IoT based Fall Detection System for Elderly Healthcare

Falls are a leading cause of immobility, morbidity, and mortality in older adults. Falls incur high cost to health services with millions of bed days. Half of the older adults over 65 years old, fall in a span of 5 years with 62% sustaining injuries and 28% protracting extensive injuries. Automatic Fall Detection System (FDS) for elderly healthcare through Internet of Things (IoT) human centered design can provide timely detection and communication of fall events for immediate medical aid in case of injury or unconsciousness. FDS have been reported to provide reduction in death rates of up to 80% due to timely medical support. In this chapter, we discuss elderly-centric IoT-based FDS for smart homes and care centers with emphasis on edge, fog and cloud IoT layers. Sensing edge devices with wearable/environmental sensors, vision-based systems, and radio frequency sensing systems, such as WiFi-based sensing and RADAR are presented for an IoT centered FDS. IoT gateways and communication protocols for the fog layer are discussed in the context of an FDS. Cloud processing of edge device data for fall activity detection and classification from activities of daily life is explained. Machine and deep learning algorithms for detection of fall events from 1D and 2D signals (image/video) are presented and various deployment scenarios are discussed in the context of edge or cloud IoT layers. This chapter is concluded with results and performance comparison of several IoT centered FDS in terms of various sensing systems and state-of-the-art machine and deep learning models for effective detection of falls for elderly healthcare. Furthermore, future work and prospective improvements in IoT centered design for fall detection in elderly healthcare is discussed.

Keywords: Internet-of-Things, Fall detection system, wearable systems, WiFi/Radar sensing, machine learning, deep learning.

1 Introduction

A fall is defined as an inadvertent descent to the ground or floor. Falls may result in fatal or non-fatal injuries. Falls in elderly may result in fractures and are costly in terms of health services. They are usually associated with multi-factorial causes including age, patient history, muscle weakness, visual impairment, poor balance and environmental causes. Falls are the second highest cause of unintentional deaths worldwide after road accidents [1]. Worldwide 37.3 million falls require medical attention with 0.68 million deaths. Adults over 60 years old suffer the highest number of deaths due to falls worldwide [1]. In UK alone, half of the older adults over 65 years old fall in a span of 5 years with 62% sustaining injuries. Amongst those who suffer from falls, 28% sustain extensive injuries, 21% lose confidence and 10% lose independence [2]. Falls cost 4 million bed days to UK National Health Service, along with 4 billion pounds in health-related costs [3]. The cost of falls in terms of high morbidity and immobility has resulted in a focus on fall detection systems for older adults. The system can detect occurrence of fall events and are imperative for older adults who live alone and may not be able to call for help due to unconsciousness or injury. Fall detection systems can provide timely intimation of fall events resulting in immediate medical help and are known to improve hospitalization by 26% and death rates reduction by 80% [4].

Internet-of-Things (IoT) consists of a large number of smart physical devices connected to the internet through gateways without human-computer interaction and can communicate data in real-time. IoT based fall detection system can transfer real-time fall events to the cloud with smart sensing devices acting as "things".

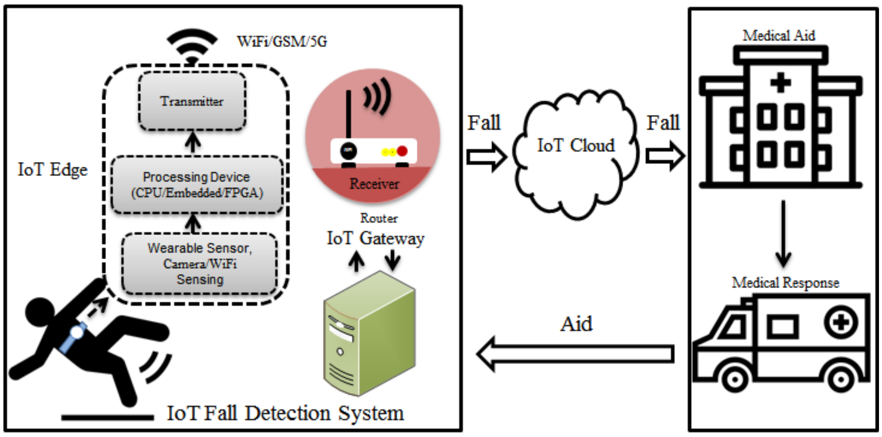

Fig. 1. Fall Detection System Overview

Figure 1 illustrates an IoT based fall detection system with IoT edge, gateway and cloud. The system can utilize various communication technologies for access to the internet, such as WiFi, GSM and 5G. The sensing devices in a fall detection system can be categorized into three broad categories 1) wearable sensors, 2) vision-based sensors and 3) WiFi/Radar sensing devices. The devices gather human activity data, such as acceleration values from human body movements. The data can be processed at the transmitting embedded device for classification and detection of falls or at the receiver side on the IoT Gateway by the corresponding processing device. The processed data is transmitted to the IoT cloud from where it can be accessed through smart-phones or desktop computers. The fall events can be communicated through cloud to a server in a medical emergency center in real-time, which is essential for dispatching timely medical aid. The level of processing done at the IoT edge device or gateway and the data to be sent to the server are flexible design decisions. Security and privacy issues that may not arise in traditional offline fall detection systems are important aspects for an IoT based fall detection system. Encryption and decryption process are part of the real-time processing tasks for IoT based sensing for fall detection.

Figure 2 illustrates the concept with three sensing devices, a wearable accelerometer, a camera and a WiFi sensing device. The accelerometer gives acceleration values of body movements in units of “g” ($9.8 m/sec^2$). The video frames and WiFi Channel State Information (CSI) are obtained from the camera and the WiFi sensing device. WiFi CSI values are variations in wireless channel estimation that vary over time due to changes in wireless channel caused by objects and human body movements in the vicinity. The signals from various sensing devices are processed with signal processing algorithms, such as time-frequency spectrograms for 1D signals or foreground-background segmentation for 2D/3D images or video frames to obtain moving objects. Machine learning classification algorithms are applied to the processed data and features to detect falls at the IoT edge device.

Deep learning algorithms are also utilized for classification in fall detection systems [5, 6]. However, deep learning techniques are computationally intensive and may be deployed at the IoT gateway or cloud due to high processing requirements. In this scenario, the signals from sensing devices can be processed at the edge, Edge processing usually involves removal of outliers in sensor readings and use of signal processing techniques, such as application of digital filters for de-noising or to obtain certain frequency ranges. The processing at edge device may also include compression along with encryption before transmission of signals. The transmitted signal can then be further processed at the IoT gateway. Deep learning techniques require higher throughput for training and can be applied at the processing device connected to the gateway, such as a desktop computer. Furthermore, the signal can be transmitted to the cloud where it can be visualized in real-time. The cloud offers more flexibility in computing resources and a large number of signals can be processed at the cloud with high throughput graphics processing units. More complex deep learning models can be utilized at the cloud for higher classification accuracy of fall events. Figure 2 illustrates both the scenarios with processing and classification performed at the edge device or at the gateway/cloud.

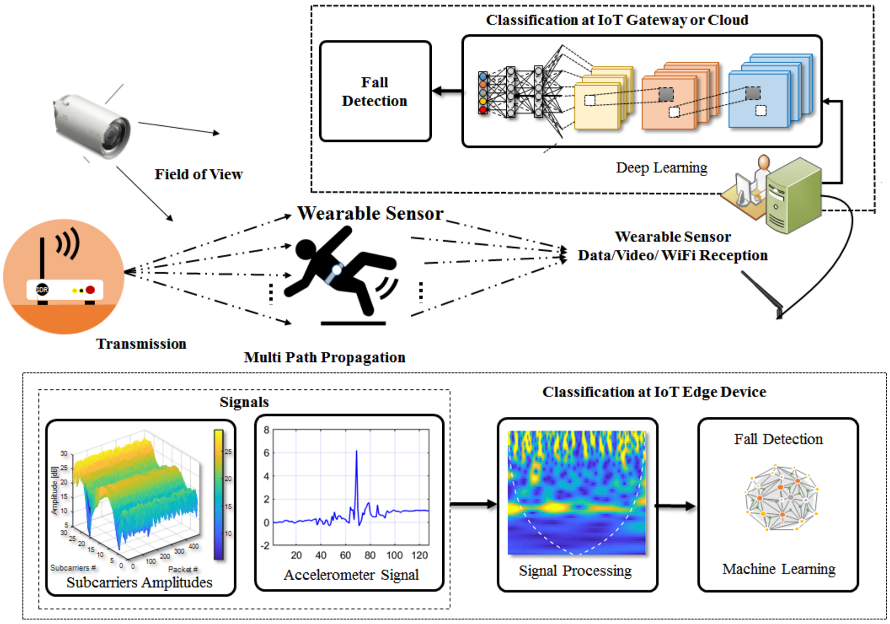

Fig. 2. IoT based Fall Detection System Processing and Classification.

2 Edge IoT Layer

Edge IoT layer consists of edge devices which make up the sensing system end of the fall detection system and consists of sensors and embedded processors.

2.1 Edge Devices

The edge devices are mostly made up of various embedded boards for processing and consist of embedded cores and Field Programmable Gate Array (FPGA) based reconfigurable System on Chip (SoC) devices. The smartphones are also part of the edge IoT layer. The IoT based fall detection systems have their own unique issues and demand stringent requirements compared to non-IOT based fall detection systems, such as privacy and security issues that may arise due to uniquely addressable devices and their global internet connectivity.
The edge IoT layer embedded edge platforms should have most if not all the following characteristics with stringent requirements for security and privacy:

1) **Uniquely Identifiable**: The edge devices used for fall detection should be uniquely identifiable. An IP address provides globally unique address to identify a particular user with the wearable sensor or an environment where the sensor is installed.
2) **Smart platform**: The device should provide a smart platform with sensor or wireless sensor connectivity with processing capabilities and an embedded core for running various tasks and algorithms.
3) **Embedded Machine Learning (ML) cores**: Now a days ML cores for learning and classification tasks are becoming a norm on edge devices. The custom application specific integrated circuits are an essential requirement for low power and real-time edge Artificial Intelligence (AI).
4) **Real-time processing**: Unlike the edge devices for measuring temperature and other environmental factors that can do with lower sampling rates of a sample per minute, the device for fall detection systems provide higher sampling rate in real-time to process or transmit data from sensors. The accelerometers can use a typical sampling rate of 50 samples per second, camera based systems should be able to provide higher rates from 30 to 60 frames per second and a wide range of resolution. Typical WiFi sensing systems have sampling rates from 50 samples per second up to 400 samples per second for 5G software defined radios used for healthcare activity classification.
5) **Energy efficiency**: The devices should provide higher energy efficiency. However, the requirements vary from system to system and depend upon the type of sensors. Wearable sensors with Inertial Measurement Units (IMU) should typically work in a current range of micro Amperes.

6) **Power management**: Power management is an important aspect and the devices should have support for sleep mode to save power, e.g, sensor readings may not be required, when the person with a wearable sensor is sleeping or resting.
7) **Privacy and security**: Privacy and security are important aspects of sensing devices for IoT based fall detection systems. The unique IP addresses allow the sensors to be accessed globally and should provide encryption of data along with authentication mechanisms for the edge devices utilized for fall detection system. The vision based sensing platforms in this regard create a higher security risk and should have stringent access authentication to avoid hacking.

A large number of embedded platforms can be utilized for IoT based fall detection system. Embedded platforms such as Raspberry Pi Pico [7], Arduino [8] and NodeMCU [9] are ubiquitous in edge devices with IMU units. Low power and small form factor IoT capable boards are a good choice for fall detection system. Embedded platforms, such as Adafruit FLORA [10] with 1.75 inch diameter are easily wearable and integrate Arduino compatible microcontroller with accelerometers and gyroscope sensors through Inter-Integrated Circuit (I^2C) bus. iNEMO [11] embedded platforms by STMicroelectronics integrate accelerometer, gyroscope and magnetometer with an embedded machine learning core in a small form factor. Similarly embedded vision platforms, such as iENSO vision board [12] combines edge based AI and image processing capabilities in a small form factor of 2.3 x 2.2 cm.

Reconfigurable SoC embedded platforms are a method of choice for edge IoT devices and have been utilized for fall detection system [13]. They provide a good trade-off between higher flexibility and low power for implementing signal processing techniques, feature extraction or machine learning classifiers on programmable logic. The signal processing can be implemented both in programmable logic and on hardware ARM cores on the SoC device depending upon the computational requirements. Xilinx Zynq SoC has been utilized for fall detection. The SoC provides to ARM cores and programmable logic for implementing computationally intensive algorithms for fall detection. Figure 3 illustrates the Zynq SoC [14] architecture with ARM cores, programmable logic and Xilinx proprietary AXI bus for connecting the accelerator with the ARM cores. Fall detection sensors, such as an IMU with accelerometer and gyroscope sensors can be connected to the ARM core through the Serial Peripheral Interface or I^2C bus.

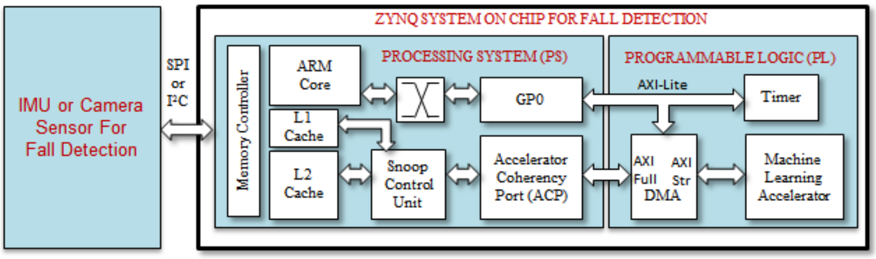

Fig. 3. Reconfigurable embedded platform for a fall detection system

An example fall detection system presented by Tahir et al. [13] based on the Zynq platform is illustrated in Figure 4. The system implements feature extraction in programmable logic and machine learning classification for fall detection in software on the ARM core. The sensor transmits accelerometer data from sensor to a Zynq edge device wirelessly. The signal is processed to compute mean and the zero meaned signal is wavelet transformed to obtain fractal features for fall detection on programmable logic. The features are passed to the ARM core for machine learning classification in software. The feature extraction process is implemented on programmable logic since it is more computationally intensive than the Linear Discriminant Analysis (LDA) algorithm. LDA is computationally less intensive than the feature extraction process [13]. Furthermore, apart from reconfigurable platforms, smart phone are ubiquitous edge devices with inbuilt sensors, such as accelerometer and gyroscope that can be utilized for fall detection. Smart phones can also be used with smart watches. Apple wearable watch can detect falls and send messages directly through the Apple smart phone to emergency contacts. The smart phone-based wearable's get internet connectivity for fall event occurrence through the smart phone directly connected through GSM or WiFi router [15].

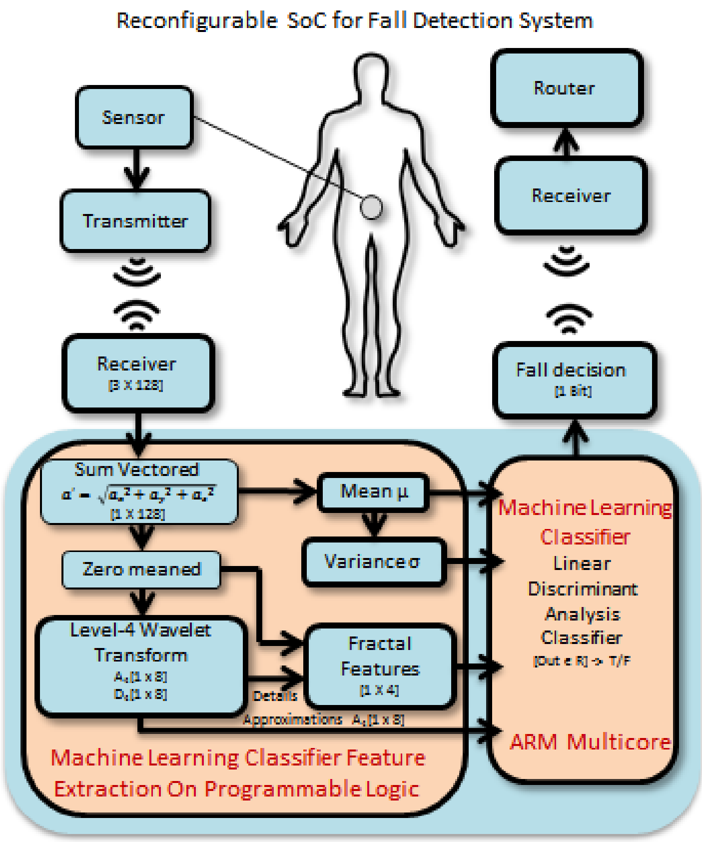

Fig. 4. An example reconfigurable fall detection system on reconfigurable SoC device.

2.2 Wearable/Environmental Sensing System

Accelerometers and Gyroscopes Accelerometers can be used to track human motion via wearable devices or embedded in smart phones that users carry [16]. This can be advantageous if the 1 user already owns a capable mobile device but can be expensive when providing mobile devices. The wearable devices can be in the form of smart watches with embedded accelerometers. Accelerometers work together with gyroscopes to be able to determine the orientation of the user's body [17]. The data received from the accelerometers can be applied to machine learning algorithms to classify the human motion taking place [18]. Pressure sensors electrode arrays woven into fabric which can be worn by the user and allows for the detection of muscle movement and thus the detection of movement [19]. Pressure sensors can also be applied to furniture material to detect sit-to-stand and stand-to sit motions [20]. Acoustic sensors are electrical devices that has the ability to measure sound waves in the environment. The sound waves can be used for fall detection systems by analyzing the acoustic signals [20]. The signals received from these sensors are then processed to remove noise from the signals. These noise removing filters can include high and/or low Butterworth filters [21]. These processed signals can then be used in AI to train models which can recognize the acoustic waves accompanying a fall [22]. Figure 5 shows the process followed for wearable fall detection systems.

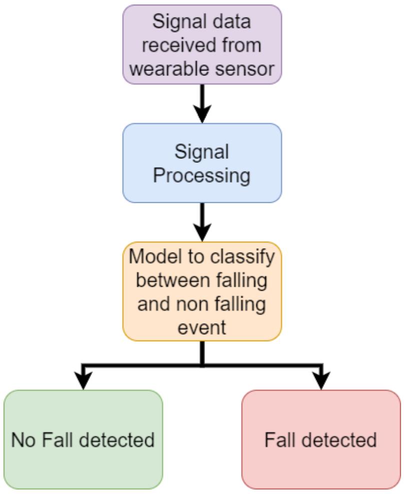

Fig. 5. Fall detection system using wearable sensors

2.3 Vision-based sensing Systems

2.3.1 Camera and Kinect depth sensors

Camera technology can be used to record individuals. This method will allow the user to not have to wear any devices while being monitored. Machine learning can be used to remove the need for a human to be observing the video footage which can prevent the subject from feeling that they are being watched. This type of system can achieve high accuracy [23]. However, camera systems 2 can sometimes be expensive. To decrease the costs of the systems, devices such as a low-cost Raspberry Pi device with a camera can be used to obtain good performance compared to more expensive devices [24, 25]. Machine learning can be applied to the frames of the footage to establish if an individual has fallen [26]. Depth sensors work by using two sensors with a known range between them to calculate the depth [27]. As the depth changes, movement can be inferred. Kinect sensors are well known devices for Microsoft's Xbox gaming console. The device allowed for gaming where peoples movements would be sensed as input. An example of this is for games where the users are required to perform dancing [28]. These sensors can be applied with deep learning to be able to classify poses of humans within healthcare applications [29].

2.4 Contactless Non-Interference Sensing

Radio Frequency (RF) and Radar are contactless method that allows for detection of human movements without the need for the user to wear a device or have vision-based sensors raising privacy concerns. This removes the problem of users having to remember to wear devices and avoids any discomfort of either wearing devices or the intrusiveness of vision-based systems in the home. These methods are known as contactless non-intrusive methods of detection.

2.4.1 Radar Sensing

Radar technology provides sensing of the environment which can be used to monitor daily routine activities of elderly people [30]. Radar based sensing works by exploiting the Doppler signatures created on radar when movement occurs [31]. This can be used in healthcare applications for example if a elderly person experiences a fall. The Doppler signatures can be presented using spectrograms. Spectrograms are visual representation of the spectrum of frequencies of a signal. An example of a spectrogram containing a fall activity is shown in Figure 6.

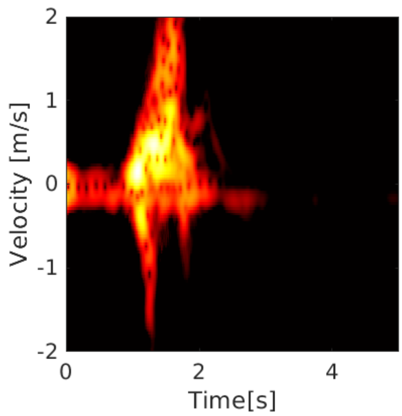

Fig 6. A Spectrogram displaying a falling action.

These spectrogram images can be used to in AI for image classification of what activity takes place in that particular spectrogram [32, 33, 34, 35]. Figure 7 displays the process used in a radar fall detection system.

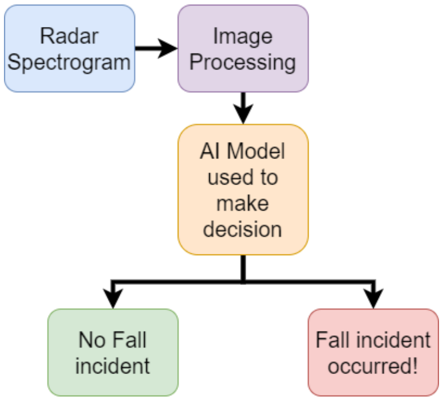

Fig 7. Radar fall detection system processing

2.4.2 Radio Frequency Sensing

Radio Frequency Sensing works by observing the state of a wireless communication link between devices such as the case with a Wi-Fi network within the home. As the signals travel through the atmosphere, they will propagate differently depending on objects in the room. These objects can include humans and the signals will propagate differently depending on the positioning of the body. Wi-Fi records the information of the signal propagation, and this is called the CSI. The CSI is used from Wi-Fi to look at the amplitude of the RF signals while the human moves between the RF signal [36, 37]. Figures 8 and 9 shows

CSI amplitude samples of a falling motion and a non-falling action. The CSI describes how the wireless signal propagates between the transmitting node and receiving node [38]. This data can be exploited to detected changes during a specific human motion such as the example of a fall occurring.

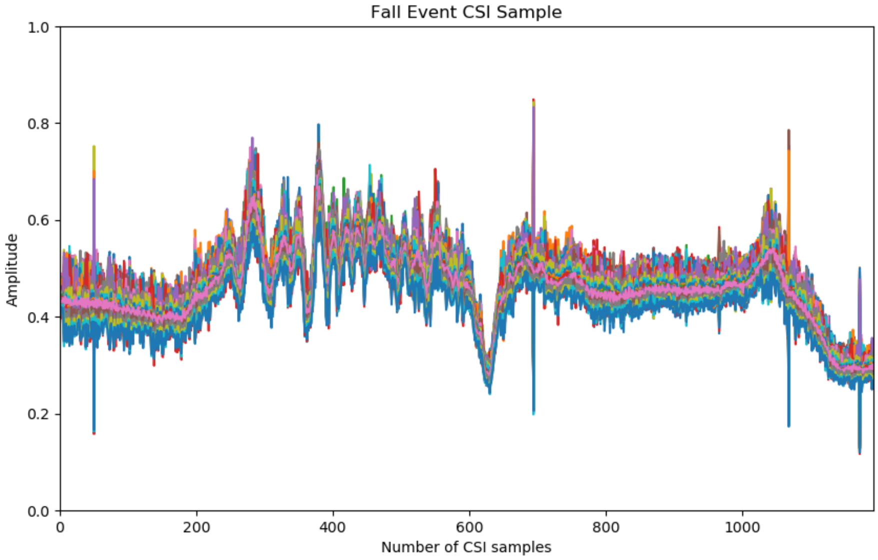

Fig. 8. Sample of a fall event captured using CSI amplitude.

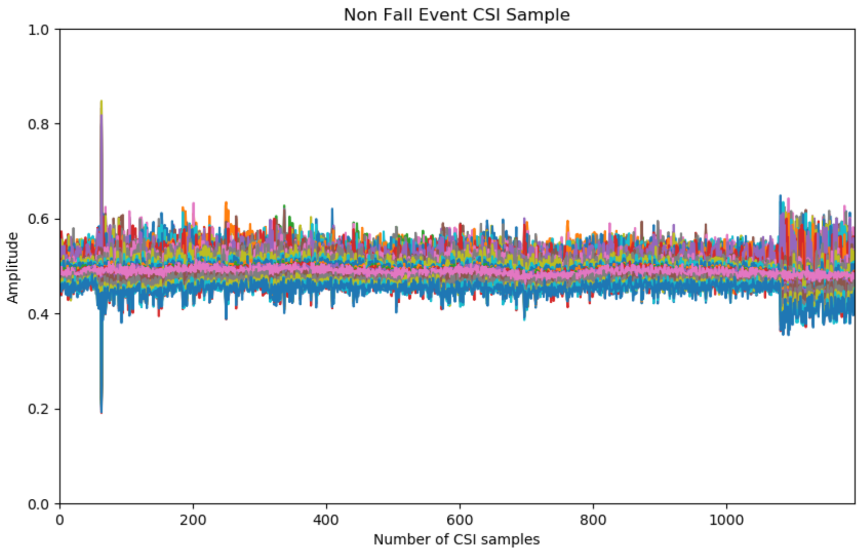

Fig. 9. Sample of a no fall event captured using CSI amplitude.

WiFi is considered superior due to its low cost due to the extensive coverage in homes already [39]. Another advantage of using Wi-Fi is that it eliminates the need for excessive equipment which can feel invasive, and the additional equipment can be expensive and require maintenance [38]. By implementing systems to only use the smallest amount of equipment costs are keep down as well as ease of installing. A system using RF signals will monitor the CSI of incoming signals. Signal processing is applied for noise reduction and to make changes within the signal more prominent. Then the signals are passed 5 through a model which can observe the patterns of the CSI and is trained to recognize a fall signature in the CSI. Then a trigger can be sent to indicate a fall has been detected in the system. Figure 10 shows the process used in a RF fall detection system.

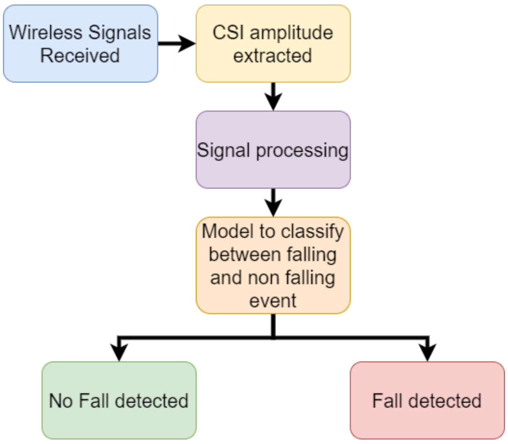

Fig. 10. RF fall detection system processing

3 Fog IoT Layer

In FDS, Fog IoT layer consists of various different devices (mainly IoT gateways) and communication technologies that connect those devices with the computing platforms. Particularly, the communication technologies in this context include, but are not limited to, cellular networks (such as 5G, 4G, and GSM), Zigbee, Bluetooth, NFC, WiFi, LoraWAN, and so on. IoT gateways have emerged as a key component of a robust IoT platform that can help enable an effective FDS. The gateway acts as a communication and computing hub wherein different sensors are connected to it via one of the communication technologies [40]. Furthermore, gateways connect those sensors to different users, applications, and the Internet. In most cases, IoT gateways act as a bridge between sensors/actuators and the Internet. The communication architecture of IoT is given in Figure 11.

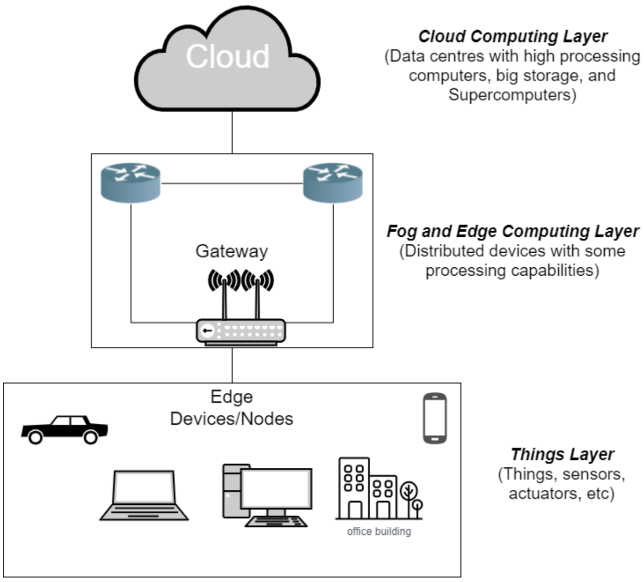

Fig. 11: The IoT Communication Architecture

In terms of features set, gateways generally host one or more of the following features:

1) Gateways facilitate communication to/from sensors and Internet/non-Internet devices.
2) Data caching, streaming, data aggregation, pre-processing, cleansing, filtering and optimization.
3) System diagnostic and configuration management.
4) Security - device and network security.

3.1 Communication Technologies

The communication architecture of IoT gateway depends on the underlying technology being used. In general, the communication technologies are divided 8 into four different categories, (i) peer-to-peer (P2P) technologies, (ii) Low power and short range mesh technologies, (iii) local area network (LAN) technologies, and (iv) low power and long distance technologies [41].

3.1.1 Peer-to-peer (P2P) Technologies

P2P means that only two devices are connected together directly. In the context of IoT networks, a sensor node can be connected to the gateway in P2P fashion. P2P technologies include legacy Bluetooth, WiFi Direct, and near field communication (NFC). Bluetooth is the best known P2P technology, which is present in almost every smartphone and tablet. Due to low range communication, it is much more power efficient than WiFi and cellular technologies. WiFi Direct is an alternative to Bluetooth, which is native to almost every modern-day smartphone. WiFi Direct is P2P technology, which works without the need of an access point (AP). The working principle is similar to infrastructure mode WiFi wherein one of the participating devices takes roles of an AP while others act as clients. Generally, WiFi Direct is faster than Bluetooth. Near field communication is another P2P technology which uses electromagnetic field between coils to enable communication between two nodes. NFC works on the principle of electromagnetic coupling, the communication range is generally within an inch or two. This makes NFC a very secure technology with a very little chance of eavesdropping.

3.1.2. Low Power and Short-Range Mesh Technologies

There are two main technologies that create a low power and short-range mesh network, which are Bluetooth low energy (BLE) and Zigbee. These technologies are very important when someone is dealing with an application which has battery-powered devices, sending a low amount of data to a shorter range. BLE is a highly power-efficient communication protocol, which works with different devices, transmitting data at different rates. BLE is highly scalable supporting up to 32, 767 devices connected in a mesh network. BLE is among the most adopted technologies in IoT networks, especially in the Internet of medical things settings. BLE uses the 2.4 GHz band. Zigbee is a competitor of BLE that uses the same 2.4 GHz band, operates in a mesh network topology and has the same range as of BLE. In terms of scalability, Zigbee supports twice as many devices as supported by BLE, that is, around 65, 000 devices. Home automation and industrial automation are a few applications of Zigbee.

3.1.3. Local Area Network (LAN) Technologies

WiFi is a good option in the scenarios wherein sensors support wireless LAN and need direct access to the Internet. The coverage area provided by the WiFi is better than Zigbee and BLE. In addition, WiFi is a readily available technology and its coverage is ubiquitous. Another advantage of WiFi is the supported data rate, which is far better than Zigbee and BLE. However, these advantages come at the expense of power consumption. The power consumption of WiFi is a way more than Zigbee and BLE.

3.1.4. Long-Distance Low-Power Technologies

There can be scenarios where IoT devices are deployed at remote locations and they send a low amount of data to the Internet. For instance, weather monitoring sensors mostly have very low data to send. The cellular technologies, such as GSM and LTE are not suitable in this scenario as they generally do not support very low data rates. These kinds of scenarios are generally known as low-power wide area networks (LPWAN). The most popular technologies in LPWAN are LoRa/LoRaWAN, narrowband IoT (NB-IoT), and LTE-M. LoRa is a long-range P2P technology having a range of more than 6 miles in some areas. The frequency range of LoRa varies in different regions. For instance, in North America, the operating frequency of LoRa is 915MHz while in Europe it is around 868MHz. In some areas it is also licensed at 169MHz and 433MHz. LoRa is the underlying technology while LoRaWAN is a network layer protocol. The only way to provide Internet access to LoRa devices is through a LoRa gateway. On the other hand, NB-IoT is a cellular based technology, which provides coverage to remote sensors. Being a cellular technology, it is complex and more expensive in terms of budget and power consumption. However, it provides a direct access to the Internet. NB-IoT supports low data rates but it is not yet deployed in many areas of the world. LTE-M is best suitable for long distance and high data rate scenarios. LTEM provides sensors a direct access to the Internet using 4G cellular network. LTE-M is fundamentally different from standard LTE technology as it is optimized for the low power consumption suitable for battery powered devices.

The most common data communication protocols are Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP), Data-Distribution Service for Real-Time Systems (DDS), and Hypertext Transfer Protocol (HTTP). The details of these protocols can be found at [42].

4. Cloud IoT Layer

The cloud layer in an IoT network supports caching and processing capabilities that can be accessed by different IoT devices and applications. The resources at the cloud (caching, processing, etc) can be accessed by different IoT applications anytime and anywhere. To this aim, different API are used which are generally made available to any HTTP client. For instance, Google cloud IoT Core API mainly comprises of two sets of REST resources: cloud IoT and cloud IoT device. In order to get the information about different subscribers virtualization technology is used, which is also helpful in getting the segregation of IoT applications. With the help of APIs and virtualization IoT applications can provide different quality of service (QoS) to different users over the same physical network.

5. Machine and Deep Learning Algorithms

Random forest algorithm works by using a collection of decision trees. These trees make predictions based on features found in the training data. Each trees prediction is considered a vote. The majority of predictions decide the final

Random Forest prediction [43]. The K-Nearest Neighbors algorithm is well known for its simplicity. KNN makes direct comparisons between the testing data and training data [44]. The features of the training data are assigned a K sample then the testing data is assigned to the K sample with the nearest match [45]. During the training phase Support Vector Machine attempts to create boundaries known as hyper planes between classes. The hyper plane is positioned as far as possible from the closest data points of the classes present in the data. These points are known as the support vectors [46]. The hyper planes are used to divide the support vectors into the different categories. The features of new data are used to place the new data between the hyper planes and provide classification [47].

The Long short-term memory (LSTM) deep learning algorithm is an extension of a recurring neural network (RNN). A recurring neural network is a type of neural network which models the dynamic behavior of sequences of data between nodes of the neural network. LSTM expands on RNN with the use of three different gates on each node. the first gate decides if the current state should be erased. the second gate is used to control if input should be considered, and the final gate decides if the state should be included in the node output. These gates allow for LSTM to decide if the sequence of data is relevant to the output of the node [48]. Bi-directional long short-term memory (BiLSTM) is a further extension LSTM. Where LSTM only considers past behaviors of data sequences, the BiLSTM considers data in both previous and upcoming data in the sequence. This is possible with the use of two LSTM networks. One LSTM network, the forward LSTM network, can review past data sequences and the backward LSTM network can review future data sequences [49]. The CNN algorithm is an emerging technology which is a powerful solution for image classification problems which were initially thought to require human intelligence [51, 52]. The CNN algorithm is made up of densely connected layers that take the activations of all the previous layers as input. The layers produce feature maps from this input which are known as growth rates [53]. CNN algorithms come in the form of 1 Dimension (1D), 2 Dimensions (2D) and 3 Dimensions (3D) with 3D resulting in highest computational power requirements [38]. 1D, 2D and 3D CNN refers to the number of directions the kernel moves in. 1D CNN makes use of 2 dimensional inputs and outputs for example time-series data. 2D CNN uses inputs and outputs of 3 dimensions and is mostly used for image data. 3D CNN is 4 dimensional for input and output and is mostly used on 3d image data such as MRI and CT scans.

6. Performance Metrics and Results

The severe consequences of falls in the older population have called for the innovative use of technology to develop systems that are capable of detecting and reporting the fall events, if and when they happen. However, without the right metrics to evaluate such systems, the wider community and healthcare systems across the world, will not be able to confidently trust the technology and rely on it.

This section therefore identifies and presents evaluation metrics for different FDSs.

6.1. Evaluation Metrics

To evaluate the performance of an FDS, it is important to consider the main building blocks of the system, from the technology used, to the hardware and software, to installation, and others. Accordingly, the following metrics have been identified to evaluate any FDS, for research or commercialization purposes:

1) System accuracy in real-time detection - Essential to all FDSs
2) Alert generation feature - Essential to all FDSs
3) System portability and ease of deployment - Optional to some FDSs

6.1.1. System Accuracy Metrics

An important aspect of FDSs is the real-time feature, meaning the system's ability to accurately report the activities of the monitored person, as they happen, to a user friendly web-interface or dashboard, in a near-instant timing. From this, three Key Performance Indicators (KPIs) need to be considered, to ensure a perfect score in this metric:

1) Event detection accuracy
2) Event logging in real-time
3) User friendliness of the web-interface

The accuracy of detection has two folds. The first is the method implemented to capture the event, whether using contactless technology, wearable sensors, and/or vision-based sensing devices. The second is the accuracy of the AI-based algorithm implemented to intelligently classify or infer the event being an normal activity or a fall. This therefore needs to be considered in the system design phase where large datasets with inter and intra-class variations collected, and extensive testing scenarios are considered. A typical framework to evaluate the accuracy FDSs can be found in [54] where four binary classifications are considered: • True Positive (TP) - A fall event was correctly detected • True Negative (TN) - A non-fall event was correctly detected • False Positive (FP) - A non-fall event incorrectly detected as fall • False Negative (FN) - A fall event incorrectly detected as non-fall The four binary classifications, TP, TN, FP, and FN, are used to generate four algorithm performance metrics, that is, Accuracy, Precision, Recall, and F1-Score. The Accuracy displays the total number of correct classifications versus the total classifications made (see Equation 1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

The Precision metric is used to measure one of the classifications against how precise it is in comparison to all classifications.

$$Prescision = \frac{TP}{TP+FP} \quad (2)$$

The Recall is used to show the ratio of the correct classification to all classifications for a particular class. This is usually run for all classes in the model and presented as an average.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

The F1-score is used to provide an average between the Precision and Recall metrics.

$$F1 - Score = 2 \times \frac{Precision \times Recal}{Precision+Recall} \quad (4)$$

A typical representation of the four values TP, TN, FP, and FN, is in the form of a confusion matrix, see Figure 12. A confusion matrix is one that is usually used to represent the performance of a classification model to tell how much of the test data has been correctly predicted. In other words, how many data samples have been confused to be a different classification as compared to the true class?

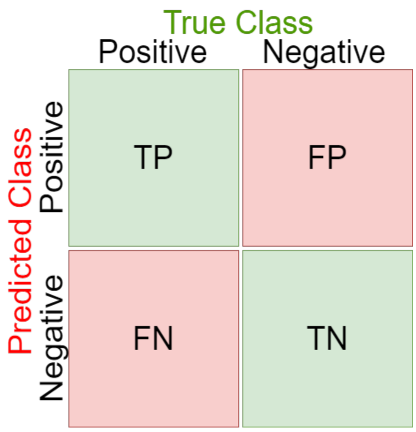

Fig. 12: Confusion matrix showing the true and false classifications.

Although all four values are important and needs to be scored and calculated with as much data as possible, the most serious one is the FN as it is the one that could mean patients or individuals can be in a serious condition, without the system reporting the status. The FP value, whilst it doesn’t have a life-threatening impact, can result in resource waste and damage to the environment if the system was linked to emergency response and constant false alarms were generated. Event logging in real-time: It is crucial to ensure the system reports the event, regardless of what it is, in a timely manner. The severity of delayed reporting is usually associated with the "fall" only, however it is important to ensure a fully functional system. The time between incident and notification/reporting can differ and therefore it needs to be measured and evaluated during the early testing stages of such systems. This can be performed by comparing the reported times to the actual event. Nevertheless, the challenge in conducting this test is coming up with a non-invasive/intrusive method to record the actual event performed and exact timing. User friendliness of the web-interface: This metric is rather a qualitative one, yet crucial to the success of such systems. To the end user, especially care takers or emergency services, the interface is "the system" as they will interact with it and not the sensing technology, majority of the time. Accordingly, design ideas could be shared with the target users, in focus groups, prior to implementing them or ideas collected through questionnaires and/or interviews. To develop a dashboard, the following can be considered:

1) Dashboard accessibility
2) Dashboard design - Color coded events reporting, animation for alerts etc.
3) Security of the reported data and the personal information

6.1.2 Alert Generation

A crucial design objective for remote healthcare monitoring in general and for FDSs in particular, is the generation of accurate alerts. The alert generation feature can be useful to indicate potential threats to the patients and/or monitored individuals, based on the recorded activity levels. This metric can also be used to reflect the accuracy of the system, as it is crucial to ensure alerts are generated only when and if necessary, as well as its real-time feature, previously discussed.

The alert generation metric therefore encompasses two things, the alert type and alert generation time. The type of alert means, what would the system output to inform the target beneficiaries of the reported event. Whilst it might seem trivial, it is crucial to design the system such that the alert is as informative as it can be whilst ensuring it is concise to enable the notified personnel to act upon it in a timely manner. Alerts can be generated based on recorded activities and can therefore be scored to reflect falls, sleep-time disturbance, sleeping time, room transition, wandering, and general activity. Each category has a different severity level and therefore the corresponding alert/notification should be different. Secondly, comes the alert generation time, which is closely tied to the real-time feature of the system, previously explained, to ensure the serious events are reported as soon as they happen, as it may mean a person's life is saved.

6.1.3. System Portability and Ease of Deployment

This metric will not apply to all FDSs and the use case would decide on its application. The purpose of the metric is to evaluate the portability of the system, in cases where it won't be fixed in one place. Depending on the use, such systems can be used for temporary monitoring of patients that are recovering from accidents or surgeries, and so on. Thereby, it is important to have, in the market, systems that can be installed and cleared out in a timely manner. This evaluation metric will therefore be based on the number of hardware nodes/units associated with the implementation of the system and would involve measuring the following:

1) Portability - Is the system mobile or fixed
2) Setup time - Is it a "Plug & Play" system or requires pre-planned setup
3) Maintenance - What level of intervention is needed to maintain the system? Can it be done remotely?

A score can be therefore given for every FDS, based on the above mentioned Metrics, to evaluate its portability and ease of deployment.

6.1.4. Comparison of Results and Systems

Table 1 illustrates some of the FDSs presented in literature and gives a comparison of four systems, each uses a different technology to implement their FDS. The able is used to highlight the main aspects of any FDS, i.e., the technology used, the applied algorithm (Where applicable), and the performance of the system in terms of its accuracy.

Table 1. Comparison between fall detection systems that utilize various technologies

Source	Technology	Algorithm	System Accuracy
Aziz et al. [55]	Accelerometer	Support Vector Machine (SVM)	80%
Bloch et al. [56]	Accelerometer and an infrared sensor	NA	Sensitivity - 62.5% Specificity - 99.5%
Debard et al. [57]	Camera	SVM	Sensitivity: 88% Specificity: 95.6%
Wang et al. [58]	Radio Frequency	SVM and Random Forest	SVM: 90% Random Forest: 90%

As can be seen in Table 1, the performance of every system is reported and from the figures, one can pick a favorite. However, as a whole system, the previously highlighted metrics in Section 6.1 need to be considered to arrive at an accurate evaluation of every system.
Nevertheless, the highlighted metrics can, to some extent, be quantified or scored for every system; the question remains. There are several other factors that

research studies have highlighted and are used to favor one system over the other. For instance, some of the drawbacks of using camera-based systems include invasion of privacy, the need for ambient light, and line-of-sight requirement for detection. As for the wearable technology, concerns are usually around the inconvenience of having to mount the device on the body, whether through direct or indirect skin contact, and the severe consequences that can result from forgetting to wear them. The contactless sensing technology has the advantage over the vision-based and wearable technologies as they do not pose any privacy concerns and are completely non-invasive, however the accuracy is in occasions questioned, compared to sensor-based systems. Therefore, it is crucial to consider several factors when designing an FDS for research.
xxx

7. Conclusions

This chapter explained IoT based FDS for older adults. FDS systems are essential tools for elderly healthcare and imperative for their health and wellbeing. The IoT edge layer with different sensing systems and embedded platforms was described. Essential elements of an IoT FDS system were discussed. Different FDS systems including wearable sensor, vision based and radio frequency, such as WiFi and radar were presented, and their processing steps were discussed. An overview of FDS systems in terms of their focus on machine and deep learning techniques was given. The Fog and Cloud IoT layers were discussed and important communication technologies for internet connectivity were presented. Finally, their performance metrics and evaluation criteria were explained.

References:

[1] World Health Organization: Falls, (2021). https://www.who.int/news-room/fact-sheets/detail/falls (Accessed: 1st July, 2021).
[2] PCP market research: Falls, Measuring the impact on older people, (2012).
[3] Y. Tian, J. Thompson, D. Buck, and L. Sonola, Exploring the system-wide costs of falls in older people in Torbay. King's Fund (2013).
[4] N. Noury, P. Rumeau, A. K. Bourke, G. ÓLaighin, and J. E. Lundy: A proposal for the classification and evaluation of fall detectors. Innovation and Research in BioMedical Engineering, vol. 29, no. 6, pp. 340–349, (2008).
[5] Tahir, A., Ahmad, J., Morison, G., Larijani, H., Gibson, R., and Skelton, D.: HRNN4f: Hybrid deep random neural network for multi-channel fall activity detection. Probability in the Engineering and Informational Sciences (2019): 1-14.
[6] Tahir, A., Morison, G., Gibson, R., and Skelton, D.: A novel functional link network stacking ensemble with fractal features for multichannel fall detection. Cognitive Computation 12, no. 5 (2020): 1024-1042.
[7] RasberryPi Inc.: RP2040 A microcontroller chip designed by Raspberry Pi.

https://www.raspberrypi.org/documentation/rp2040/getting-started/ (Accessed 2nd June, 2021)
[8] Arduino Inc.: Arduino Nano, https://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano. (Accessed 2nd June, 2021)
[9] NodeMCU, https://www.nodemcu.com/index_en.html (Accessed 2nd June, 2021)
[10] Adafruit Inc.: Adafruit FLORA. https://learn.adafruit.com/category/flora (Accessed 2nd June, 2021)
[11] STMicroelectronics Ins.: iNEMO-Inertial Modules, https://www.st.com/en/mems-and-sensors/inemo-inertial-modules.html (Accessed 2nd June, 2021)
[12] iENSO Inc.: iVS-AWV3-AR0521. https://www.ienso.com/product/ivs-awv3-ar0521/ (Accessed 2nd June, 2021)
[13] Tahir, A., Morison, G., Skelton, D.A. and Gibson, R.M., 2020. Hardware/software co-design of fractal features based fall detection system. Sensors, 20(8), p.2322 (2020).
[14] Xilinx: Zynq-7000 SoC Data Sheet: Overview. July 2, 2018 https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf (Accessed 2nd June, 2021)
[15] Apple Inc, https://support.apple.com/en-us/HT208944 (Accessed: 1st June, 2021)
[16] Ahmet Cengizhan DIrican and Selim Aksoy. “Step counting using smartphone accelerometer and fast fourier transform”. In: Sigma J. Eng. Nat. Sci 8 (2017), pp. 175–182.
[17] Alexis Nez et al. “Comparison of calibration methods for accelerometers used in human motion analysis”. In: Medical engineering & physics 38.11 (2016), pp. 1289–1299.
[18] Andrey D Ignatov and Vadim V Strijov. “Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer”. In: Multimedia tools and applications 75.12 (2016), pp. 7257–7270.
[19] Jan Meyer, Paul Lukowicz, and Gerhard Troster. “Textile pressure sensor for muscle activity and motion detection”. In: 2006 10th IEEE International Symposium on Wearable Computers. IEEE. 2006, pp. 69–72.
[20] Md Uddin, Weria Khaksar, Jim Torresen, et al. “Ambient sensors for elderly care and independent living: a survey”. In: Sensors 18.7 (2018), p. 2027.
[21] Luis Sigcha et al. “Deep learning approaches for detecting freezing of gait in Parkinson’s disease patients through on-body acceleration sensors”. In: Sensors 20.7 (2020), p. 1895.
[22] Syed M Adnan et al. “Fall detection through acoustic local ternary patterns”. In: Applied Acoustics 140 (2018), pp. 296–300.
[23] O Dorgham, Sanad Abu Rass, and Habes Alkhraisat. “Improved Elderly Fall Detection by Surveillance Video using Real-Time Human Motion Analysis”. In: International Journal of Soft Computing 12.4 (2017), pp. 253– 262.

[24] Koldo De Miguel et al. “Home camera-based fall detection system for the elderly”. In: Sensors 17.12 (2017), p. 2864.

[25] Angela Antony and GR Gidveer. “Live Streaming Motion Detection Camera Security System with Email Notification using Raspberry Pi”. In: IOSR Journal of Electronics and Communication Engineering (IOSRJECE), Special Issue-AETM’16 (2016), pp. 142–147.

[26] Shengke Wang et al. “Human fall detection in surveillance video based on PCANet”. In: Multimedia tools and applications 75.19 (2016), pp. 11603– 11613.

[27] Abhijit Bhattacharya and Rodney Vaughan. “Deep Learning Radar Design for Breathing and Fall Detection”. In: IEEE Sensors Journal 20.9 (2020), pp. 5072–5085.

[28] Zhengyou Zhang. “Microsoft kinect sensor and its effect”. In: IEEE multimedia 19.2 (2012), pp. 4–10.

[29] Utkarsh Tripathi et al. “Advancing remote healthcare using humanoid and affective systems”. In: IEEE Sensors Journal (2021).

[30] Kamal Sehairi, Fatima Chouireb, and Jean Meunier. “Elderly fall detection system based on multiple shape features and motion analysis”. In: 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). IEEE. 2018, pp. 1–8.

[31] Baris Erol, Moeness G Amin, and Boualem Boashash. “Range-Doppler radar sensor fusion for fall detection”. In: 2017 IEEE Radar Conference (RadarConf). IEEE. 2017, pp. 0819–0824.

[32] Branka Jokanovic, Moeness Amin, and Fauzia Ahmad. “Radar fall motion detection using deep learning”. In: 2016 IEEE radar conference (RadarConf). IEEE. 2016, pp. 1–6.

[33] Baris Erol et al. “Wideband radar based fall motion detection for a generic elderly”. In: 2016 50th Asilomar Conference on Signals, Systems and Computers. IEEE. 2016, pp. 1768–1772.

[34] Mingyang Wang, Yimin D Zhang, and Guolong Cui. “Human motion recognition exploiting radar with stacked recurrent neural network”. In: Digital Signal Processing 87 (2019), pp. 125–131.

[35] Baris Erol and Moeness Amin. “Generalized pca fusion for improved radar human motion recognition”. In: 2019 IEEE Radar Conference (RadarConf). IEEE. 2019, pp. 1–5.

[36] Jijun Zhao et al. “R-DEHM: CSI-based robust duration estimation of human motion with WiFi”. In: Sensors 19.6 (2019), p. 1421.

[37] Nishtha Chopra et al. “THz time-domain spectroscopy of human skin tissue for in-body nanonetworks”. In: IEEE Transactions on Terahertz Science and Technology 6.6 (2016), pp. 803–809.

[38] Sadhana Lolla and Amy Zhao. “WiFi Motion Detection: A Study into Efficacy and Classification”. In: 2019 IEEE Integrated STEM Education Conference (ISEC). IEEE. 2019, pp. 375–378.

[39] Tao Wang et al. “Wi-Alarm: Low-Cost Passive Intrusion Detection Using WiFi”. In: Sensors 19.10 (2019), p. 2335.
[40] Ikram Ud Din et al. “The Internet of Things: A Review of Enabled Technologies and Future Challenges”. In: IEEE Access 7 (2019), pp. 7606– 7640.
[41] John Teel. Comparison of Wireless Technologies: Bluetooth, WiFi, BLE,
Zigbee, Z-Wave, 6LoWPAN, NFC, WiFi Direct, GSM, LTE, LoRa, NBIoT, and LTE-M.
url: https://predictabledesigns.com/wireless_ technologies_bluetooth_
wifi_zigbee_gsm_lte_lora_nb-iot_ltem/ (visited on 06/24/2021).
[42] IoT Standards & Protocols. 2020. url: https://www.postscapes.com/ internet-of-things-protocols/ (visited on 06/24/2021).
[43] Torgyn Shaikhina et al. “Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation”. In: Biomedical Signal Processing and Control 52 (2019), pp. 456–462.
[44] Banu Sa¸clı et al. “Microwave dielectric property based classification of renal calculi: Application of a kNN algorithm”. In: Computers in biology and medicine 112 (2019), p. 103366.
[45] Kunming Li et al. “Research on KNN Algorithm in Malicious PDF Files Classification under Adversarial Environment”. In: Proceedings of the 2019 4th International Conference on Big Data and Computing. 2019, pp. 156– 159.
[46] Shujun Huang et al. “Applications of support vector machine (SVM) learning in cancer genomics”. In: Cancer Genomics-Proteomics 15.1 (2018), pp. 41–51.
[47] Manas Jain et al. “Speech emotion recognition using support vector machine”. In: arXiv preprint arXiv:2002.07590 (2020).
[48] Jiang Wang et al. “Cnn-rnn: A unified framework for multi-label image classification”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 2285–2294.
[49] Shouxiang Wang et al. “Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting”. In: International Journal of Electrical Power & Energy Systems 109 (2019), pp. 470–479.
[50] Peter D Chang et al. “Hybrid 3D/2D convolutional neural network for hemorrhage evaluation on head CT”. In: American Journal of Neuroradiology 39.9 (2018), pp. 1609–1616.
[51] Aifeng Ren et al. “Machine learning driven approach towards the quality assessment of fresh fruits using non-invasive sensing”. In: IEEE Sensors Journal 20.4 (2019), pp. 2075–2083.
[52] Maha Elbayad, Laurent Besacier, and Jakob Verbeek. “Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction”. In: arXiv preprint arXiv:1808.03867 (2018).
[53] Juezhao Yu et al. “2D CNN versus 3D CNN for false-positive reduction in lung cancer screening”. In: Journal of Medical Imaging 7.5 (2020), p. 051202.
[54] Robert W. Broadley et al. “Methods for the Real-World Evaluation of Fall Detection Technology: A Scoping Review”. In: Sensors 18.7 (2018). issn: 1424-

8220. doi: 10.3390/s18072060. url: https://www.mdpi. com/1424-8220/18/7/2060.

[55] Omar Aziz et al. “Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets”. In: PLOS ONE 12.7 (July 2017), pp. 1–11.

[56] F. Bloch et al. “Evaluation under real-life conditions of a stand-alone fall detector for the elderly subjects”. In: Annals of Physical and Rehabilitation Medicine 54.6 (2011), pp. 391–398. issn: 1877-0657.

[57] Glen Debard et al. “Camera-based fall detection using real-world versus simulated data: How far are we from the solution?” In: Journal of Ambient Intelligence and Smart Environments 8.2 (Mar. 2016), pp. 149–168. issn: 18761364.

[58] Yuxi Wang, Kaishun Wu, and Lionel M. Ni. “WiFall: Device-Free Fall Detection by Wireless Networks”. In: IEEE Transactions on Mobile Computing 16.2 (2017), pp. 581–594.