 1)$ takes place. Standardization is essential and effective when dealing with many features and data at different scales. Standardization causes functions to be in saturated areas later, and as a result, the training process encounters problems such as Vanishing Gradient later.

- **Smoothing Data**

In addition to standardization, one of the essential steps in signal processing is removing noise and unwanted factors that filters can do. Filters are divided into analog and digital. Digital filters have advantages over analog filters that make them more widely used. These include high reliability, high performance, and no need for unique settings. Digital filters are divided into two general categories Infinite Impulse Response (IIR) and Finite Impulse Response (FIR)[2].

At a glance, if the Y(z) function represents a digital filter,

$$Y(z) = X(z)\, H(z) = \frac{b(1)+b(2)z^{-1}+ \dots\dots\dots +b(n+1)z^{-n}}{a(1)+a(2)z^{-1} + \dots\dots\dots +a(m+1)z^{-m}} \tag{4}$$

Then:
1) If n = 0, we will have an IIR filter.
2) If m = 0, we will have an FIR filter.

FIR filters have an utterly linear phase, which is very effective in image and audio processing applications, while IIR filters are nonlinear. FIR filters are non-reversible, but IIR filters have a feedback path. One of the advantages of IIR filters over FIR filters is that analog filters can be converted to digital IIR filters.

According to the nature of the data, the IIR 1st order low-pass filter was used to remove noise. This filter was chosen because of its simplicity, and at the same time, it has provided similar results to other filters. The relationship of this filter is shown in Equation (5) [9].

$$S^*[n] = \alpha\, S[n-1] + (1 - \alpha)S[n]. where\ 0 \le \alpha \le 1 \tag{5}$$

---

[1] https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832

[2] https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832

However, after the α coefficient is infiltrated, it can be converted to the relation $S^*[n] = S[n] + \alpha\,(S[n-1] - S[n])$. In this case, $S^*[n]$ is the current filtered signal, $S\,[n]$ is the current instantaneous signal, and $S\,[n-1]$ is the previous order signal.

Here α is an adaptive coefficient in the range [0, 1] that can have the following states:

1) If $\alpha \neq 0$, then it will be an IIR filter.

2) If $\alpha = 0$, then the output is exactly equal to the input; in fact, no filter is applied.

3) If $\alpha \rightarrow 1$, the output moves towards a constant value and eventually becomes linear.

If the value of the α coefficient is close to 1, many of the signal behaviors may be lost, and the classification task may be difficult. "Alert" labels may even be identified as "BKG". If α is considered too small, no noise may be detected or eliminated. Given the conditions of the data and repeated experiments, the best option for the α coefficient value (0.9) was considered. Figure 5 shows the results of applying the filter to the three-axis signals of the first accelerometer.

Signal magnitude area (SMA), which describes changes in human activity, has been used to determine falls. This parameter is represented by Equation 6, where x_t, y_t and z_t represent the accelerometer sensor readings on the x, y, and z axes, respectively.

$$SMA = \sqrt{X_t^2 + Y_t^2 + Z_t^2} \tag{6}$$

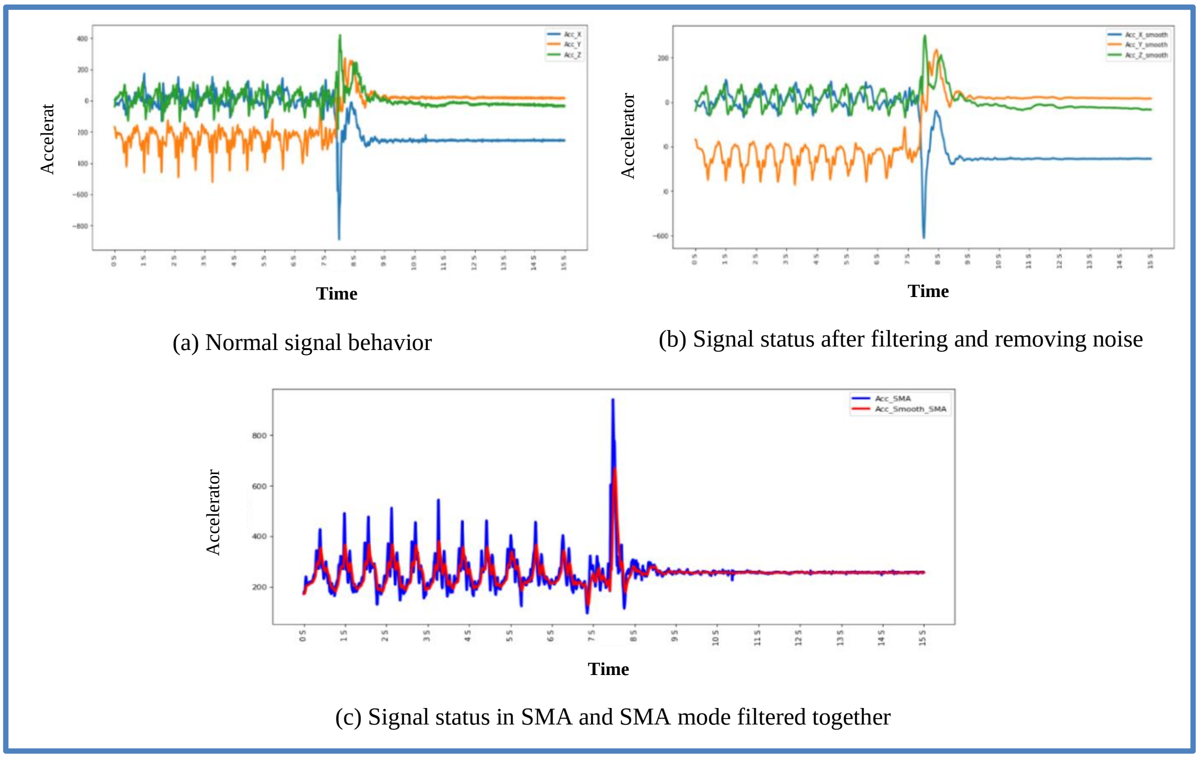
For a better comparison, the SMA mode of the signals is shown in Figure 5. Figure 5(a) shows the normal behavior of the signal, which was transformed into Figure 5(b) by applying a filter and removing noise. Figure 5(c) also indicates the status of the signals simultaneously in SMA and SMA filtered mode, and the effectiveness of filtering in this mode is quite visible.

In this study, a standardization method was used for data preprocessing, and IIR filtering was used to remove noise. The models were trained once without noise cancellation and again with noise cancellation, and the results were evaluated.

4. 1. Dynamic Sampling of Data with the Approach of Increasing the Alignment Rate

Using a proper data sampling method can ensure the success of learning algorithms in generalizing the training to the experimental stage. In solving the issue of imbalanced data, a new technique with a data amplification approach for sampling and increasing the number of minority class samples is proposed and has been operational.

Ordinally, for data sampling, the input data sequence must be adjusted for each training, validation, and testing set with fixed-size windows (w). Of course, the window size of w is a hyperparameter, the precise adjustment of which is significant for the effectiveness of the network architecture. On the other hand, when calculating data flow, the data should be placed at the edge of a sliding window. Accordingly, determining the degree of overlap and the choice of movement steps (Stride) is also very important and influential.



(a) Normal signal behavior

(b) Signal status after filtering and removing noise

(c) Signal status in SMA and SMA mode filtered together

Figure 5. Display the three-axis signals of the first accelerometer corresponding to the sample SA02/F03_R01 in three positions

To move the slider in the data, according to Equation (7), because it always starts with the BKG label, we first use a 50% overlap rate, and as soon as we reach the first Fall label, we increase the overlap percentage. For simplicity, we used overlaps 75, 90, and 95. Then, after finishing the fall label and reaching the BKG label again, a 50% overlapping window is used. Figure 6, illustrates this approach, where the compaction and increase of overlap in the Fall range are evident.

$$Stride = \begin{cases} 50\% & lable = BKG \\ 75\%\ or\ 90\%\ or\ 95\% & lable \neq BKG \end{cases} \tag{7}$$

As shown in Figure 7, based on the average_sensititvity parameter, the best accuracy in 64-width windowing was on CNN. But in LSTM and Conv-LSTM networks, the best result is obtained in the w=32 with 95% overlap for the Alert and Fall sections.

The best values for the window width and overlap parameters were determined by examining the sensitivity criterion, which is shown in Tables 2, 3, and 4. As it turns out, as the overlap increases, the results improve, and the classification error decreases. The best result is the Conv-LSTM hybrid model, which uses a window width of w=32 and 95% overlap in the Fall and Alert classes.

Figure 8 shows the sampling status with windowing w = 32 and dynamic overlaps, and it is clear that at 95% overlap status, a better balance is established between classes. The initial dataset was unbalanced; with a normal overlap of 50%, the equilibrium rate (ratio of the number of falling samples to the number of normal behavior samples) was less than 1%, with a 95% overlap in the Fall and Alert sections, this value reached 13.92%, which almost increased 14 times. This data increase in the minority class can be very effective in system performance and accuracy improvement.

So with the same technique, without much increase for the BKG class, a lot of new data from the Alert and Fall class was added to the collection, which significantly improved the imbalance problem.

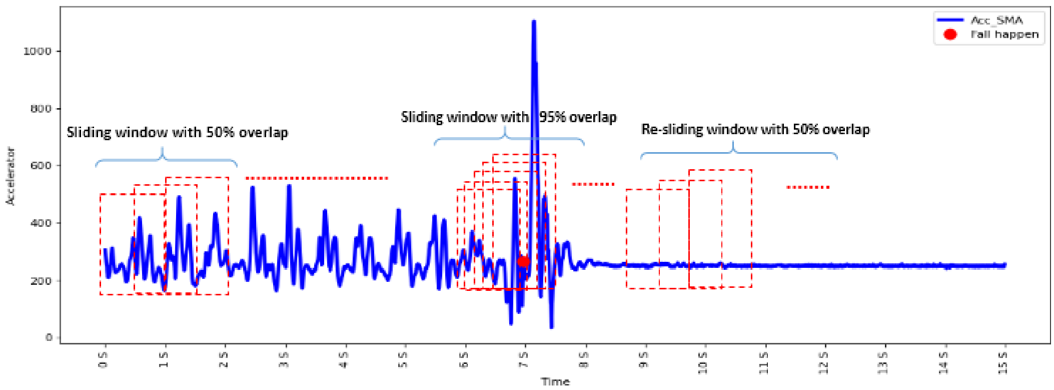


Figure 6. Sampling process and overlap in different classes related to sample SA01/F10_ R01

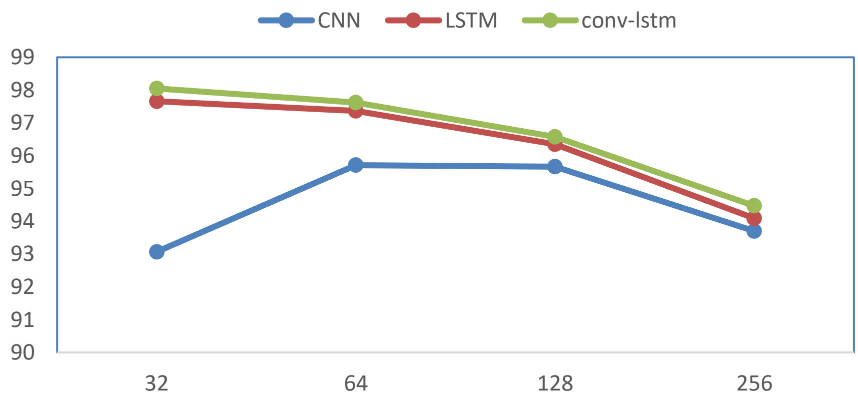


Figure 7. Status diagram of the Average_sensititvity average parameter in three models with different windowing

TABLE 2. Status of window values and overlap in sampling in the CNN model

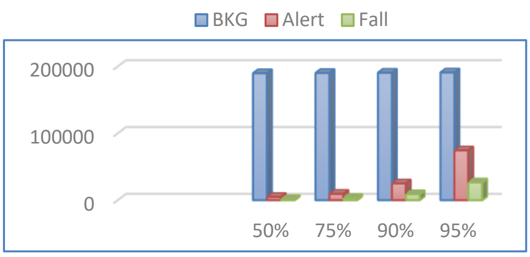
ROW	Window size (W)	degree of overlap (S%)	Accuracy	CNN: Sensitivity (%)			
				BKG	Alert	Fall	Average_sensititvity
1	32	50	87.33	87.69	73.89	86.03	82.54
2	32	75	89.67	90.31	77.24	88.64	85.40
3	32	90	91.08	92.01	83.69	92.03	89.24
4	32	95	92.57	93.33	89.16	96.71	93.07
5	64	50	90.48	90.83	77.36	87.33	85.17
6	64	75	90.45	90.89	81.26	91.03	87.73
7	64	90	93.54	93.71	92.41	93.07	93.06
8*	**64**	**95**	**94.98**	**94.68**	**95.53**	**96.92**	**95.71**
9	128	50	92.44	92.81	77.15	93.22	87.73
10	128	75	92.61	92.96	79.79	91.78	88.18
1	128	90	92.89	93.26	89.43	94.45	92.38
12	128	95	95.39	95.26	95.76	95.97	95.67
13	256	50	92.11	92.24	64.73	95.35	84.11
14	256	75	92.07	92.49	81.96	86.86	87.11
15	256	90	92.76	93.10	89.11	91.86	91.36
16	256	95	93.93	94.06	93.26	93.78	93.70

TABLE 3. Status of windowing values and overlap in sampling in LSTM model

ROW	Window size (W)	degree of overlap (S%)	Accuracy	LSTM: Sensitivity (%)			
				BKG	Alert	Fall	Average_sensititvity
1	32	50	88.90	89.28	73.87	89.48	84.21
2	32	75	90.90	91.47	79.62	90.95	87.35
3	32	90	95.02	95.14	93.58	96.27	94.99
4*	**32**	**95**	**96.61**	**95.85**	**98.08**	**99.06**	**97.66**
5	64	50	91.46	91.82	78.29	88.81	86.31
6	64	75	93.27	93.85	81.36	93.31	89.51
7	64	90	95.84	96.61	93.75	96.49	95.62
8	64	95	97.07	97.01	96.95	98.14	97.37
9	128	50	91.88	92.37	75.36	94.72	87.48
10	128	75	93.27	93.91	77.99	94.71	88.87
11	128	90	95.49	95.96	91.40	94.51	93.96
12	128	95	95.98	96.20	95.52	97.34	96.35
13	256	50	92.86	93.31	74.86	84.88	84.35
14	256	75	91.87	92.59	82.65	85.40	86.88
15	256	90	92.99	93.07	89.73	93.90	92.23
16	256	95	94.78	95.00	93.85	93.43	94.09

TABLE 4. Status of windowing values and overlap in sampling in Conv-LSTM hybrid model

ROW	Window size (W)	degree of overlap (S%)	Accuracy (%)	Conv-Lstm: Sensitivity (%)			
				BKG	Alert	Fall	Average_sensititvity
1	32	50	90.15	90.57	73.87	89.48	84.64
2	32	75	91.84	92.38	81.25	90.86	88.16
3	32	90	84.77	94.70	94.65	96.58	95.31
4*	32	95	97.12	96.23	98.59	99.38	98.07
5	64	50	92.65	93.06	77.57	89.50	86.71
6	64	75	93.06	93.56	83.01	92.53	89.70
7	64	90	96.47	96.42	94.08	96.64	95.71
8	64	95	97.41	97.31	97.62	97.94	97.62
9	128	50	93.46	93.95	78.43	93.22	88.53
10	128	75	93.58	94.22	81.64	93.98	89.95
11	128	90	96.06	96.45	93.56	93.62	94.54
12	128	95	96.39	96.33	96.51	96.87	96.57
13	256	50	92.38	92.39	72.98	93.02	86.13
14	256	75	90.93	91.41	82.31	84.67	86.13
15	256	90	93.48	93.60	89.83	92.54	91.99
16	256	95	94.56	94.82	94.80	93.78	94.47



Degree of overlap	Number of BKG	Number of Alert	Number of Fall	Total data	Balance rate (Fall to BKG ratio)
50%	190041	4855	1777	196673	0.94 %
75%	190360	9486	3481	203327	1.83 %
90%	190745	25071	8936	224752	4.68 %
95%	191085	74549	26599	292233	13.92 %

Figure 8. Status of samples number in different classes by dynamic overlap method

5. IMPLEMENTATION AND EVALUATION

5. 1. Implementation In this study, as shown in the flowchart of Figure 9, after selecting the data set, the data were pre-processed, and a new approach was used for data sampling. Due to the structure of the Sisfall dataset and the use of two accelerometers and a three-axis gyroscope, nine channels can be used to receive data. For implementation, all available 9-channel data have been used.

The dataset was divided into three smaller and independent sets, entitled Train with a ratio of 60%, Validation with a ratio of 20%, and Test with a ratio of 20%. The model is trained with Train data and its learning level experiments with validation data. Next, the model's performance is evaluated with test data that it has not seen before.

5. 2. Evaluation Criteria As shown in Figure 10, the data identification and classification results are classified into the following four groups.

Of course, the main challenge of fall detection systems is to reduce false positive (FP) warnings and also to reduce false negative (FN) warnings [24-27]. There are various criteria for evaluating the performance of machine learning algorithms for classification problems; the following parameters can be mentioned [28-30].

$$Accuracy\ (\%) = 100 * \frac{TP+TN}{TN+FN+TP+FP} \tag{8}$$

$$precision\ (\%) = 100 * \frac{TP}{TP+FP} \tag{9}$$

$$Sensitivity\ (\%) = 100 * \frac{TP}{TP+FN} \tag{10}$$

$$Specificity\ (\%) = 100 * \frac{TN}{TN+Fp} \tag{11}$$

$$F-Score\ (\%) = \frac{2*Recal*precision}{Recal+precision} \tag{12}$$

The sensitivity criterion describes the ability to detect a fall, and the specificity criterion describes the FDS's ability to prevent false alarms. The goal is for the model to accurately detect a fall, which implies a fall detection model with a high sensitivity value. We also do not want to have too many false warnings. Therefore, the main criteria for evaluating the models in this study are the sensitivity and specificity parameters, and we will look at other parameters as well.

5. 2. Experiments and Evaluation Results

Figure 11 shows the general laboratory process of this study. All training and testing steps are performed on a system equipped with an Intel Core i7 processor and NVIDIA GeForce 930MX graphics, and 8 GB of main memory with the Windows 10 operating system.

We performed three general experiments to evaluate the proposed models and approach. In the first experiment, the method which was presented in by Musci et al. [22] in 2018, every window that contains at least 10% of the Fall class is labeled Fall. Every non-Fall window in which most samples are in the Alert class is labeled Alerts, and the remaining windows are labeled in the BKG class. Of course, we considered the overlap rate constant and the value of 50%.

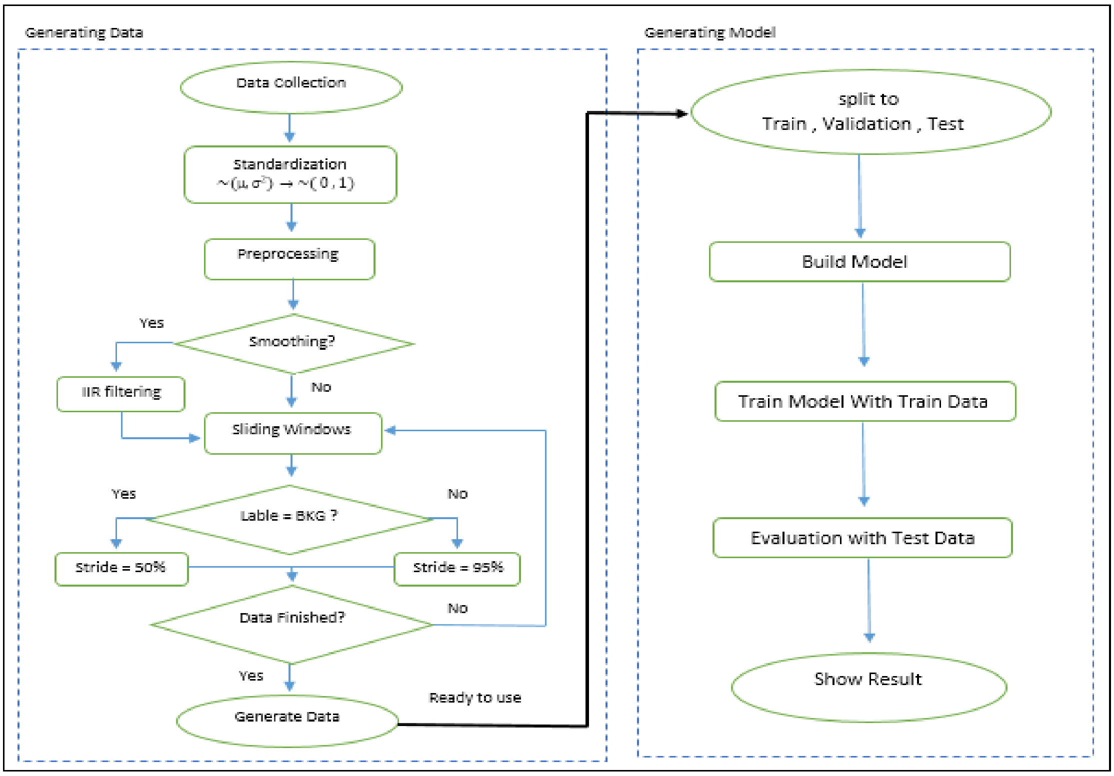


Figure 9. Flowchart of the fall detection process

		Predicted Class: Positive	Predicted Class: Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 10. Classification of real and predicted classes into four groups

In the second experiment, we considered the overlap rate during sampling constant, and 50% and the majority vote approach were used for labeling.

In the third experiment, which is the proposed method of this study, the dynamic overlap was used. We first used a 50% overlap for sampling and increased the overlap to 95% as soon as we reached the first Alert and then the first drop. This process continued until we reached the first normal state (BKG) again, after which we sampled the path again with 50% overlap.

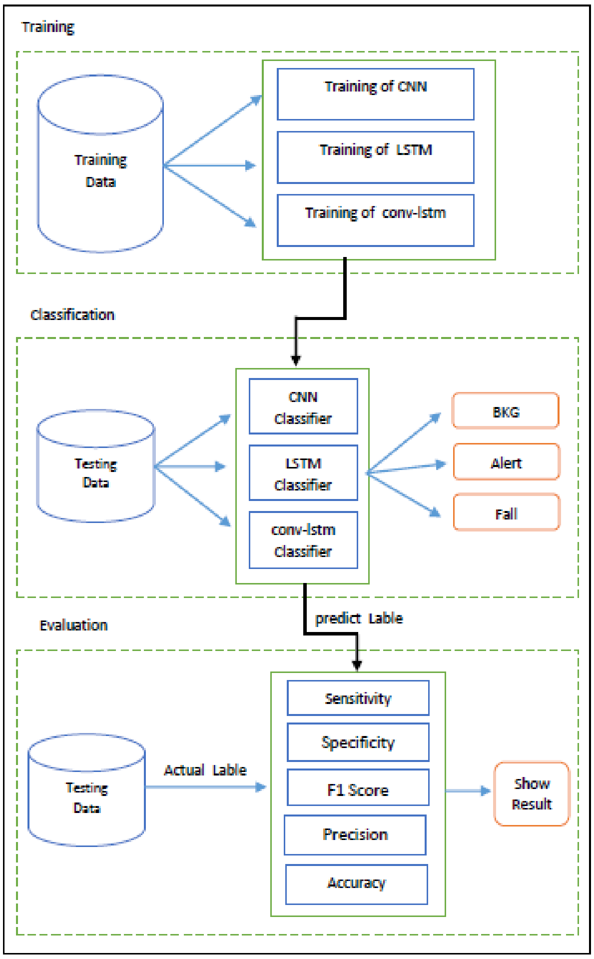


Figure 11. General laboratory process

Tables 5, 6, and 7 summarized the results of these experiments separately for the three proposed algorithms based on sensitivity and specitifity criteria.

The experimental results were also compared with the data reported by Yu et al. [31] studied in 2020 and in 2019 by Torti et al. [21].

Yu et al. [31] studied two models of CNN, LSTM, and a combined model of these two called Conv-LSTM were used to detect the fall, and their performance was evaluated on the Sisfall dataset. Torti et al. [21] also used the sliding windowing technique and the Sisfall dataset to detect falls. They classified the fall process into three stages: "no fall, Alert and fall" and used the LSTM model to classify the three classes.

And as it turns out, the proposed method with 32-width windowing and 95% overlap, compared to other methods, has shown the best accuracy in most classes.

As shown in Table 5, the value of the Accuracy parameter of the proposed method in the CNN model is 94.98%. More importantly, in this model, the value of the Sensitivity parameter in the proposed method in the BKG, Alert, and Fall classes is 94.68%, 95.53 %, and 96.92%, respectively, the best result in all classes compared to other methods.

Of course, only the sensitivity value of the fall class in the torti method is higher than the proposed method. But in this research, only the LSTM model is used, and in the other two classes, the results are weaker than the proposed method.

Also, the value of the specificity parameter in these three classes is 98.93%, 95.13%, and 99.15%, respectively; which is the best result compared to other cases.

According to Table 6, the accuracy parameter of the proposed method in the LSTM model shows a value of 96.61%. The value of the sensitivity parameter in the proposed method for the BKG, alert, and fall classes are 95.85, 98.08, and 99.06, respectively, which is still the best compared to other methods in all classes. At the same time, there is a relative balance between the results of all three classes. Also, the value of the specificity parameter in these three classes is 99.48%, 97.26%, and 99.39%, respectively, which is the best result compared to other cases. On the other hand, a review of the two tables shows that the results of the LSTM model performed much better than the CNN model in all classes.

TABLE 5. Comparison of the performance results of the proposed method with the results of some similar studies in the CNN model

CNN - Best ACC : 94.98%

Evaluation criteria	Class	Study results Torti -2019	Study results Yu -2020	Experimental results of the proposed method		
				With study approach Musci-2018	Fixed overlap 50%	overlap 95%
Sensitivity	BKG	88.39	89.9	93.19	90.83	94.68
	Alert	91.08	90.33	94.60	77.36	95.53
	Fall	98.73	93.76	94.42	87.33	96.92
Specificity	BKG	97.85	95.05	98.34	93.45	98.93
	Alert	90.77	91.52	94.4	92.05	95.13
	Fall	97.93	98.42	98.82	98.44	99.15

TABLE 6. Comparison of the performance results of the proposed method with the results of some similar studies in the LSTM model

LSTM – Best ACC : 96.61%						
Evaluation criteria	**Class**	**Study results Torti -2019**	**Study results Yu -2020**	**Experimental results of the proposed method**		
				With study approach Musci-2018	**Fixed overlap 50%**	**overlap 95%**
Sensitivity	BKG	88.39	91.5	93.21	sensitivity	BKG
	Alert	91.08	91.48	91.37	77.36	Alert
	Fall	98.73	96.22	91.82	87.33	Fall
Specificity	BKG	97.85	95.93	96.96	specificity	BKG
	Alert	90.77	94	94.39	92.05	Alert
	Fall	97.93	97.54	98.78	98.44	Fall

TABLE 7. Comparison of the performance results of the proposed method with the results of some similar studies in the Conv-Lstm hybrid model

Conv-Lstm – Best ACC : 97.12%						
Evaluation criteria	**Class**	**Study results Torti -2019**	**Study results Yu -2020**	**Experimental results of the proposed method**		
				With study approach Musci-2018	**Fixed overlap 50%**	**overlap 95%**
Sensitivity	BKG	88.39	93.15	92.61	93.06	**96.23**
	Alert	91.08	93.78	94.6	77.57	**98.59**
	Fall	98.73	96	92.07	89.5	**99.38**
Specificity	BKG	97.85	96.59	98.62	93.64	**99.60**
	Alert	90.77	94.49	93.88	93.89	**96.82**
	Fall	97.93	98.69	98.68	98.81	**99.59**

According to Table 7, the accuracy parameter of the proposed method in the Conv-LSTM hybrid model shows a value of 97.12%. The sensitivity parameter values in the proposed method in the BKG, Alert, and Fall classes are 96.23, 98.59, and 99.38, respectively, known as the best in all classes compared to other methods. Also, the values of the specificity parameter in these three classes are 99.60%, 96.82%, and 99.59%, respectively, as the best result compared to other cases. Meanwhile, there is a balance between the results of the classes.

On the other hand, an examination of the three tables, the summary chart of which is presented in Figure 12, shows that the results of the Conv-LSTM hybrid model performed better than the CNN model in all classes. Also, the hybrid model results were better than the LSTM model, and the training time in the hybrid model was much shorter than in the LSTM Network (78 seconds vs. 143 seconds per Epoch).

Therefore, the hybrid model was able to detect the fall with high accuracy, and even the alert class was able to identify with high accuracy, which helps to announce the necessary warning with high confidence before the fall. There was also a balance between the accuracy values of each class in all three models, which could indicate the good quality of the models and good network training with the proposed method. On the other hand, although the number of samples in the fall

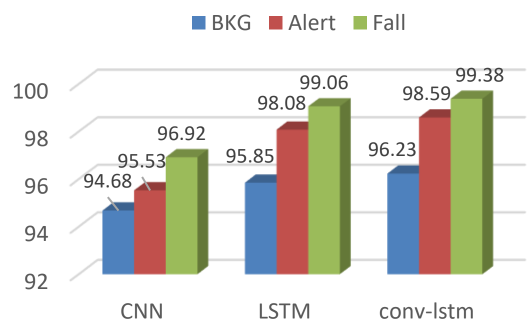


Figure 12. Status diagram of Sensitivity parameter values in three models and in three classes

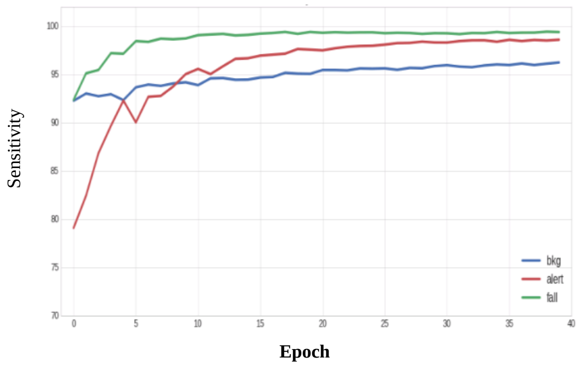
and alert classes was much lower than in the BKG class, a review of all three models shows that higher sensitivity and specificity values were obtained in these classes. This could result from the approach of dynamic overlap in this study.

Therefore, these comparisons show that the hybrid model has performed better with the proposed approach to fall detection using the Sisfall dataset.

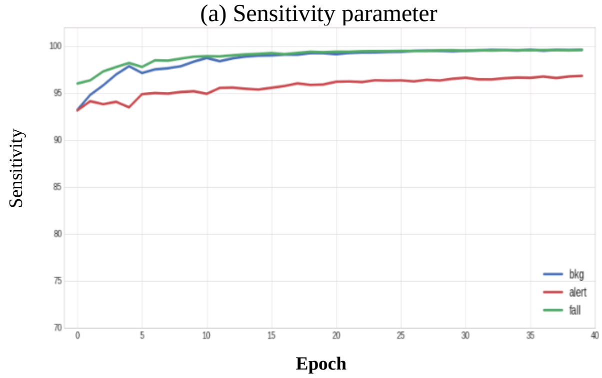
Also, the graphs of sensitivity/specificity values and loss/accuracy parameters related to the hybrid model can be seen in Figures 13 and 14, respectively. The process of improvement of these parameters is clearly observed.

Examining the results obtained in the Confusion Matrix and observing the results of other evaluation criteria such as Precision and F-score, it is observed that in these parameters, high accuracy is provided. These results are presented in Figure 15. As can be seen, the Precision parameter values in all classes are higher than 90%, with an average value of 95.72%. In the F-Score criterion, all classes have values close to 1.

On the other hand, in the continuation of the work, removing noise from the data was performed, for which the first-time low-pass IIR filter was used. After preprocessing and noise removal from the data, the networks were re-implemented on them. The status of the values of the three criteria, Sensitivity, Precision,

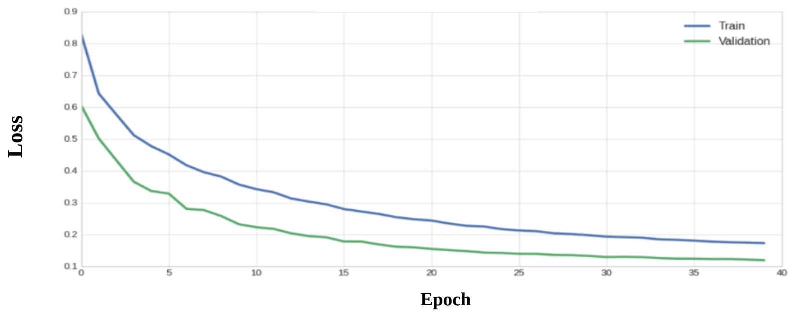


(a) Sensitivity parameter

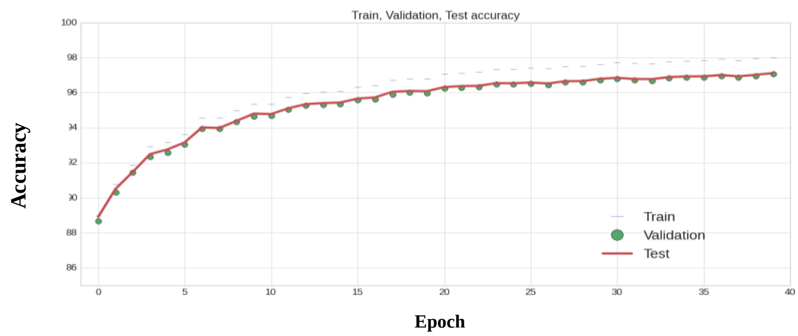


(b) Specificity parameter

Figure 13. Status diagrams of Sensitivity and Specificity parameter values related to the Conv-Lstm hybrid model

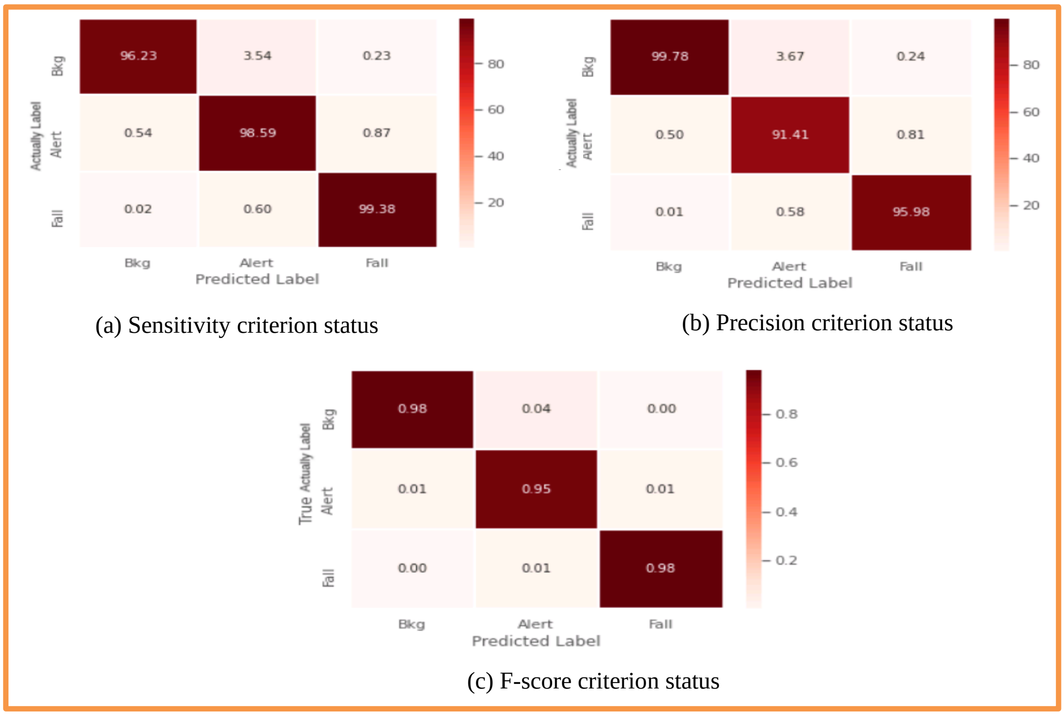


(a) Loss parameter status



(b) Accuracy parameter status

Figure 14. Status diagram of Loss parameter and Accuracy parameter in Conv-Lstm hybrid model



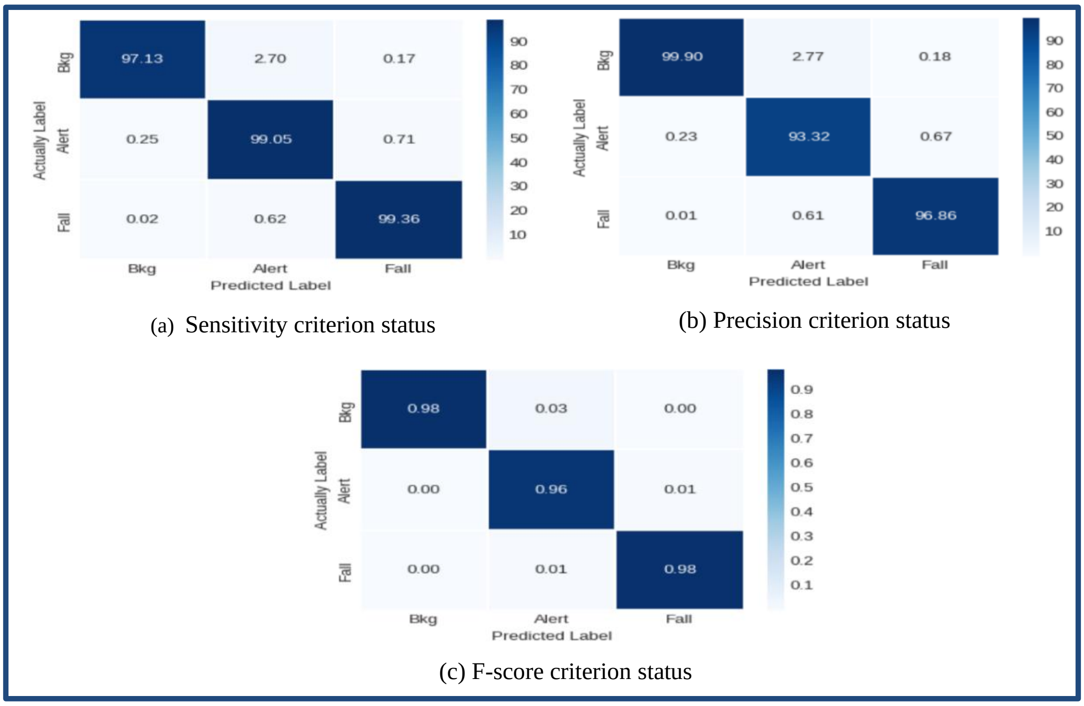
(a) Sensitivity criterion status (b) Precision criterion status

(c) F-score criterion status

Figure 15. Confusion Matrix values in the Conv-Lstm hybrid model

and F-score, based on the Confusion Matrix table, is shown in Figure 16. Based on the results, high accuracy has been achieved in all classes. In the hybrid model, by removing noise from the data, the sensitivity criterion in the three classes of BKG, Alert, and Fall has presented 97.13%, 99.05%, and 99.36%, respectively, which is a relative improvement compared to before.

Also, the results obtained with the smoothed data are compared with the initial data without noise removal, the results of which are shown in Table 8 and their status diagram in Figure 17.



(a) Sensitivity criterion status (b) Precision criterion status

(c) F-score criterion status

Figure 16. Confusion matrix values of the Conv-Lstm hybrid model with Smooth data

TABLE 8. Comparison of performance results of three models, without noise removal and with noise removal (smooth) from the data

º	Class	CNN	CNN_Smooth	LSTM	LSTM_Smooth	Conv-Lstm	Conv-Lstm_Smooth
	BKG	94.68	95.90	95.85	97.44	96.23	97.13
Sensitivity	Alert	95.53	96.73	98.08	98.75	98.59	99.05
	Fall	96.92	96.84	99.06	99.35	99.38	99.36
	BKG	98.93	99.20	99.48	99.65	99.60	99.81
Specificity	Alert	95.13	96.17	97.26	97.70	96.82	97.56
	Fall	99.15	99.38	99.39	99.47	99.59	99.68
Accuracy		94.98	96.11	96.61	97.91	97.12	97.83

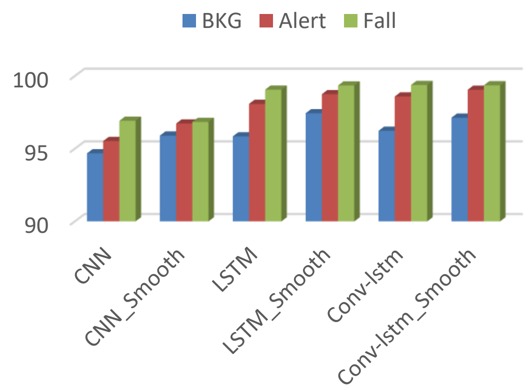


Figure 17. Performance diagram of three models with noise removal from the data and compare it the noise-free mode based on Sensitivity parameter

As shown in Figure 17, noise cancellation from data has improved performance in almost all models and all classes. But among the models, it had the most impact on the CNN model. Also, the BKG and Alert classes were more affected by noise cancellation than the Fall class, which could be due to the number of instances of the classes or more smoothing of the signals of these classes than the Fall class due to filtering.

6. DISCUSSION AND CONCLUSION

The problems of the aging population worldwide are becoming more and more, and the fall is one of the significant health challenges among the elderly and the disabled, which can cause great harm and even death. However, falls can happen to anyone, including young people. Therefore, early detection of falls is one of the essential components of improving the quality of life for people, especially the elderly.

This study presented a framework for identifying fall behavior using deep learning neural networks based on wearable sensor data. Based on this, a three-class Sisfall dataset is used, in which two accelerometer sensors and a gyroscope sensor are used. Studies have shown that accelerometer data are essential for fall detection and are widely used. The data included 34 falls and normal behavior activities performed by 38 participants with a wearable device attached to their waist.

Due to the imbalance between the number of data class samples in data science projects, this study aims to provide a new approach to data sampling and data windowing to increase the accuracy of fall detection. Many experiments were performed, and finally, it was shown that sampling the data with 32 widths and 95% overlap gives the best results.

Three architectures, CNN, LSTM, and the Conv-LSTM hybrid model, have also been applied to the dataset to determine the best model. The results showed that the LSTM and Conv-LSTM models performed better than the CNN model.

Also, in most cases, the Conv-LSTM hybrid model has better performance than the LSTM model, and better results have been obtained based on evaluation criteria. Accordingly, the Conv-LSTM hybrid model in the three classes of BKG, alert, and fall with sensitivity values of 96.23%, 98.59%, and 99.38%, as well as 97.12% in the Accuracy parameter, could provide the best result. Good results have been obtained in other criteria as well.

We also compared our approach with other similar tasks, which, according to the results, were superior to them in all classes.

The noise cancellation process was also performed with a first-time low-pass IIR filter, and retesting showed that it could have a positive effect, especially on the CNN model, and also improved the accuracy of the Fall and Alert classes. In the combined model, the sensitivity parameter values for the BKG, alert, and fall classes are 97.13%, 99.05%, and 99.36%, respectively. For the Accuracy parameter, an accuracy of 97.83% has been obtained.

Therefore, the Conv-LSTM hybrid model with filtered data can provide the best accuracy in all classes and in all evaluation criteria.

The proposed approach for dynamic data sampling led to a more excellent balance between the number of samples in different classes. This has increased accuracy and reduced false alarms. Also, the combination of CNN and LSTM algorithms and using the advantages of these two algorithms have greatly improved the accuracy of fall detection.

However, among the various methods, there is no clear evaluation framework. So it is a bit difficult to evaluate and compare the results somewhat. Lack of access to actual data was one of the limitations of this study, which required us to use datasets with simulated movements in laboratories. This may reduce the accuracy of the results of these systems in real life.

On the other hand, parameters such as age, gender, height, and weight of people and external factors affecting the signals are also effective in the accuracy of fall detection. The impact of these parameters can be seriously considered in future studies of fall detection systems.

7. REFERENCES

1. Florence, C.S., Bergen, G., Atherly, A., Burns, E., Stevens, J. and Drake, C., "Medical costs of fatal and nonfatal falls in older adults", ***Journal of the American Geriatrics Society***, Vol. 66, No. 4, (2018), 693-698. doi: 10.1111/jgs.15304.
2. Qiu, H. and Xiong, S., "Center-of-pressure based postural sway measures: Reliability and ability to distinguish between age, fear of falling and fall history", ***International Journal of Industrial Ergonomics***, Vol. 47, (2015), 37-44. doi: 10.1016/j.ergon.2015.02.004.
3. Organization, W.H., Ageing, W.H.O. and Unit, L.C., "Who global report on falls prevention in older age, World Health Organization, (2008).
4. Zhang, D., Wang, H., Wang, Y. and Ma, J., "Anti-fall: A non-intrusive and real-time fall detector leveraging csi from commodity wifi devices", in International Conference on Smart Homes and Health Telematics, Springer., (2015), 181-193.
5. Santos, G.L., Endo, P.T., Monteiro, K.H.d.C., Rocha, E.d.S., Silva, I. and Lynn, T., "Accelerometer-based human fall detection using convolutional neural networks", ***Sensors***, Vol. 19, No. 7, (2019), 1644. doi: 10.3390/s19071644.
6. Casilari, E., Lora-Rivera, R. and García-Lagos, F., "A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets", ***Sensors***, Vol. 20, No. 5, (2020), 1466. doi: 10.3390/s20051466.
7. Pannurat, N., Thiemjarus, S. and Nantajeewarawat, E., "Automatic fall monitoring: A review", ***Sensors***, Vol. 14, No. 7, (2014), 12900-12936. doi: 10.3390/s140712900.
8. Xu, T., Zhou, Y. and Zhu, J., "New advances and challenges of fall detection systems: A survey", ***Applied Sciences***, Vol. 8, No. 3, (2018), 418. doi: 10.3390/app8030418.
9. Chen, X., Xue, H., Kim, M., Wang, C. and Youn, H.Y., "Detection of falls with smartphone using machine learning technique", in 2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE., (2019), 611-616.
10. Özdemir, A.T., "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice", ***Sensors***, Vol. 16, No. 8, (2016), 1161. doi: 10.3390/s16081161.
11. Ren, L. and Peng, Y., "Research of fall detection and fall prevention technologies: A systematic review", ***IEEE Access***, Vol. 7, (2019), 77702-77722. doi: 10.1109/ACCESS.2019.2922708.
12. Liang, S., Chu, T., Lin, D., Ning, Y., Li, H. and Zhao, G., "Pre-impact alarm system for fall detection using mems sensors and hmm-based svm classifier", in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE., (2018), 4401-4405.
13. Wu, Y., Su, Y., Feng, R., Yu, N. and Zang, X., "Wearable-sensor-based pre-impact fall detection system with a hierarchical classifier", ***Measurement***, Vol. 140, (2019), 283-292. doi: 10.1016/j.measurement.2019.04.002.
14. Aziz, O., Musngi, M., Park, E.J., Mori, G. and Robinovitch, S.N., "A comparison of accuracy of fall detection algorithms (threshold-based vs. Machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials", ***Medical & Biological Engineering & Computing***, Vol. 55, No. 1, (2017), 45-55. doi: 10.1007/s11517-016-1504-y.
15. Özdemir, A.T. and Barshan, B., "Detecting falls with wearable sensors using machine learning techniques", ***Sensors***, Vol. 14, No. 6, (2014), 10691-10708. doi: 10.3390/s140610691.
16. Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L., "Deep learning for sensor-based activity recognition: A survey", ***Pattern Recognition Letters***, Vol. 119, (2019), 3-11. doi: 10.1016/j.patrec.2018.02.010.
17. LeCun, Y., Bengio, Y. and Hinton, G., "Deep learning", ***nature***, Vol. 521, No. 7553, (2015), 436-444. doi: 10.1038/nature14539.
18. Ordóñez, F.J. and Roggen, D., "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition", ***Sensors***, Vol. 16, No. 1, (2016), 115. doi: 10.3390/s16010115.
19. Chen, Y. and Xue, Y., "A deep learning approach to human activity recognition based on single accelerometer", in 2015 IEEE international conference on systems, man, and cybernetics, IEEE., (2015), 1488-1492.
20. Tao, X. and Yun, Z., "Fall prediction based on biomechanics equilibrium using kinect", ***International Journal of Distributed Sensor Networks***, Vol. 13, No. 4, (2017), doi: 10.1177/1550147717703257.
21. Torti, E., Fontanella, A., Musci, M., Blago, N., Pau, D., Leporati, F. and Piastra, M., "Embedding recurrent neural networks in wearable systems for real-time fall detection", ***Microprocessors and Microsystems***, Vol. 71, (2019), 102895. doi: 10.1016/j.micpro.2019.102895.
22. Musci, M., De Martini, D., Blago, N., Facchinetti, T. and Piastra, M., "Online fall detection using recurrent neural networks", arXiv preprint arXiv:1804.04976, (2018). doi: 10.48550/arXiv.1804.04976.
23. Sucerquia, A., López, J.D. and Vargas-Bonilla, J.F., "Sisfall: A fall and movement dataset", ***Sensors***, Vol. 17, No. 1, (2017), 198. doi: 10.3390/s17010198.
24. Rastogi, S. and Singh, J., "A systematic review on machine learning for fall detection system", ***Computational Intelligence***, Vol. 37, No. 2, (2021), 951-974. doi: 10.1111/coin.12441.
25. Usmani, S., Saboor, A., Haris, M., Khan, M.A. and Park, H., "Latest research trends in fall detection and prevention using machine learning: A systematic review", ***Sensors***, Vol. 21, No. 15, (2021), 5134. doi: 10.3390/s21155134.
26. Syed, A.S., Sierra-Sosa, D., Kumar, A. and Elmaghraby, A., "A deep convolutional neural network-xgb for direction and severity aware fall detection and activity recognition", ***Sensors***, Vol. 22, No. 7, (2022), 2547. doi: 10.3390/s22072547.
27. Bourjandi, M., Yadollahzadeh Tabari, M. and Golsorkhtabaramiri, M., "Fuzzy centralized coordinate learning

and hybrid loss for human activity recognition", ***International Journal of Engineering, Transactions A: Basics,***, Vol. 35, No. 1, (2022), 130-141. doi: 10.5829/IJE.2022.35.01A.12.

28. Siddharth, D., Saini, D. and Singh, P., "An efficient approach for edge detection technique using kalman filter with artificial neural network", ***International Journal of Engineering, Transactions C: Aspects,***, Vol. 34, No. 12, (2021), 2604-2610. doi: 10.5829/IJE.2021.34.12C.04.
29. Bourjandi, M., Yadollahzadeh-Tabari, M. and Golsorkhtabaramiri, M., "Combined deep centralized coordinate learning and hybrid loss for human activity recognition", ***Concurrency and Computation: Practice and Experience***, (2022), e6870. doi: 10.1002/cpe.6870.
30. Bourjandi, M., Yadollahzadeh-Tabari, M. and GolsorkhtabariAmiri, M., "Predicting user's movement path in indoor environments using the stacked deep learning method and the fuzzy soft-max classifier", ***IET Signal Processing***, (2022). doi: 10.1049/sil2.12125.
31. Yu, X., Qiu, H. and Xiong, S., "A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors", ***Frontiers in Bioengineering and Biotechnology***, Vol. 8, (2020), 63. doi: 10.3389/fbioe.2020.00063.

Persian Abstract

چکیده

سقوط یکی از مهم‌ترین چالش‌های سلامتی در جامعه است که می‌تواند باعث صدمات شدید و حتی مرگ افراد شود. هدف اصلی این مطالعه، توسعه شبکه‌های عصبی عمیق با استفاده از داده‌های سنسورهای پوشیدنی برای شناسایی سقوط می‌باشد. اکثر مجموعه داده‌ها در این زمینه از مشکل عدم توازن رنج می برند به طوری که نمونه‌های متعلق به کلاس‌های Fall به طور قابل توجهی کمتر از داده‌های کلاس عادی هستند. این مطالعه یک تکنیک نمونه گیری پویا برای افزایش نرخ تعادل بین نمونه های متعلق به کلاس های سقوط و عادی ارائه می دهد تا دقت الگوریتم های یادگیری را بهبود بخشد. از مجموعه داده Sisfall استفاده شده است که در آن، فعالیت انسان‌ها به سه دسته فعالیت عادی(BKG) ، لحظات قبل از سقوط (Alert) و نقش روی زمین (Fall) تقسیم می شود. سه مدل یادگیری عمیق CNN، LSTM و یک مدل ترکیبی به نام-Conv Lstmبر روی این مجموعه داده پیاده‌سازی شدند و عملکرد آنها مورد ارزیابی قرار گرفت. بر این اساس، مدل ترکیبی Conv-Lstm مقادیر ۹۶.۲۳٪، ۹۸.۵۹٪ و ۹۹.۳۸٪ را در پارامتر Sensitivity برای کلاس‌هایBKG، Alert و Fall ارائه می‌کند. برای پارامتر Accuracy نیز موفق شدیم به نرخ ۹۷.۱۲٪ برسیم. علاوه بر این، با استفاده از تکنیک‌های هموارسازی و حذف نویز، می‌توان به میزان دقت ۹۷.۸۳ درصد نیز رسید. نتایج حاکی از برتری مدل پیشنهادی نسبت به سایر مطالعات مشابه دارد.