AI-Based Fall Detection Using Contactless Sensing

Ahmad Taha †, Mohammad M.A. Taha ‡, Basel Barakat §, William Taylor †,
Qammer H. Abbasi †, and Muhammad Ali Imran †

†James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK
‡Independent Scholar, New Hampshire, US
§School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh, EH10 5DT, UK
ahmad.taha@glasgow.ac.uk, mohammad.m.a.taha@gmail.com, 40507310@live.napier.ac.uk, 2536400T@student.gla.ac.uk,
Qammer.Abbasi@glasgow.ac.uk, Muhammad.Imran@glasgow.ac.uk

***Abstract*—Falls are a major health concern for the elderly as it threatens their livelihood and independence. Nearly 50% of the older adults, aged over 65 years old, fall in a span of 5 years, with 62% sustaining injuries and 28% extensive protracting injuries. This paper presents a high accuracy contactless falls detection framework based on channel state information extracted from software-defined radios. The aim is to develop a system capable of detecting whether an individual subject is present within the sensing area, or if the subject is falling, and, finally, if the subject is performing one of three other activities, including sitting, standing, and walking. The results showed a promising detection accuracy of 95.6% and 98%, using the 10-fold cross-validation and train-test split methods, based on the Random Forest classifier, respectively. Furthermore, we present a real-time analysis of the system to highlight its capability to detect, analyze, and report falls in real-time.**

***Index Terms*—Falls detection, Channel State information, Machine learning, Random Forest**

I. Introduction

Falls are the second leading cause of unintentional injury deaths worldwide [1], as well as a major cause of distress, pain, injury, loss of confidence and loss of independence. In England, around a third of people aged 65+ and around half of the people aged 80+ fall at least once a year [2].

Detecting the falls can be life-saving, especially if the person becomes unconscious or immobilized. Several technologies have been investigated for fall detection, such as multi-sensor-based [3], radar-based [4], wearables [5], [6], vision-based [7]–[9], and using *Channel State Information* (CSI) [10] systems. However, some drawbacks and limitations exist that needs to be addressed. For example, vision-based systems raise several privacy challenges. Wearable systems have to be on the user, which can be restricting and uncomfortable, and radar-based systems can be complicated and costly to install. Other systems such as [10] developed a fall detection system based on CSI; however, their system relied on the use of multiple transceivers, each having more than a single antenna.

In this paper, a contactless fall detection system using CSI extracted from *Software Defined Radio* (SDR) devices is proposed. We chose to use CSI in our fall detection system because CSI systems do not suffer from the aforementioned drawbacks. I.e. they are contactless and not costly since they can be set up using a single WiFi router which is commonly available in each household. Moreover, CSI from *Commercial Off-The-Shelf* (COTS) WiFi devices have been successfully used for minute movement recognition, such as gesture recognition [11] and keystroke tracking [12].

The falls, amongst other activities, are detected and classified using a Machine Learning (ML) algorithm, particularly the Random forest classifier. The algorithm was trained on data collected after conducting an experiment using the Universal Software Radio Peripheral (USRP) devices, working as SDRs, as highlighted in Section II. The contributions presented in this paper are as follows:

1) Multiple-activities and fall detection system, with high classification accuracy, using a single antenna transmitter and a single antenna receiver.
2) Individual occupancy monitoring - Identifying whether the subject is performing an activity, falling, or if the room is empty.
3) Capability of the framework to detect the falls in real-time by measuring and analyzing fall detection classification time.

The rest of the paper is structured as follows: Section II presents and discusses the data collection technique, experimental setup including the hardware and software components, and the analysis techniques; Section III presents the results and a discussion of them; Finally, the paper is concluded in section IV.

II. Methodology

The system consists of hardware and software components for data collection, training, and system testing.

A. Data Collection

We chose to create our dataset in this work to have more control over the setup configuration parameters. The data was collected by performing the action (activity) in between transmitting and receiving USRP devices, as depicted in Fig. 1, which also shows the main system components.

The activities chosen for this work are *Falling*, *Walking*, *Sitting*, and *Standing*. These activities were the main focus because they are the most common activities performed by

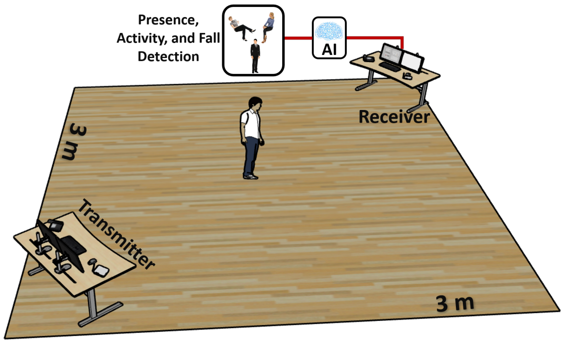

Fig. 1. Hardware data collection setup. The USRP X300 was used as the transmitter (Tx), the X310 was used as the receiver (Rx) and both USRPs utilized the VERT2450 omnidirectional antenna. The USRPs were operating at 5.00 GHz and were connected, individually, to *Personal Computers* (PCs) via a 1*G Small Form-Factor Pluggable* (SPF) connector. The PCs were running Ubuntu 16.04 on an Intel(R) Core (TM) i7-7700 3.60 GHz with 16 GB of RAM.

one person in a closed environment. To create the dataset, the CSI for the aforementioned activities were generated by using the system setup mentioned earlier in a $3\,\mathrm{m} \times 3\,\mathrm{m}$ room at the James Watt South building of the University of Glasgow, where there is an active ethical application. The USRP devices were placed at two opposite corners in the room at $45°$ angle, and the activities were performed between them, as shown in Fig. 1. The created dataset consisted of 5 classes *Falling*, *Walking*, *Sitting*, *Standing*, and *Empty*, where every class had 50 samples with ≈ 1200 feature each. The *Empty* class was the CSI for an empty room.

B. Training and Testing Algorithm

The ML algorithm used in this work is the Random Forest with 10-fold cross-validation and the train-test split methods. It was used because of its superiority over other algorithms, namely K-Nearest Neighbours (KNN) and Support Vector Machine (SVM) in similar cases, as per a previous study conducted by the authors [13]. The Random Forest algorithm establishes and utilizes a collection of decision trees to predict the output based on features learnt during training [14].

C. Evaluating the Framework's Real-time Performance

We have modelled the framework as a single queue with a deterministic inter-arrival rate $\lambda(1/s)$ [15], that depends on the duration of the fall and the USRP sampling rate. The time taken by the system to detect a fall is represented as the queue service time $\mu(1/s)$, which is a random variable that can be assumed to follow an exponential distribution, as will be validated in Section III. The number of servers here is one, since we have one server where the detection is performed. Thus the framework is modelled as a D/M/1 queue.

To measure the framework's performance and responsivity, we have used the queue model to evaluate its delay time. The mean delay time ($\mathbb{E}[T]$), accommodates the queuing time (W) and the service time (S), as follows,

$$\mathbb{E}[T] = \mathbb{E}[S] + \mathbb{E}[W], \tag{1}$$

where $\mathbb{E}[.]$, is the expectation operator. The service time is calculated using $\mathbb{E}[S] = 1/\mu$. Hence, the expected delay is [16],

$$\mathbb{E}[T] = \frac{1}{\mu} + \frac{\beta}{\mu(1-\beta)}, \tag{2}$$

where β for deterministic inter-arrival time can be obtained using Lambert W function (W) [17], as follows:

$$\beta = -\rho W\Big(- e^{\rho^{-1}} \rho^{-1}\Big), \tag{3}$$

where ρ refers to the server utilization, that is,

$$\rho = \frac{\lambda}{\mu}.$$

In the performed experiment, the detection time (service time) was measured for 50 falls, then averaged to find their median, to minimize the effect of any outliers. The inter-arrival time was set to 3 s to ensure that the activity occurs and the change in the CSI is observed. It is worth mentioning that in 3 s, the number of CSI readings is 1200 samples.

III. Results and Discussion

To highlight the contributions outlined earlier in Section I, this section presents two sets of results. The first will evaluate the classification accuracy of the ML model, with emphasis on the fall detection, through a 10-fold cross-validation and the train-test split methods, based on the Random Forest classifier. The second will present a method to compute and analyse the time taken to classify the activity in real-time once the model is built.

A. Fall Detection Accuracy

To evaluate the system's capability to detect falls when they happen, all 250 samples, representing the 5 classes, were used to build one data set. On this data set, and as mentioned earlier, two experiments were performed to test the strength of the ML model. The first was the Random Forest 10-fold cross-validation which showed a high accuracy of 95.6%. The second experiment was a train-test split where 20% of the data were taken as "unseen testing data." The remaining 80% were used to train the ML model. The results of the train-test split method reported 98% accuracy. The confusion matrices for both experiments were generated to look closer at each class and evaluate the system's performance, see Fig. 2.

The confusion matrices presented in Fig. 2 show the results of the 10-fold cross-validation and train-test split experiments. Both matrices show the number of samples correctly classified and the ones that were misclassified. In Fig. 2a, that is, the confusion matrix for the 10-fold cross-validation, the two best performing classes are the *Empty* and the *Falling*, with accuracies of 100% and 98%, respectively. The recorded accuracies show that the system can differentiate, with high accuracy, between an empty room and a subject falling in it. What further strengthens the system and the ML model is the inter-class variation from the presence of three more classes, that is, *Sitting*, *Standing*, and *Walking*. However, the misclassification

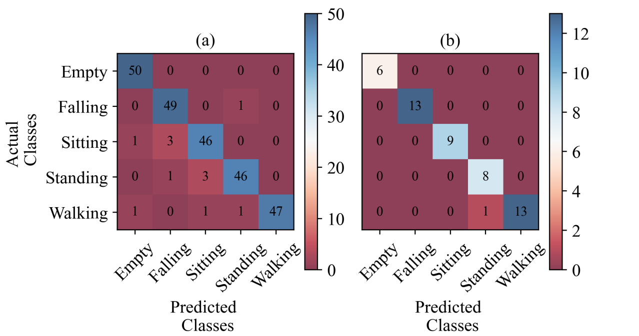

Fig. 2. (a) 10-fold Cross Validation, (b) Train-Test Split. Confusion matrix for all five classes with the Random Forest algorithm.

was slightly higher with those three classes, and the recorded accuracies were 92%, 92%, and 94%, respectively.

In Fig. 2b, the results further support the strength and capability of the system to detect, amongst other activities, if the subject has fallen. As per the presented confusion matrix, the trained ML model managed to classify with an accuracy of 100% the "unseen testing data" from the first four classes and misclassified a single sample from the *Walking* class, as *Standing*.

To shed some light on the previously reported misclassifications, as per the confusion matrices, Fig. 3 shows the CSI captures of the five classes *Falling*, *Walking*, *Sitting*, *Standing*, and *Empty*. Although the variation in patterns is quite clear, showing the distinct features within each class/activity, some of the features, represented by the CSI data, have similar patterns as can be seen in parts of the plot. Given the limited number of samples collected for this experiment, this similarity in the patterns can mean that the ML model will confuse samples that belong to one class, with another. This is potentially the reason for the misclassification presented in the confusion matrix; however, further investigation is needed through more data collection.

B. Real Time Detection Performance

Initially we started by validating the exponential service time assumption. Fig. 5, shows the histogram of the measured detection time, showing that the falls detection time follows an exponential distribution. In particular, the measurements show that even for 50 data points the distribution is observable, thus we can verify the assumption.

Next, we measured the average detection time. Fig. 4 shows the detection time for the 50 iterations. We can observe that the majority of the falls were detected in less than $0.079\,\mathrm{s}$. Nevertheless, an outlier exists in the tested data points. Hence, the median was used to evaluate the expected detection time.

The delay time calculated using Equation (2) gives us a value of $0.073\,64\,\mathrm{s}$. Thus we can conclude that the framework can detect falls in real-time. In particular, the framework can detect in less than 0.1 seconds and with high accuracy.

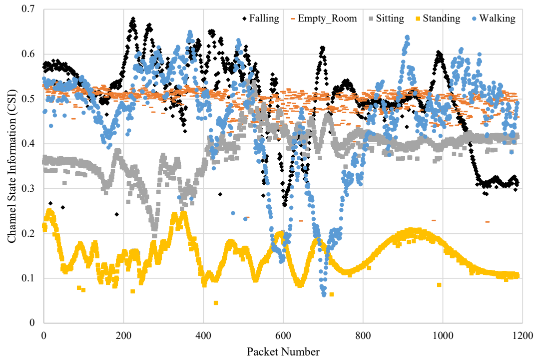

Fig. 3. Channel State Information for the five classes.

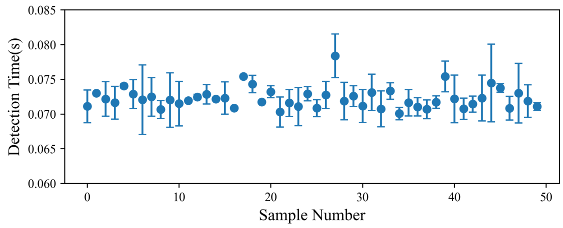

Fig. 4. Measured detection time for all of the 50 *Falling* samples.

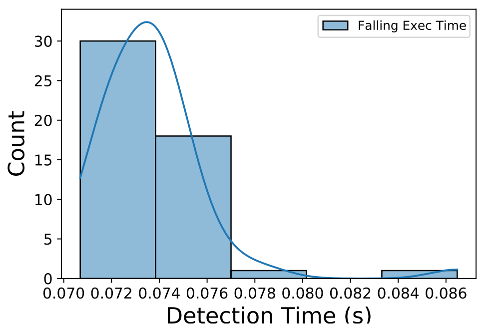

Fig. 5. Histogram of the measured falls detection time.

IV. Conclusions and Future Work

This paper presented an AI-enabled contactless sensing system for presence, activity, and falls detection. The accuracies reported by the Random Forest 10-fold cross-validation 95.6% and the train-test split method 98% are promising and show the potential of the proposed framework to play a vital role in in-home activity monitoring and falls detection systems. Moreover, the real-time analysis of the falling data has shown that the system can also detect falls in real-time. The steps to follow involve recreating the environment from this paper and collecting more data whilst introducing further inter-class variations, adaptation of transfer learning, and implementing the system such that classification is performed in real-time.

Acknowledgment

This work is supported in parts by EPSRC grants no. $EP/T021020/1$ and $EP/T021063/1$.

References

[1] World Health Organisation, "Falls," 2021. Available at: https://www.who.int/news-room/fact-sheets/detail/falls. Accessed: 30.06.2021.

[2] E. Public Health, "Falls: applying All Our Health - GOV.UK," 2018.

[3] N. Hnoohom, A. Jitpattanakul, P. Inluergsri, P. Wongbudsri, and W. Ployput, "Multi-sensor-based fall detection and activity daily living classification by using ensemble learning," in *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*, pp. 111–115, 2018.

[4] B. Jokanović and M. Amin, "Fall detection using deep learning in range-doppler radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 180–189, 2017.

[5] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & posture*, vol. 28, no. 2, pp. 285–291, 2008.

[6] M. Nyan, F. E. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *Journal of biomechanics*, vol. 41, no. 16, pp. 3475–3481, 2008.

[7] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 611–622, may 2011.

[8] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Scientific Reports*, vol. 11, p. 2471, dec 2021.

[9] K. De Miguel, A. Brunete, M. Hernando, and E. Gambao, "Home camera-based fall detection system for the elderly," *Sensors*, vol. 17, no. 12, 2017.

[10] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2017.

[11] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–7, IEEE, 2015.

[12] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st annual international conference on mobile computing and networking*, pp. 90–102, 2015.

[13] W. Taylor, S. A. Shah, K. Dashtipour, A. Zahid, Q. H. Abbasi, and M. A. Imran, "An intelligent non-invasive real-time human activity recognition system for next-generation healthcare," *Sensors*, vol. 20, no. 9, 2020.

[14] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation," *Biomedical Signal Processing and Control*, vol. 52, pp. 456–462, 2019.

[15] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*, vol. 2. Prentice-Hall International New Jersey, 1992.

[16] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *2012 Proceedings IEEE INFOCOM*, pp. 2731–2735, IEEE, mar 2012.

[17] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth, "On the lambertw function," *Advances in Computational mathematics*, vol. 5, no. 1, pp. 329–359, 1996.