Proceeding of the 25^{th} International Conference
Automation & Computing, Lancaster University,
Lancaster UK, 5-7 September 2019

Using Machine Learning Techniques to Optimize Fall Detection Algorithms in Smart Wristband

Ge Zheng[1], Hongtao Zhang[2], Keming Zhou[2], Huosheng Hu[3]
[1]Department of Computing and Informatics, Bournemouth University, Dorset, BH12 5BB, U.K.
[2]August International Limited, Rawmec Business Park, Plumpton Road, Hoddesdon, EN11 0EE, U.K.
[3]Department of Computer Science and Electronics Engineering, University of Essex, Colchester CO4 3SQ, U.K.

***Abstract*—The consumer electronics market is already saturated with wearable devices that intend to be used to detect falls and request help from carers or family members. However, these products have a high rate of false alarms which affect their reliable performance. To provide the high accuracy and high precision of fall detection for the elderly, this paper presents a machine learning approach to improve the fall detection accuracy and reduce the false alarms. Three machine learning algorithms are deployed in this research, namely the K-Means, Perceptron Neural Network (PNN), and Convolutional Neural Network (CNN) algorithms. A development board with a 9-axis inertial sensor unit is used as a prototype of wristband to collect data and identify falls from seven daily activities. These data is then used to train and test machine learning algorithms. Experimental results show that the CNN algorithm achieves the highest accuracy comparing with K-mean, PNN and the algorithm used in the existing wristbands.**

***Index Terms*—Fall Detection, MPU6050, Accelerometer, Gyroscope, Machine Learning**

I. Introduction

As reported by the 2017 Population Division [1], the population in Europe consists of 25% being 60 years or older, that proportion will reach 35% in 2050 and 36% in 2100. In other regions, the population is also expected to age significantly over the next several decades. Coupled with aging tendency of the population, health problems among the elderly over 65 years old; such as fall, high blood pressure, and diabetes; are becoming serious. Based on the health problems described above, it is clear that being able to prevent falling, or providing timely assistance in the event of a fall for the elderly people is greatly beneficial.

In general, fall detection systems can be divided into three categories; (i) fall detection with accelerometer, (ii) fall detection with gyroscope and (iii) fall detection with accelerometer and gyroscope.

- Fall detection with accelerometer: Razum and Seketa [2] used a acceleration sensor unit Shimmer3 to collect data for fall detection. They applied two simple threshold-based algorithms consisting of Euler Angle & Sum Vector Magnitude (THETA&SVM) and SVM&THETA. Chen [3] developed a wearable device worn around the waist without obstructing normal activities, in which an accelerometer sensor was used based on a threshold method. A fall detection monitoring system was designed in [4], aimed to reduce the time of emergency response by using an accelerometer. Moreover, a fall detection device with an 3D accelerometer was developed in [5], in which a threshold method was used to detect the fall from normal activities such as standing, sitting and supine position. In [6], an IoT-based system was developed for detecting falls, which consisted of a microcontroller, accelerometer and Wireless Sensor Network (WSN). The device in [6] can continuously work for 35 hours.
- For fall detection with gyroscope: Bourke and Lyons built a fall detection system using only a gyroscope, which aimed to distinguish falls from daily activities [7]. In this system, the fall detection algorithm depends on three threshold values which are a resultant angular velocity, a resultant angular acceleration and a resultant change in a trunk-angle. The results indicated that the accuracy and specificity of this method was 100% successful.
- For fall detection with accelerometer and gyroscope: A wearable device with tri-axial accelerometer and gyroscope was developed to record eight types of falls and six types of daily activity, and then the total sum acceleration and the total sum angular velocity were compared to predetermined thresholds for a fall decision [8]. Moreover, two separate sub-sensing-devices worn on left and right wrists were created to detect falls based on a threshold method [9]. A three-axis accelerometer and a three-axis gyroscope were included in each sub-sensing-device and used to collect data. On the other hand, a MPU6050 was used to collect data and the threshold method was utilised to decide a fall event [10].

Most of these products, however, have limitations such as failing to detect a fall in a certain situation, delaying response times, uncomfortable for the user to wear and only detecting falls indoors. Therefore, it becomes necessary to develop accurate fall detection algorithm to be used in wearable devices to improve their accuracy and reduce the rate of false alarm. This paper aims to use machine learning techniques to develop new fall detection algorithms for smart wristbands and make them adaptive to individual users so that the emergency assistance can be provided immediately when a fall is detected.

The rest of this paper is organized as follows. Section II describes the configuration of our fall detection device in terms of sensor selection, development board and software. Then, in Section III, we describe our experiment settings including data collection and data preparation. The experimental results and

This research is financially supported by
Innovate UK and August International Limited
under the Knowledge Transfer Partnership grant (KTP009841).

analyses are given in Section IV to show the performance of our proposed algorithms in comparison with other existing fall detection algorithms. Finally, a brief conclusion and future work are presented in Section V.

II. Configuration of Fall Detection Device

A. Sensor Selection

In this paper, both accelerometer and gyroscope will be used to characterize fall behaviours so that more parameters and features can be used in decision making. In terms of hardware, we select the MPU6050 [11] to build our wearable fall detection device, which can automatically remove the cross talk between accelerator and gyroscope and reduce the impact of the settling and the error drift. It also has low power consumption, high speed data sampling and full scale range of acceleration and gyroscope, which is similar to MPU9250 [12] and LSM6DS3 [13]. Its another advantage is that it has a digital motion processing (DMP) engine which can reduce complex loading data, sensor synchronization, posture sensing and other computational requirements. Its motion processing database supports the operating time deviation inherent with the Android, Linux and Windows operating systems and the magnetic sensor correction calculus, which eliminates the need for additional correction by the customer.

B. Development Board for Data Collection

Figure 1 shows the development board used for collecting data. It consists of an MPU6050 [11] module with accelerometer sensor and gyroscope sensor, Microchip PIC microcontroller, I2C EEPROM [14], RS485/422 Driver [15], in-circuit-serial-programming (ICSP) [16], power connection and serial connection. The programming code is loaded into microcontroller via the ICSP [16] interface. On operation, the program first interrogates the I2C EEPROM [14] to obtain stored configuration and calibration data, allowing it to initialize the MPU6050 [11] module.

The program then collects the data sent from the MPU6050 [11] module to the microcontroller via the I2C bus which the MPU6050 [11] module has sampled from its internal accelerometer and gyroscope sensor. The microcontroller sends the processed data to the RS485/422 Driver [15], and the data is then relayed to the computer terminal via USB connection from the RS485/422 Driver [15]. Figure 2 details the data collection process of the development board. A sample rate of 20 Hz is used to collect data in this paper. In other words, the module can collect 20 full set of data per second. Every data set consists of acceleration in x-y-z-axis and gyroscope in x-y-z-axis, and an example is displayed in Figure 3 below.

C. Three Machine Learning Algorithms

We have used three machine learning algorithms in this research in order to compare their performance and the improvement over the existing fall detection algorithms. A brief introduction of these algorithms is given below.

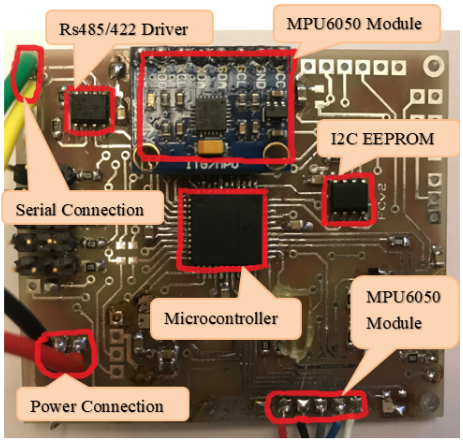


Fig. 1: Development board

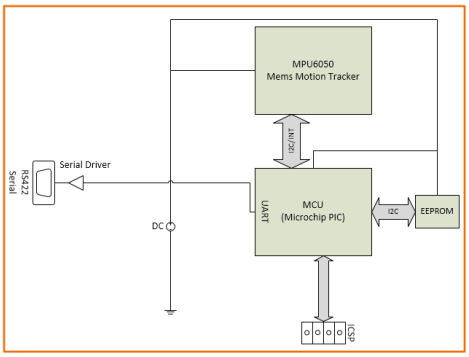


Fig. 2: Data collection process

1) K-Means Fall Detection Algorithm: The K-Means algorithm is the simplest and most practical one available [17]. Three datasets (M-dataset, V-dataset, MV-dataset) supplied to the K-Means algorithm are divided into two groups and labelled either fall (1) or non-fall (0). The one group with the bigger center value responds to the predicted fall events, and the other group responds to the predicted non-fall events.

2) PNN Fall Detection Algorithm: Perceptron Neural Network is the basic unit of an artificial neural network, which consists of multiple inputs and one output [18]. The PNN algorithm, as a linear classifier, is trained using all data including fall data and non-fall data, and also tested using the same dataset. Datasets used for deciding an fall event using this algorithm are same as datasets used in the K-means algorithm.

3) CNN Fall Detection Algorithm: Convolutional neural networks (CNNs) are widely used in image recognition, and able to achieve high accuracy, compared to other techniques [19]. In this paper, we adopt Method 1 (CNN-dataset1) and Method 2 (CNN-dataset2) to collect data related to fall and other daily activities in order to train and test the CNN algorithms for fall events.

III. Data Collection and Preparation

All data related to daily activities, such as walking, running, sitting, jumping, stooping, hand clapping and up-down stair

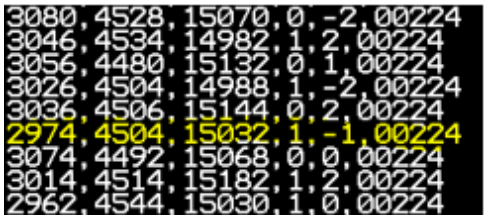


Fig. 3: Data units.

movements, have been collected. A fall event can be decided by comparing the different data characteristics between fall events and daily activity events. Figure 4 shows the data collection process when a person wearing our smart wearable device.

Fig. 4: Data collection using wearable device

We adopt two different methods to collect data in this research. The first is applied for the K-means, PNN, and CNN algorithms. The second is only applied for the CNN algorithm.

A. Method 1

To detect a fall, we need to select a time interval to cover the whole fall event. After many experiments, it was found that a time interval of 6 seconds was the most efficient way of catching a fall event. As 20 data units with 6-dimension were collected every second, we use 120 data sets as a sample to determine a fall event. Finally, 1160 samples were collected for experiments. 50% of samples were from falling and the rest of 50% was from other seven daily activities. Figure 5 shows the related raw data representing falling, hand clapping, jumping, running, sitting, stooping, stair traversing and walking.

B. Method 2

This method was used to collect data used for the CNN algorithm only. The CNN algorithm is mainly used to solve image recognition problems, e.g. to recognize handwriting in MNIST dataset. We reshape our datasets to simulate pixel based image data. Considering the size of pixel value and samples, more data needed to be collected in order to successfully determine a fall event. As the time lying on the ground after falling may be long, adequate time duration must be incorporated into the dataset to determine a fall.

(a) Falling and Hand clapping

(b) Jumping and Running

(c) Sitting-down and Stooping

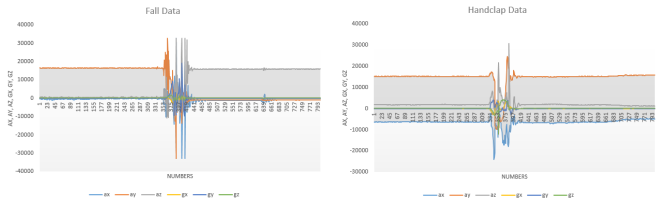
(d) Up-down stairs and Walking

Fig. 5: Eight types of daily activity data collected using Method 1.

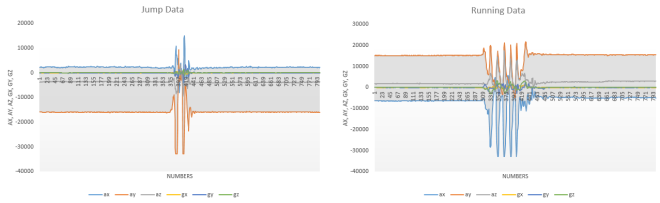
The classification method used to identify the different activities is same as one used in Method 1. However, every type of data consisted of 600 data sets collected over a period of 30 seconds. It means data collected from a period of 30 seconds as a sample was used to decide an event. Finally, 5800 samples were used for this experiments, in which 50% of samples was from falling and the rest of 50% was from other seven daily activities. Figure 6 shows the related raw data including falling, hand-clapping, jumping, running, sitting, stooping, stair traversing and walking.

For the K-Means and PNN algorithms, data was collected using Method 1. As they are able to manipulate two-dimension data, the collected data from MPU6050 sensor consisting of six-dimensions was processed using the dimension reduction algorithm that is the process of reducing the number of variables by extracting its principal ones. It is also called feature extraction. There were two features extracted to fit K-Means and PNN algorithms, which were mean and variance. Finally, three datasets were integrated as inputs for the K-means and PNN algorithms, namely M-dataset, V-dataset, MV-dataset. M-dataset with two dimensions describes the features of the means of acceleration and gyroscope. V-dataset describes the features of the variances of acceleration and gyroscope, and MV-dataset presents the features of the means of acceleration and gyroscope and the variances of acceleration and gyroscope.

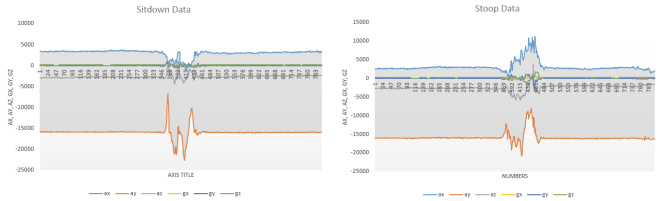
For the CNN algorithm, data was collected using both



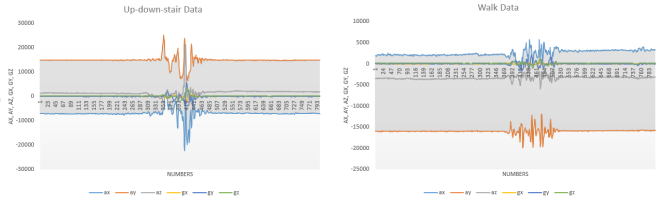
(a) Falling and Hand clapping



(b) Jumping and Running



(c) Sitting-down and Stooping



(d) Up-down stairs and Walking

Fig. 6: Eight types of daily activity data collected using Method 2.

Method 1 and Method 2. For data collected using Method 1, it was integrated into CNN-dataset1 and its shape was (1160 * 120 * 6) in which 1160 means the number of samples, and 120 * 6 means 120 data sets with 6 dimensions. For data collected using Method 2, it was integrated into CNN-dataset2 with the shape of (5800 * 600 * 6). 5800 means the number of samples, and 600 * 6 means 600 data units with 6 dimensions.

IV. Results and Discussion

A. Evaluation Metrics

Three different algorithms (K-Means, PNN, and CNN) were evaluated using the same parameters: accuracy rate, recall rate, specificity rate, and precision rate. The formulas used to calculate these are:

- True Positive (TP): The model predicts positive samples as positive samples.
- True Negative (TN): The model predicts negative samples as negative samples.
- False Positive (FP): The model predicts negative samples as positive samples.
- False Negative (FN): The model predicts positive samples as negative samples.
- Accuracy Rate = (TP + TN) / (TP + FN + FP +TN)
- Recall Rate = TP / (TP + FN)
- Specificity Rate = TN / (TN + FP)
- Precision Rate = TP / (TP + FP)

B. K-Means Fall Detection Algorithm

Figure 7 presents the classification result from the K-Means algorithm using M-dataset with acceleration and gyroscope mean value in 6 seconds. The cluster with deep purple was predicted as fall events and the cluster with yellow was predicted as non-fall events. Samples were divided into two groups. One group with the maximum center was predicted as fall events, and the other group was predicted as non-fall events.

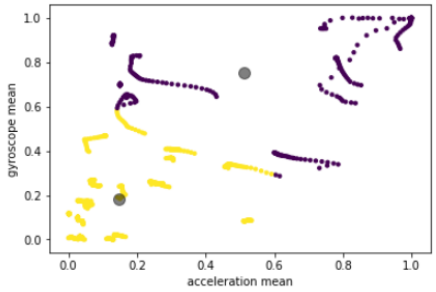


Fig. 7: Results of the K-means algorithm using M-dataset.

Figure 8 and Figure 9 demonstrate the experimental results of the K-Means algorithm applied in datasets prepared using Method 1. The results indicate that approximately 50% accuracy and precision rates were achieved by the K-Means algorithm using three different datasets with 2-dimensions. The best one was obtained by the K-Means algorithm using V-dataset consisting of the acceleration variance and the gyroscope variance which show the change between every sample data and the mean value of all samples.

However, The accuracy and recall rates of the K-Means algorithm using these two feature values to detect falls were low. It suggests it would not be reliable if it is applied in a healthcare system to help the elderly. One possible reason resulting in the lower accuracy rate is that there is not enough features to represent a fall event, and the other possible reason is that noisy data was not filtered before it was input into the K-Means algorithm.

C. PNN Fall Detection Algorithm

Identical feature values used in K-Means algorithm were also used in the PNN algorithm. 50% of the dataset (580 samples) was used to train PNN, and the other 50% was used to test the PNN algorithm. Figure 10 visualizes the classification result from the PNN algorithm trained 200 times using data representing acceleration and gyroscope mean value for 6 seconds. The data drawn with blue marks represents real

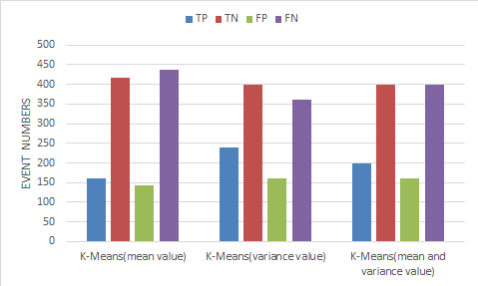


Fig. 8: The event numbers of the predicted TP, TN, FP and FN on three datasets

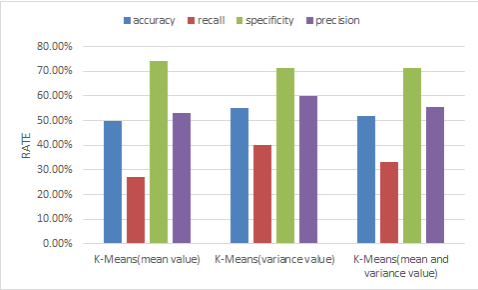


Fig. 9: The rates of accuracy, recall, specificity and precision on three datasets

fall data, and the data drawn with pink marks represents non-fall data. The data in blue area are predicted as falls.

Results using different epochs, from 100 to 1000, were achieved on three datasets. As a result, we found epochs to obtained the best results on M-dataset, V-dataset and MV-dataset were respectively 200, 400 and 300, as shown in TABLE I. As can be seen, there were 80% of samples to be recognized correctly on M-dataset. However, roughly 68% of samples were recognized correctly on MV-dataset followed by 64.66% on V-dataset. This means that the PNN algorithm on M-dataset had good ability to recognize fall events and non-fall event. Moreover, the PNN algorithm on V-dataset had better ability to recognize non-fall events than to recognize fall events. On MV-dataset, all evaluation parameters were about 68%.

It is clear that the PNN algorithm has achieved the higher accuracy and precision rates, compared to the K-Means algorithm. However, it has the same problems presented in the K-Means algorithm, such as noisy input data and fewer features to replace original data. Possible solutions are the implementation of filter algorithms in the data pre-processing stage and extending the data dimension to smooth the original data and avoid losing detailed information.

D. CNN Fall Detection Algorithm

TABLE II shows the results documented using 6-seconds data (CNN-dataset1) to train the CNN algorithm with between 2 and 5 layers at different training times ranging from 10 to 50. 50% of all data (1160 samples) was used to train the algorithm, and the other 50% was used to test it. As can be seen, the CNN

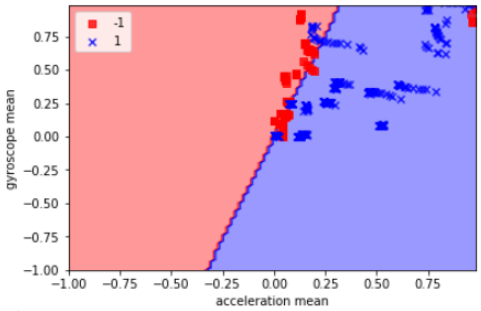


Fig. 10: Result of the PNN algorithm on M-dataset

algorithm achieved better result with the increase of the layers. To achieve the best performance, the CNN algorithms with 2 layers, 3 layers, 4 layers and 5 layers, needed to be trained 30, 50, 40 and 50 times. The algorithm with 5 layers applied on CNN-dataset1 had best performance for fall detection, and the accuracy was 99%. Note that the performances of the CNN algorithms with different layers just had small differences.

TABLE III shows the results from using CNN-dataset2 (consisting of 5600 samples) in the CNN algorithms with different layers in which every data sample was collected in 30 seconds. 50% of CNN-dataset2 was used to train the CNN algorithm, and the other 50% was used to test it. To find best result, epochs were used to train our algorithm from 1 to 50 with 5 step as a gap. We found the epochs to provide the best performance of the CNN algorithms with 2 layers, 3 layers , 4 layers and 5 layers were 30, 40, 50 and 40 respectively.

The CNN algorithm with 5 layers applied on CNN-dataset2 had best performance for fall detection, and it just had 3 fall events to be predicted as non-fall events and 1 non-fall event to be predicted as fall event. Related accuracy, recall, specificity and precision rates were more than 99%. Compared to the CNN algorithm using CNN-dataset1 with a sample data collection in 6 seconds, the CNN algorithm using CNN-dataset2 (a sample data collection in 30 seconds) to train presents the best performance in fall detection. More data means more accuracy and more detailed information.

V. Conclusion and Future Work

In this paper, we deployed three machine learning algorithms to improve fall detection software, namely the K-Means, PNN, and CNN algorithms. Two methods were used to prepare data for our algorithms. Method 1 was to collect data in 6 seconds as a sample to decide a fall event, and Method 2 was to collect data in 30 seconds as a sample to decide a fall event. The data preparation in Method 1 implemented on three algorithms, but the data preparation in Method 2 was only used in the CNN algorithm. The accuracy rates of fall detection from the K-Means, PNN, and CNN algorithms were approximate 50%, 70%, and 99%, respectively. It is clear that the CNN algorithm obtained the highest accuracy rate, compared to other two algorithms.

In the feature work, more data will be collected for the CNN algorithm. Moreover, we will also explore ensemble

TABLE I: Results of the PNN Algorithm on Three Datasets

Datasets	Epochs	TP	TN	FP	FN	Accuracy(%)	Recall(%)	Specificity(%)	Precision(%)
M-dataset	200	269	207	41	63	82.07	81.02	83.47	86.77
V-dataset	400	262	113	41	164	64.66	61.50	73.38	86.47
MV-dataset	300	196	200	92	92	68.28	68.06	68.49	68.06

TABLE II: Results of the CNN Algorithm on CNN-Dataset1

Layers	Epochs	TP	TN	FP	FN	Accuracy(%)	Recall(%)	Specificity(%)	Precision(%)
2	30	302	271	4	3	98.79	99.02	98.55	98.69
3	50	306	267	1	6	98.79	98.08	99.63	99.67
4	40	295	279	3	3	98.97	98.99	98.94	98.99
5	50	308	266	0	6	99.00	98.09	100.00	100.00

TABLE III: Results of the CNN Algorithm on CNN-Dataset2

Layers	Epochs	TP	TN	FP	FN	Accuracy(%)	Recall(%)	Specificity(%)	Precision(%)
2	30	1492	1398	0	10	99.66	99.33	100.00	100.00
3	40	1497	1396	0	7	99.76	99.54	100.00	100.00
4	50	1521	1372	7	0	99.76	100.00	99.49	99.54
5	40	1503	1393	1	3	99.86	99.80	99.93	99.93

algorithms which combine several simple algorithms such as the K-NN algorithm with the CNN algorithm for fall detection. In addition, this paper used data consisting of acceleration and gyroscope. Considering the power consumption of the gyroscope sensor, we will do an evaluation on the system using only one accelerometer to collect data for fall detection.

ACKNOWLEDGMENTS

Our thanks go to Mr Robin Dowling and Mr Ian Dukes for their technical support during the experiment.

REFERENCES

[1] U. D. of Economic and S. Affairs, *World Population Prospects: 2017 Revision-Key Findings and Advance Tables*, 16 June 2017. [Online]. Available: https://reliefweb.int/report/world/world-population-prospects-2017-revision-key-findings-and-advance-tables

[2] G. Šeketa, J. Vugrin, and I. Lacković, "Optimal threshold selection for acceleration-based fall detection," in *Precision Medicine Powered by pHealth and Connected Health*. Springer, 2018, pp. 151–155.

[3] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, 2006, pp. 3551–3554.

[4] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao, "Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics," *IEEE Access*, vol. 6, pp. 9324–9335, 2018.

[5] A. Jefiza, E. Pramunanto, H. Boedinoegroho, and M. H. Purnomo, "Fall detection based on accelerometer and gyroscope using back propagation," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE, 2017, pp. 1–6.

[6] T. N. Gia, M. Jiang, V. K. Sarker, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Low-cost fog-assisted health-care iot system with energy-efficient sensor nodes," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1765–1770.

[7] A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical engineering & physics*, vol. 30, no. 1, pp. 84–90, 2008.

[8] H. W. Guo, Y. T. Hsieh, Y. S. Huang, J. C. Chien, K. Haraikawa, and J. S. Shieh, "A threshold-based algorithm of fall detection using a wearable device with tri-axial accelerometer and gyroscope," in *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, Nov 2015, pp. 54–57.

[9] S. Hsieh, C. Chen, S. Wu, and T. Yue, "A wrist -worn fall detection system using accelerometers and gyroscopes," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, April 2014, pp. 518–523.

[10] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544–4553, Aug 2015.

[11] InvenSense, *MPU-6000 and MPU-6050 Product Specification Revision 3.4*, 19 Aug. 2013. [Online]. Available: https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3\%204.pdf

[12] ——, *MPU-9250 Product Specification Revision 1.1*, 20 June 2016. [Online]. Available: https://www.invensense.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

[13] life.augmented, *iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope*, September 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6ds33.pdf

[14] ——, *M24C64-W M24C64-R M24C64-F M24C64-DF 64-Kbit serial IC bus EEPROM*, March 2018. [Online]. Available: https://www.st.com/resource/en/datasheet/m24c64-df.pdf

[15] T. INSTRUMENTS, *5-V FULL-DUPLEX RS-485/RS-422 DRIVER AND BALANCED RECEIVER*, July 2008. [Online]. Available: http://www.ti.com/lit/ds/slls668c/slls668c.pdf

[16] WIKIPEDIA, *In-system programming*, March 2019. [Online]. Available: https://en.wikipedia.org/wiki/In-system_programming

[17] L. Morissette and S. Chartier, "The k-means clustering technique: General considerations and implementation in mathematica," *Tutorials in Quantitative Methods for Psychology*, vol. 9, no. 1, pp. 15–24, 2013.

[18] S. S. Haykin, S. S. Haykin, S. S. Haykin, K. Elektroingenieur, and S. S. Haykin, *Neural networks and learning machines*. Pearson Upper Saddle River, 2009, vol. 3.

[19] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.