Missing Values Imputation & Classification Using Stream Mining Algorithms in Internet of Things (IoT)

by

NS Marvi Waheed

Supervisor

Dr Hammad Afzal

A thesis submitted to the faculty of Electrical Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Electrical Engineering

July 2020

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by Ms. **NSMarvi Waheed**,Registration No.**00000104123** of Military College of Signals has been vetted by undersigned,found complete in all respect as per NUST Statues/Regulations, is free of plagiarism,errors and mistakes and is accepted as partial fulfillment for award of MS degree. It is furthercertified that necessary amendments as pointed out by GEC members of the studenthave been also incorporated in the said thesis.

Signature:______________

Name of Supervisor:**Dr Hammad Afzal, PhD**

Date: ____

Signature (HoD):______________________________

Date: ____

Signature (Dean):_____________________________

Date: ____

ABSTRACT

The Internet of Things (IoT) has established itself as an indispensable part of current age of user centric connectivity. The domain of IoT covers a widespread spectrum of daily life applications such as smart healthcare facilities which have greatly benefitted with the evolutionary advancements in sensor technologies and IoT. This progress has led to innovative developments like Human Activity Recognition (HAR) systems, smart movement detectors, fitness tracking, ingestible sensors, personalized emergency response systems and Fall Detection Systems (FDS) etc. Fall detection is now a pertinent publicconcern because of high prevalence and detrimental impact of falls on the young and the elderly. A fall detection system gathers information from sensors to differentiate between falls and activities of daily life (ADLs). Hence, the integrity of collected data becomes imperative. A pressing challenge when dealing with wearable sensors to detect falls is that of unreliable data delivery and loss of information leading to missing values observed in data. This missingness can occur due to several reasons and has crucial effects on the performance of a fall detection system resulting in inaccurate, faulty outcomes. Dealing with such insufficient and incorrect sensor data becomes critical for patient health and safety. This research investigates the missing data problem in terms of IoT applications and in particular for sensor based fall detection. Moreover, the current imputation methods and proposed solutions are also analyzed. This analysis leads to the conclusion that current solutions for missing data problem in fall detection systems are very limited. The manuscript proposes a deep learning based fall detection solution to handle missing values and identify falls. This is achieved by a Recurrent Neural Network model, with underlying stacked bidirectional LSTM blocks, which treats fall detection as a sequence classification problem and exploits the patterns and intrinsic association between the variables in data.

DEDICATION

Dedicated to

My parents for their incredible support and kindness that always lightened my ways

&

My husband and daughter, as they are my strength in everything I do.

ACKNOWLEDGEMENTS

I am thankful to God Almighty (all praise belongs to Him) for giving me the strength,ability and perseverance to achieve my objectives and successfully complete this research work.

I am highly grateful to my supervisor, Dr. Hammad Afzal for guiding and supporting me throughout my research work. His timely encouragement has led me to complete this research successfully.He not only imparted significant knowledge to me but also contributed extensively in my work. I am highly obliged to Maj Khawir Mehmood for dedicating his precious time for my research work and helping me develop my ideas. His unwavering enthusiasmand keen insights have been especially valuable to keep me constantly motivated about my research work. My profound gratitude goes to my co-supervisor, Dr. Seemab Latif and my Guidance and Evaluation Committee (GEC) members, which included Dr. Muhammad Imran and Dr. Alina Mirza, for their prompt assistance, useful advice and constructive feedback that helped me accomplish my targeted goals.

Above ground, I am greatly indebted to my family and in-laws for their unconditional love, sincere prayers and cordial cooperation without which this research would not have been possible. A special word of gratitude for my husband, Umair, for always showing confidence in my work, supporting my ambitions and pushing me forward. I would like to paymy gratitude to my close friends for the persistent encouragement, academic supportand kind words which boosted my morale throughout. Finally, I am grateful for my daughter, Parnia, who kept me going and is the reason behind it all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

Activities of Daily Life	ADLs
Bidirectional Long Short-Term Memory Network	BiLSTM
Fall Detection System	FDS
Gated Recurrent Units	GRU
Human Activity Recognition	HAR
Internet of Things	IoT
Internet of Medical Things	IoMT
K-Nearest Neighbors	KNN
Long Short-Term Memory Network	LSTM
Missing Completely at Random	MCAR
Recurrent Neural Network	RNN

CHAPTER: 1

INTRODUCTION

1.1 Background

The current age of connectivity has opened a plethora of domains for user centric applications and environments. This development stems from the evolution of computers where everything is getting faster, smaller and easy-to-use. Although Moore's Law - a prediction made by Gordon Moore in 1965 [1] that reduction in transistor size shall result in exponentially cheaper and faster computing (see Figure 1.1), is widely expected to hold true only until 2020-2025. However, it exhibits the remarkable advancements in computing in the last 50 years.

Another field that has received considerable popularity in recent years is that of wireless sensors for real time monitoring of surroundings like temperature, humidity, pressure, positioning, acceleration, synchronization measurements etc. The sensors with their flexible nature provide infinite potential for effective problem solving in fields of healthcare monitoring, environmental sensing, surveillance and threat detection.

Because of commoditization of sensors and the evolution of computing and communication networks, the Internet of Things (IoTs) has emerged as an integral part of current age of connectivity.

1.1.1 Internet of Things – What's it about?

The term Internet of Things (IoT)was first devised in 1999 by Kevin Ashton which refers to a meshed interlink of daily life objects.IoT is thought of as asystem of physical objects in the world that connect to the internet via a sensor.It is an evolving prototype that connects the pervasive existence around us of diverse utilities to the Internet by making use of wireless/wired technologies to meet desired goals. It can be viewed as an intelligent global network with components consisting of self-configuring capabilities that connect billions of devices via the Internet by using a variety of communications technologies. Theinteroperability among different types of devices and communications technologies is swiftly transforming IoT into a highly heterogeneous ecosystem. [2] The fundamental principles of interconnection of IoTs allow access to remote sensor data and control of complex physical environment from a distance that inherently permits efficient decision making, realistic automation, pragmatic productivity, greater wealth generation, enhanced public safety and national defence.

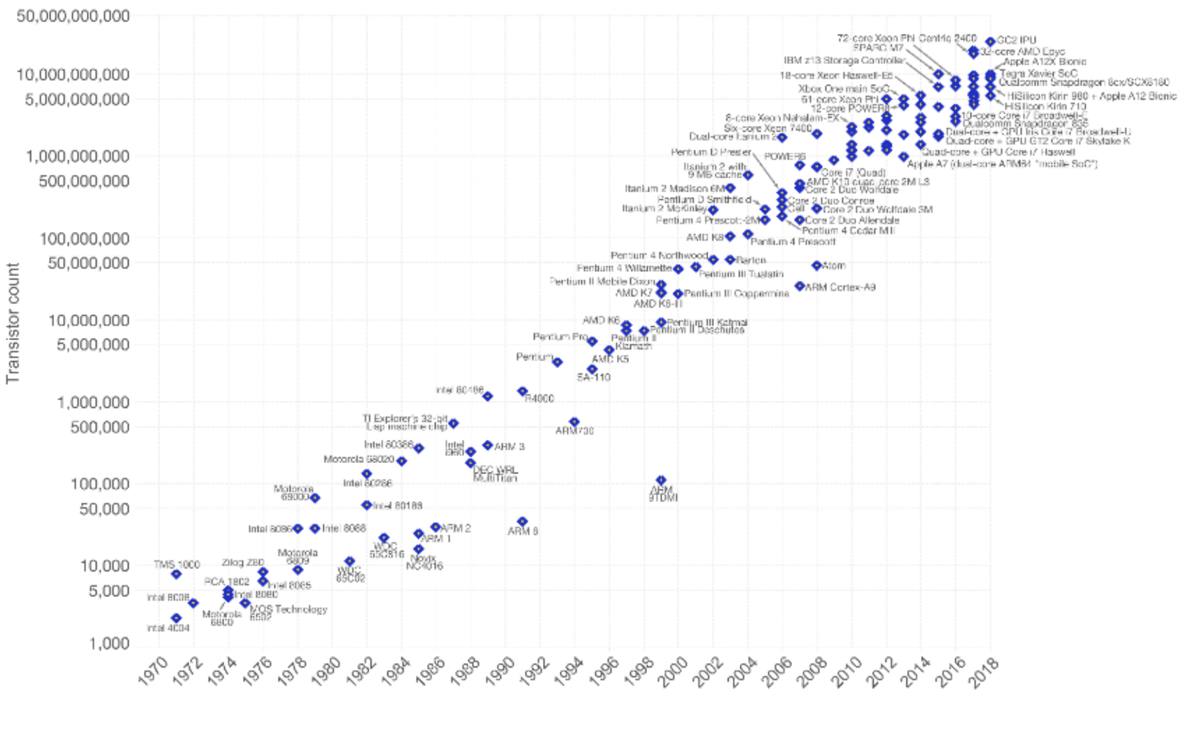


Figure 1.1: The number of transistors in integrated circuits chips (1971-2011) [1]

According to the 2020 conceptual framework [3], the Internet of Things can be expressed as a straightforward formula:

$$\text{IoT} = \text{Services} + \text{Data} + \text{Networks} + \text{Sensors} \quad (1.1)$$

1.1.2 The IoT Architecture

The IoT integrates the DIKW (Data, Information, Knowledge, Wisdom) hierarchy to learn from the huge amounts of cumulated sensor data. The knowledge of basic architecture of IoT is imperative for its successful implementation.

The conventional IoT includes three layers consisting of perception layer, network layer and application layer. Additional layers have been added to this model with the introduction of support layer that lies in between the network and application layers.

Another widely accepted model of IoT comprises of seven layers (see Figure 1.2) as described below:

- **Level 1: The Things Layer**

  It consists of connected physical objectswhichessentially enable the IoT environment such as sensors, micro controller units and mobile devices.This layer acts as the real endpoint for IoT.

- **Leve 2: Connectivity/Edge Computing Layer**

  It includes a distributed architecture which consists of a variety of networks and communication protocols essential for connectivity and edge computing. Data processing occurs at the edge.

- **Leve 3: Global Infrastructure Layer**

  Since most IoT solutions are implemented within a cloud infrastructure, the third layer of IoT consists of a comprehensive set of integrated cloud services.

- **Level 4: Data Ingestion Layer**

  It is responsible for data cleaning, streaming and storing.

- **Level 5: Data Analysis Layer**

  Data analysis taskslike reporting, mining, machine learning of data occurs at this layer.

- **Level 6: The Application Layer**

  It consists of custom applications which utilize the actual “things” information.

- **Level 7: People and Process Layer**

  It integrates theusers, businesses, partnerships and management derived from the fruitful information of IoT computing.

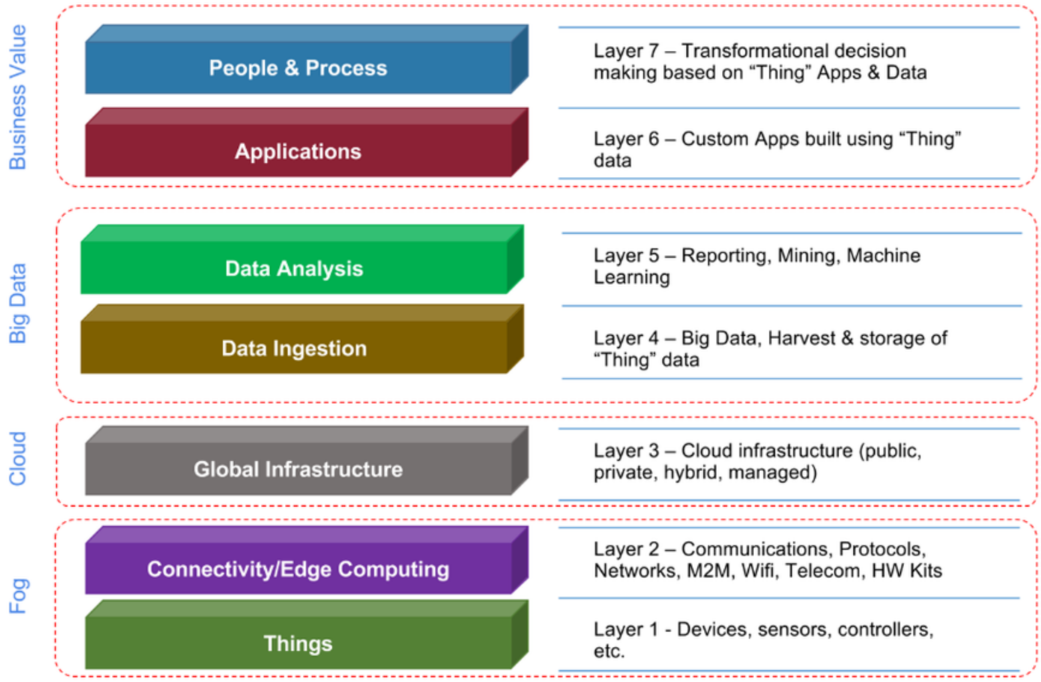


Figure 1.2 A layered view of the IoT architecture

1.1.3 Challenges of IoT

With its integral advantages the IoT delivers undeniable challenges as well. The IoT is essentially an environment where enormous amounts of heterogenous entities are interconnected. These heterogenous entities such as sensors have limited capabilities of their own such as less power consumption, small sizes, limited memory resources and little battery life. All in all, these challenges make IoT a resource constraint network which strives on low computational costs, efficient energy and memory management.

Moreover, the pertinent security threats impose greater risks like identity and data thefts, remote recording, ransomware and malware attacks, denial of service attacks and manipulative use of social engineering to obtain sensitive information etc.

Apart from infrastructure complexities and security threats another important challenge in an IoT environment is that of imperfect data extending from noisy, unreliable data to missing sensor data all together. This blockade of missing data can occur due to synchronization disturbances, fluctuating network communication, lossy sensor devices, untrustworthy environmental aspects and miscellaneous device malfunctions that cause deficiencies of data incompleteness. Occurrence of faulty incomplete data due to missing values is a common phenomenon in IoT that leads to critical influence on the inferences drawn from the accumulated data. If not managed appropriately, such imperfect data results in inaccurate and questionable analytical results.

1.1.4 Areas of IoT Application - IoMT

The domain of IoT covers a widespread spectrum of daily life applications such asintelligent transportation, smart buildings, smart healthcare facilities, positioning and navigation field, and logistics field.The potential impact of IoT is predicted to bring forward surplus business opportunities and to expandthe economic growth of IoTbased services. According to a report published by McKinsey [4], the annualeconomic implications of IoT in 2025 would amount to $3.9 trillion to $11.1

trillion a year. About 41% of this market share is earned by IoT based healthcare services.

In fact, in the recent years the field of medicine and healthcare have greatly benefitted with the evolutionary advancements in sensor technologies and IoT providing user friendly and efficient services to patients. So much so that the term *"Internet of Medical Things (IoMT)"* has been coined. Some of these innovative developments include Human Activity Recognition (HAR) systems, smart movement detectors, fitness tracking, ingestible sensors, asset management systems, personalized emergency response systems, Fall Detection Systems (FDS), diagnostics, development of robust EHR systems etc. With IoT enabling healthcare services to achieve maximum potential, the possibilities are endless.

1.1.5 Fall Detection Systems

A fall is as an occurrence of a subject coming at rest with the ground level or lower, as defined by the World Health Organization (WHO). From children to elderly people, falls are encountered by all. Unintentional falls can occur due to various reasons like accidental situations, performing high risk strenuous activities, subject surrounding factors like slippery floors, physical factors like loss of consciousness, tripping, poor balance, effects of wrong or overdose of medication etc. The impact and detrimental effects caused by a fall are dependent on the severity of the fall and the age and physical wellbeing of the subject experiencing the fall. For example, a standard fall will have varying effects on elderly patients as compared to younger audience. After road traffic injuries, unintentional falls related injuries make up for the majorcause of death with an approximation of 650,000 fatal falls occurring each year [5]. Falls are considered as a fatal threat to morbidity and mortality of elderly people. Around the world, mortalityestimates are highest among elderly population of 60 years and above. Approximately 50% of injury-relatedhospital admissions are observed in the elderly of 65 or more. As a result,an estimated 40% of the injury-related deaths occur due to falls in the senior population [5].

The advancement in microsensors and IoT have paved way for research in mobile healthcare monitoring like Fall Detection Systems (FDS). Fall detection systems promise to provide efficient and reliant alarms that can lead to immediate medical assistance (see figure 1.3). Generally, the fall detection systems rely on ambience sensing devices and/or wearable sensing devices. Ambient sensors are usually comprised of video cameras used for surveillance and monitoring whereas

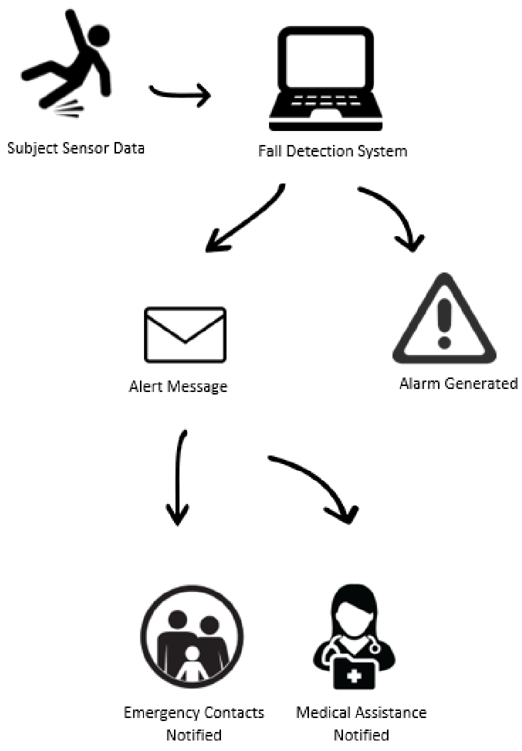


Figure 1.3 A typical IoT based Fall Detection System

wearable sensor-based system analyzes raw sensor data to recognize human activity. The downside of ambient based systems is the intrusion of personal privacy and the ineffectiveness to correctly identify falls for highly mobile subjects. On the other hand, wearable fall detection systems protect user privacy, works for frequently mobile subjects and provide results on new unknown locations as well. These wearable devices can be utilized in a non-invasive approach to safeguard user comfort.

1.1.6 Missing Values – A Challenge for FDS

One challenge when working with wearable sensors in FDS is that of unreliable data delivery and loss of information leading to missing values. Missingness in data can appear due to several shortcomings like power outage of the sensing devices, local interferences, synchronization disturbances, fluctuating network communication, lossy sensor devices, untrustworthy environmental aspects, security attacks etc. These incomplete observations can severely hamper the performance of the fall detection system leading to inaccurate, faulty outcomes. Dealing with such type of insufficient, incorrect sensor data based monitoring system can prove fatal for patient health and safety. Hence providing effective solutions to detect precise activity recognition with fall detection systems defines justifiable research opportunities.

1.2 Problem Statement

Statistics show that next to road traffic injuries, unintentional falls related injuries make up for the second leading cause of death with an approximation of 650,000 fatal falls occurring each year. Hence FDS based on sensor technology is a thriving topic of research since the past decade. Fall detection systems integrate the basics of an IoT network including wide scale streaming data, heterogeneity, correlation of space and time and highly noisy data. In the given challenging environment, these Fall detection systems are often subjected to the concerning predicament of observing missing values in sensor data or accumulating incomplete and thus incorrect information which results in providing inaccurate detection of falls. If neglected, the missing values can yield biased, untrue predictions which degrade the performance of the monitoring system and put patient safety at high risk. Consequently, the aim of this research is to formulate an efficient solution for human activity recognition which identifies legit falls and distinguishes daily life activities in presence of missing values in a wearable sensor-based Fall Detection Systems (FDS).

1.3 Goals and Objectives

The objectives of this research are highlighted as follows:

- To analyze the performance of FDS based on imperfect data streams with missing sensor values.
- To study and analyze the existing solutions for fall detection in presence of incomplete, noisy data.
- To propose a reliable technique for imputation of missing values in data.
- To propose an effective mechanism for detecting accurate falls in the presence of missing values in sensor observations for an IoT based FDS.
- To prove the concept of the recommended mechanism by augmenting results.

1.4 Research Contributions

The research works on aiming to provide a fall detection system that performs in the presence of missing values by replacing the noisy missingness through imputation using stacked Bidirectional LSTMs. An effectiveness analysis is further carried out to give a detailed breakdown of performance evaluation.

1.4.1 Research Significance

In the current age of big data where IoT applications give rise to new synergetic services that encompass the capabilities of isolated embedded systems, provision of safe, secure and trusted information is of utmost importance for the sustenance of the innovative concept of IoT. In order to make valuable contributions to the healthcare services, IoT based FDS need to take into account the disparities observed in sensor data and devise required mitigation procedures. Improvements made in this area of research can yield timely responses to falls, effective rescue and medical assistance, increase in the survival rates for injury effected patients, safer nursing home and old age living facilities, reduction in the healthcare costs generated due to fall related injuries. Consequently, this research can enable healthcare monitoring services to reach their optimal potential.

1.4.2 Relevance to National Needs

With Fall Detection Systems incorporating solutions for missing value in sensor observations as proposed in this research E-health platforms can safely utilize the collected, crucial patient data where results and analyses are sensitive to imperfections like lossy data, delays and fluctuations**.** The national healthcare system can greatly benefit from the improved Fall Detection Systems when installed in crucial environments like Intensive Care Units (ICUs) of hospitals for patient vitals monitoring, nursing homes for surveillance of elderly wellbeing, improving response time of emergency rescue providers, E-health applications and Smart Home Projects etc.

1.5 Manuscript Organization

The research is organized as follows. Chapter 2 presents a comprehensive literature review along with existing solutions for fall detection and related works. Chapter 3 describes in detail the proposed fall detection mechanism while Chapter 4 reports the analytical results and performance evaluation of the proposed solution. Finally, Chapter 5 concludes the research.

CHAPTER: 2

LITERATURE REVIEW

2.1 Fall Detection Systems

The basic aim of a FDS is to notify in an event of fall. These systems are designed with the essential purpose of identifying falls and distinguishing them from Activities of Daily Life (ADL). The researchers in [6] categorize fall detection in three techniques; vision-based approach, ambient sensor-based approach and wearable sensor-based approach. To meet the end of these approaches several experimental tactics have been devised, where each procedure achieves fall detection using different techniques. A classification of various fall detection techniques that fall under the hierarchy of approaches is essential to understand the research done on the topic. (see Figure 2.1)

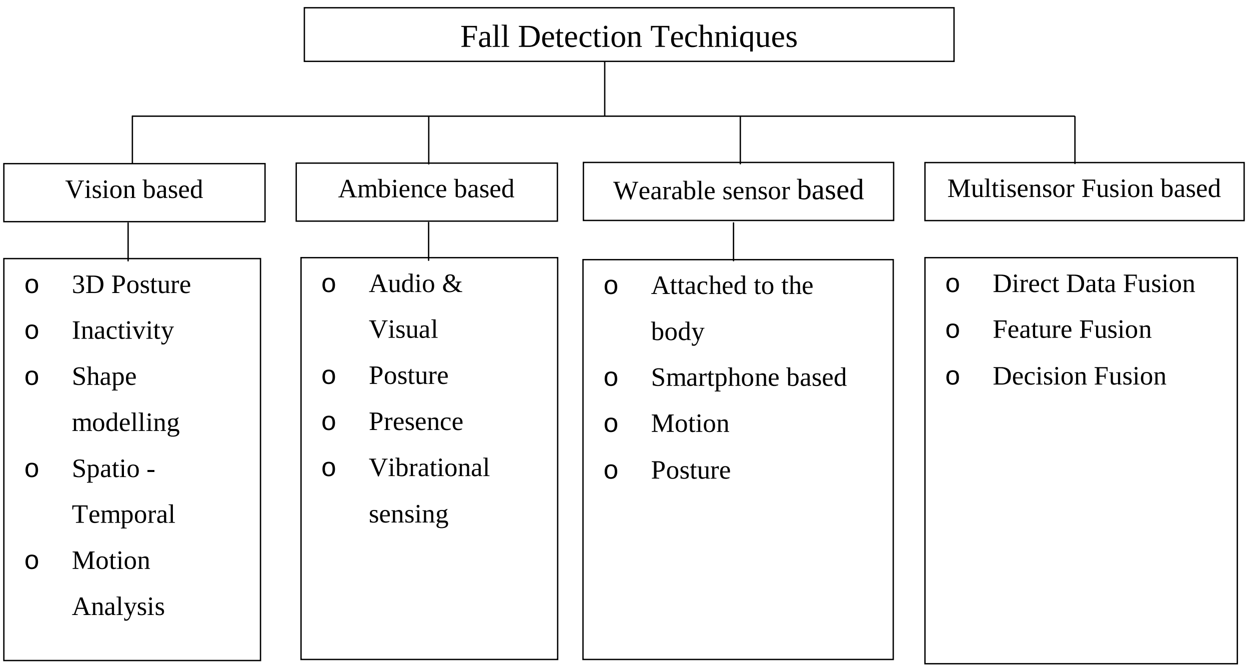


Figure 2.1 Classification of the Fall Detection Techniques

2.1.1 Vision Based Approaches

Since cameras are progressively utilized in home surveillance and patient monitoring systems, a vision-based fall detection technique includes single or multiple cameras installed in an overhead position to monitor an environment and detect falls. It is a less intrusive approach where cameras can be used to monitor multiple actions or activities (see Figure 2.2). The vision-based fall detection approach can loosely be classified into fall identificationby employing a single RGB camera, 3D posture method using depth camera or multiple RGB cameras [7].

The area of using single RGB camera to detect falls has been thoroughly researched. Features related to shape along with detection of inactivity and human motion analysis are factors commonly utilized in the single RGB camera approach [7]. Mirmahboub et al. [8] propose feature extraction from silhouette of a person generated by simple background separation techniques. Background separation techniques is achieved by the running moving average method. The paper exploits drawbacks in the background separation and proposes to utilize fluctuations in the silhouette area as an important feature that is robust from viewpoint. An SVM classifier is used to make classifications based on silhouette related features. The comparison between proposed feature and benchmark techniques depicts a lower error rate for the silhouette technique along with the computational burden being low enough for hardware implementation to be executed within the camera. Chua et al. [12] propose a technique relying on human shape variation using vision technology for fall detection. The paper uses a three-point representation of human figure as a substitute for standard ellipse or bounding box methods. Detection of falls is achieved by analyzing the changes in simplified human silhouette.

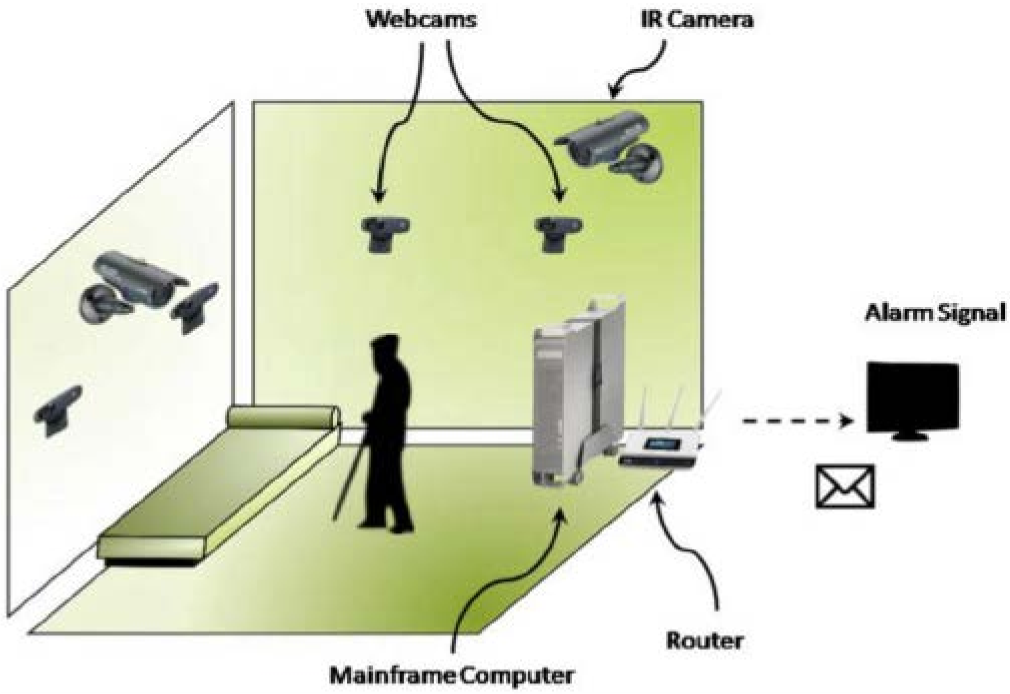


Figure 2.2 Fall Detection based on vision technology[7]

Motion analysis can be utilized to identify falls and Activities of Daily Life (ADL) based on the differences of motion patterns for respective events. Liao et al. [9] proposes detection of slip-only and fall events by using human silhouette shape variations and motion analysis. Energy analysis of operational and frequent area of motion in the integrated spatio-temporal energy (ISTE) map helps in obtaining the required motion measure. Anh Nguyen et al. [10] presentan indoor fall detection mechanism using single camera by deeply analyzing fall characteristics like alterations in human shape, motion orientation and magnitude. Motion histogram images are used to make improvements in motion characteristics. Falls are detecting by examining different features before, after and during the fall like speed of change in human shape, motion orientation and magnitude.

Activity recognition can be done through shape modelling by extracting spatio-temporal features. Foroughi et al. [11] works on fall detection based on combination of eigenspace technique for feature reduction and integrated time motion images (ITMI) as temporal templates. An MLP neural network with an average recognition rate of 89.99% is chosen as optimal classifier for human activity classification.

Another approach towards vision-based fall detection is the 3D head position analysis which is implemented by monitoring head position that dictates motion of considerable magnitude in a video streaming. Auvinet et al. [13] implements fall detection using Occlusion-resistant algorithm and Vertical Volume Distribution Ratio (VVDR) by reconstructing 3D shape of the subject. Vertical axis volume distribution is analyzed to detect falls and generate alarm when distribution is near to the floor. The approach achieves 99.7% sensitivity and enhanced specificity with the challenge of maintaining multi-camera adjustments and multi-camera video synchronizations. The overall technique results in complexity and increase in cost for system design.

Fall detection using 3D depth cameras initiated with the Time of Flight cameras is proposed by Diraco et al. [14]. However, this technology was rarely pursued by researchers due to expensive equipment costs. Recent innovations in depth sensing technologies like Microsoft Kinect has revived the use of depth cameras for vision-based fall detection. Fall detection is achieved by extracting significant features based on basic distance calculations between the top of person and the floor [15]. Kinect overcomes the challenges of using traditional cameras, providing captures of full figure 3D motion by making use of a camera, infrared projector and a specific microchip. Mastorakis et al. [16] introduces a computer vision-based real time fall detection technique utilizing Kinect's infrared sensor's ability to perform well in compromised lighting conditions. Inactivity and velocity calculations measured by analyzing changes in parameters of the 3D bounding box achieve a real time FDS that is independent of prior knowledge of the scene. A ceiling mounted Kinect infrastructure is presented by Gasparrini et al. [17] to utilize raw depth images to detect potential falls by preserving privacy in an indoor environment. The proposed method incorporates Ad-Hoc Segmentation algorithm to automatically detect falls.

Wang et al. [18] proposes a computer vision-based fall detection based on automatic feature learning techniques. The proposed method includes feature extraction by PCANet and training two SVM classifiers for detection of falls. Experimental results produce a specificity of 98.4% and a 93.8% sensitivity which come at par with common techniques using multiple cameras for fall detection.

2.1.2 Ambience Based Approaches

Ambience based devices integrate audio, video, vibrational signals measured through several environmental sensors to achieve detection of falls. It is the most basic system for detecting falls by monitoring and evaluating the combined effect of the subject of interest's environmental variations (see Figure 2.3). Vibrational data often recorded for monitoring, tracking and localization purposes is commonly used with audio-visual and other external sensors data to achieve ambience-based fall detection.

Proximity sensors are a common type of external sensors widely deployed for patient monitoring. These sensors are usually installed in an external device not attached to the patient's body, like a cane, walker or a mobility aid device. The sudden change in the user's position, motion and the proximity from the sensors are measured to detect falls. Hirata et al. [19] proposes a solid body link model by analyzing the center of gravity of the model in order to detect falls. The researchers propose a fall prevention control mechanism by analyzing the support polygon and the movement characteristics of the subject. However, these sensors are expensive and usually experience short proximity range which can lead to misinterpretations and faulty detection of falls. Pressure sensors also rely on the proximity principle for fall detection. When a subject is nearer to a sensing device the pressure is supposedly greater. Though inexpensive and non-obtrusive these sensors yield poor accuracy for detecting falls in general.

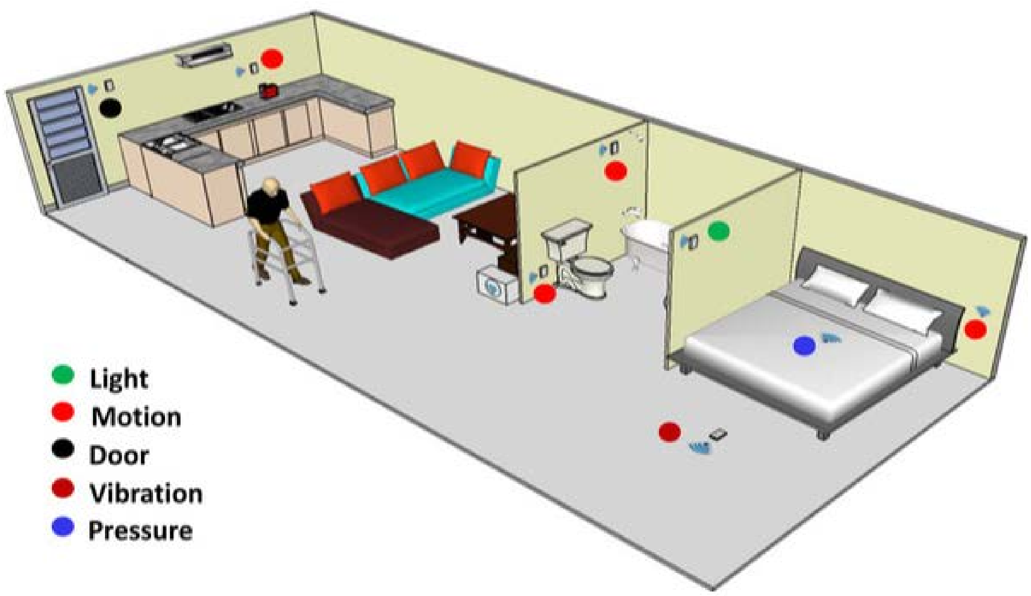


Figure 2.3 Ambience based sensing for patient monitoring [23]

Alwan et al. [20] propose a floor vibration pattern learning scheme to detect falls. The system depends on learning from floor vibrations generated by subjects performing ADLs to detect falls. The non-invasive approach yields high accuracy of detection specially in an event of user non-compliance to obtrusive methods. Sixsmith et al. [21] make use of the thermal imaging applications of wall mounted Pyroelectric IR sensor arrays for detecting falls in the elderly. Target movements and inactivity durations are utilized to detect falls without having to sense background by analyzing changes in IR flux. A bed exit detection mechanism is proposed in [22] by installing multiple sensors adjacent to the patient support surface. A processor monitors the sensor signals to report changes in patient body position. Automatic recalibration in case of substantial change or object movement along the axis of sensors is executed. The system is proposed for caregivers to monitor patient movements and falls.

Toreyin et al. [25] utilize sound, vibration and passive infrared (PIR) sensors to collect data on human activities of daily life (ADL). A Hidden Markov Model (HMM) is trained on ADL and unusual events for detection of falls. Decisions of the HMM depending on audio and PIR sensor data are integrated together for final decision makingregarding falls. In another research, Toreyin et al. [24] uses Hidden Markov Model (HMM) along with wavelet processing to differentiate between falling and ADLs like walking and sitting down based on audio-visual data.

Popescu et al. [32] present a FDSrelying on acoustic sensor comprised of linear arrays of electret condensers mounted on a pre-amplifier board. Information about sound height was captured by placing the sensor array along the z-axis. The drawback of this research is that only a single subject can be monitored in the vicinity.

Ambient Assisted Living (AAL) aims to deliver an ecosystem of sensing devices, computers, mobile devices, wireless networks etc. for patient monitoring, e-healthcare systems and telehealth systems. An important objective of AAL is to provide quality of life support to dependent subjects like elderly and handicapped patients in performing daily life activities. Yazar et al. [26], propose a multi-sensor Ambient Assisted Living ecosystem for tracking human footsteps, motion detection and unusual event detection like falls. A combination of PIR sensors along with vibration sensors detecting footsteps is used. The proposed system is affordable, privacy friendly and generates real-time detection of falls executed on chipKIT Uno32 microprocessors. Werner et al. [29] proposed an AAL solution called eHome to make elderly patients feel secure in the comfort of their homes. The proposed solution uses floor mounted accelerometers that gather the vibration patterns generated by a user. This information is shared wirelessly with a base station where data analytics lead to effective results. The results yield a 97% specificity with an 87% recall.

2.1.3 Wearable Sensor Based Approaches

Wearable sensor technology is unarguably the most employed method for reliable fall detection. An alternative to external sensors, body mounted wearable sensors are fastened to the body of subject of interest (see Figure 2.4). These sensors, collecting important data related to the patient’s body movement make up for an efficient solution for fall detection with their low costs,weight, small size, low power usage, portability, convenient usefulness and attractive features. Commonly used wearable sensors include accelerometer, heart rate sensor,gyroscope, magnetometeretc. which can be attached at multiple positions on the subject's body. A significant volume of research has also been conducted using multiple wearable detectors and fusing the results for optimum fall detection. This topic of research will be addressed in the section of multi-fusion sensors for fall detection.

Accelerometer sensors measure the variation in velocity of the body or speed divided by time. The measured acceleration is in units of (m/s)/s. Tri-axial accelerometer with X, Y and Z axes are used to determine thelocation of the body and its motion. Fall detection relies on the sudden increase of negative acceleration caused by shift in orientation from upright to lying flat position.

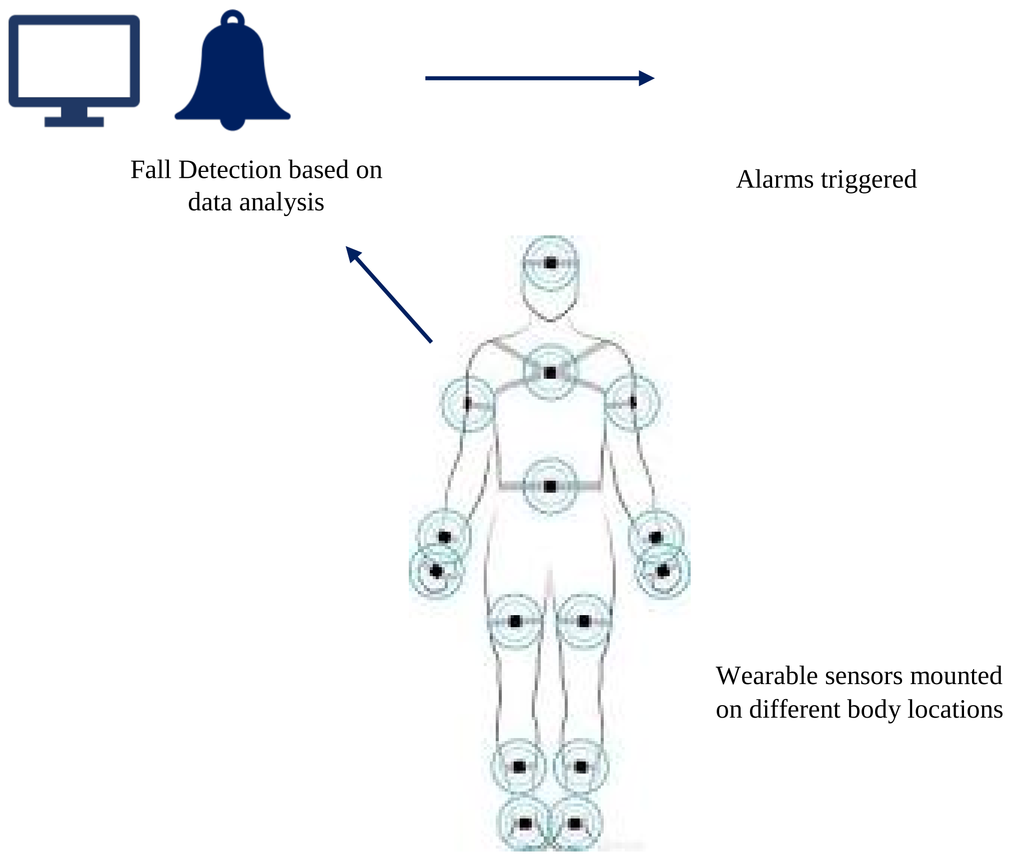


Figure 2.4 Wearable sensors-based Fall Detection

Chen et al. [27] propose a pelvis mounted tri-axial accelerometer sensor to identify human movements. Acceleration along the horizontal plane, reference velocity and vector addition of magnitude ofacceleration are used to make up three-level criteria for detecting falls. Acceleration along the horizontal plane asses the body inclination, reference velocity calculates if the body is at rest or motion and lastly, the sum vector magnitude of acceleration is used to analyze the spatial changes in acceleration in an event of fall. The eight fall scenarios to show the usefulness of the accelerometer-based data include walking, walking backwards, stoop, jump, sit to standing upright, stand to sitting down, standing still and lying on the bed. In another research, Chen et al. [28] propose to use inertial sensors (accelerometers) to create a wireless FDScomprised of two modules; a fall detection module and a remote module. The fall detection unitfunctions independently while the remote unit stores the sensor signals and raises alarms. The two modules communicate via wireless network. The system achieves a high specificity of 100% with 97% recall. Lai et al. [30] proposes an integrated technique to detect fall incidents in the elderly as well as the joint sensing of the injured body part in case of fall. A fall is traced when the acceleration obtained by the tri-axial accelerometer exceeds the normal acceleration range significantly. The gathered information is wirelessly transferred to a computer for further analysis. Wang et al. [31] propose a threshold reliant fall detection mechanism by accumulating data from various sensory devices. Combined data from accelerometer, cardiotachometer and smart sensors are used to approach a high detection accuracy of 97.5%.

Current smartphones are equipped with a diverse set of embedded sensors like biometrics, camera, GPS, proximity sensor, accelerometer, barometer, digital compass, magnetometer, microphone, gyroscope etc. Based on in built sensors functionality Smartphones can be used as wearable devices for detecting falls. Andò et al. [33] utilizes a smartphone accelerometer to create anFDS that differentiates between ADL and falls. The specificity and sensitivity metrics of the proposed methodmeet the standards for AAL. Figueiredo et al. [34] propose a threshold-based algorithm for detecting falls by the information provided from smartphone's embedded sensors in particular the tri-axial accelerometer, magnetometer and gyroscope. The algorithm is proposed to work in a low battery consumption set up. The chosen features to help differentiate between a fall and ADL were vertical acceleration, sum of the components of the acceleration vector, angle variation, change in orientation and gyroscope's angular velocity and angular acceleration. The performance evaluation depicts 100% sensitivity and 93% specificity. He et al. [35] discuss the challenges faced by conventional body worn sensors when detecting falls and propose a fall detecting mechanism byintegrating Fisher's discriminant ratio criterion and $J3$ criterion to create an algorithm for feature selection. The method utilizes built-in kinematic sensors (tri-axial accelerometer, gyroscope, and magnetic sensor) for data accumulation. A hierarchal classifier used to classify human activities reaches an accuracy of 95.03%, proving the practicality of utilizing embedded smartphone sensors for fall detection. A non-invasive approach with real-time fall detection and alarm reporting is proposed by Gravina et al. [36] using aa smartphone and body mounted accelerometer. Performance evaluation demonstrates sensitivity,precision and specificity metrics to be 97%, 90% and 83% respectively, triggering prompt emergency alerts using different modalities. Moreover, Habib et al. [37] review the smartphone-based fall detection techniques in detail and comprehensively evaluates the embedded inertial sensors, sophisticated wireless technology and latest open source operating systems for the purpose. The research concludes that smartphone-based FDS make a good alternative to the conventional methods but are restricted by the relatively low standard of built in sensors and the need to wear the smartphones in predetermined secure positions for smartphone-based FDS.

Gyroscope sensors measure the angular velocity which is the change in rotational angle per unit of time. The metric is degree per second. Most research approaches incorporate a combination of multiple sensors when using gyroscope to track angular velocity while few use gyroscopes exclusively for fall detection. Bourke et al. [38] propose a thresholdreliant algorithm for detecting fall events by using information from a bi-axial gyroscope sensor array. The trunk mounted sensor is used to calculate roll angular velocities and pitch. The approach sets thresholds for resultant values of trunk angle, angular velocity and angular acceleration each. In case of an event surpassing these set thresholds, alarms are triggered, and a fall is detected. The proposed system attains a robust 100% accuracy of distinguishing falls from ADL when data analysis is performed using MATLAB.

2.1.4 Multi-Sensor Fusion Based Approaches

Multisensor fusion is an approach to integrate information from different sensor sources to formulate a unified picture. In comparison to a single sensor method, multisensor fusion approach is set to produce robust measurements and accurate detections. Sensor fusion approaches can fall into three categories from the viewpoint of data processing; fusion of direct data,fusion of decisions and fusion o features. (see Figure 2.5) [47]. There are numerous ways to reach this approach. A basic idea is to use the same type of sensor positioned at different locations e.g. accelerometers attached at different body parts or vibrational sensors installed at different locations in a vicinity. Alternatively, a multimodal device composed of multiple sensors e.g. accelerometer, barometer, gyroscopes etc. can be worn at a single location or multiple such devices can be used at different locations. Latest healthcare systems generally utilize the data fusion approach to demonstrate enhanced performances with respect to accuracy and credibility in comparison to a single source dependent system. Data fusion algorithms are deployed to avoid intrinsic ambiguities when unrelated sensors are exploited [39]. Independent data analysis is performed for each sensor technology followed by fusion procedure as a last measure for detecting falls. Two main categories used to classify the fusion based systems for fall detection include context aware sensors and wearable sensors.

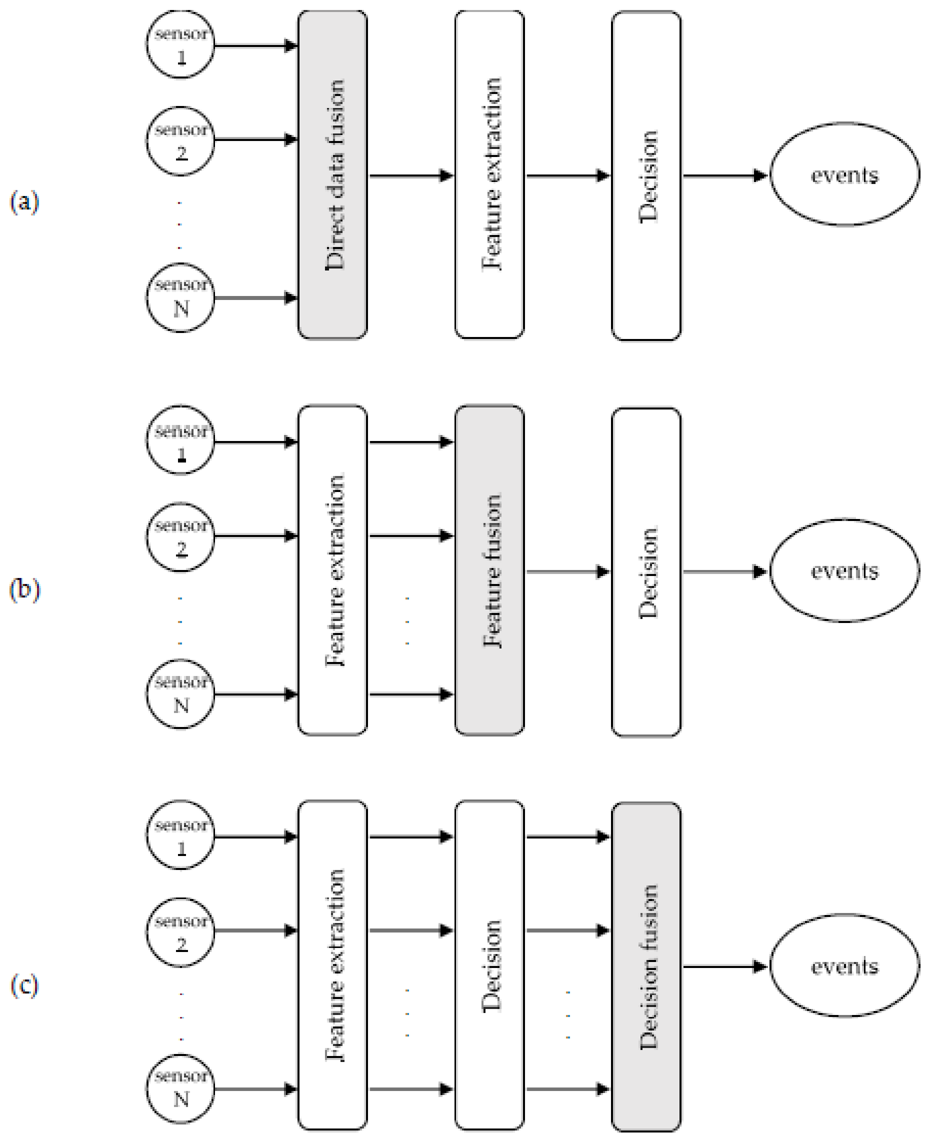


Figure 2.5 Classes of fusion architectures at different levels: (a) data, (b) featureand (c) decision

Camera sensors accompanied by acoustic sensors are a common approach for context aware sensors fusion for detecting falls. Li et al. [40] uses a similar approach for thresholding-based fall detection with segmentation techniques. Yazar et al. [41] propose fusion of two PIR sensors with a vibration sensor in a winner-takes-all (WTA) decision fusion algorithm approach to detect falls. The algorithm includes preliminary processing of data collected from the multiple sources. Li et al. [42] propose a combination of wearable tri-axial accelerometer and context aware sensors like a depth camera and a micro-Doppler radar for fusion system-based fall detection. The fusion of mentioned heterogenous sensors results in improvement of overall performance with overall classification accuracy increasing up to 91.3%.

Wearable sensors fusion based approaches mainly rely on accelerometer as the main source of information delivery complimented by other body worn sensors like gyroscopes, magnetometers, location tags and barometric pressure sensors etc. Cillis et al. [43] propose the fusion of accelerometer and gyroscope in a sliding window method using sudden variations in the subject of interest's orientation along with abrupt changes in the AVM to identify a fall. The approach incorporates a threshold-based fall detection.Ando et al. [44] propose fusion of accelerometer and gyroscope to analyze patterns of performed activities, behaviors of motion, and the correlation function for categorizing falls. The results depict a specificity of 99% with a 78% sensitivity.Pierleoni et al. [45] employ accelerometer, gyroscope, barometer and magnetometer to detect fusion based falls with a quaternion filter extracting acceleration relative to Earth's frame from the IMU sensors and the barometer estimating the altitude. A set of thresholding standards applied on several features like altitude, angular velocity and acceleration identify falls. Consequently, a 100% sensitivity and a 99% specificity is achieved.

More recently, Ntanasis et al. [46] propose using Machine Learning with the adequate amount of data gathered from multiple heterogenous (accelerometer, gyroscope, magnetometer) sensors. A waist mounted IMU sensory device with Support Vector Machine (SVM) classification result in 99.50% sensitivity and 99.19% specificity.

2.2 IoT based Fall Detection

Internet of Things (IoT) is an idea that embraces the use of multiple technologies in order to facilitate real-world objects with the use of internet. Wireless sensor networks (WSNs) that play an fundamental part in an IoT environment can help digitize quantities of the physical world like temperature, pressure, humidity, acceleration etc. Data analysis of the collected sensor information is commonly done by fog and cloud computing in an IoT environment (see Figure 2.6). With the areas of application of IoT broadening into healthcare, IoT based fall detection techniques are a thriving topic of research.

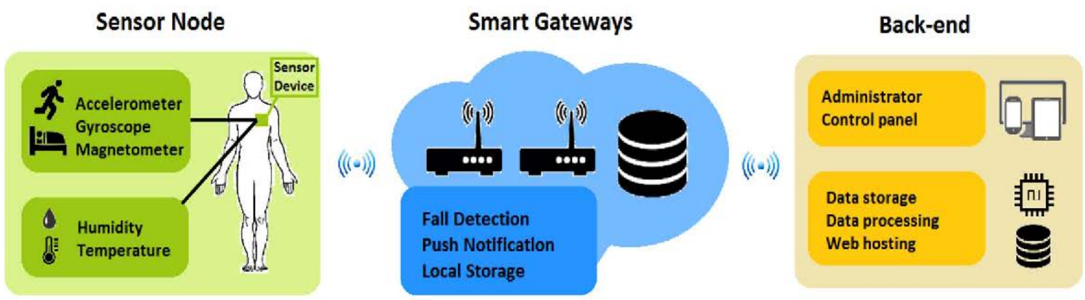


Figure 2.6 IoT based Fall Detection System [48]

Gia et al. [48] propose an energy efficient designing method of sensor nodes in an IoT FDS. A simple custom-built sensor node is designed to achieve high level of accuracy for detecting falls. The system architecture includes a sensor node, gateway with a fog layer unit and a backend system. A multi-level threshold check implemented on tri-axial accelerometer data calculates sum vector magnitude of features and detects events of fall. Yacchirema et al. [49] propose a decision tree based Big Data model running on a Smart IoT Gateway using a tri-axial accelerometer embedded in a 6LowPAN wearable device. If a fall is detected, the caregivers are duly notified by an alarm. The proposed smart system for fall detection provides service on cloud. High success rates for precision, accuracy and gain are observed in experimental evaluation. Ajerla et al. [50] propose a real-time patient monitoring framework for fall detection using wearable sensor devices from MbientLab, an open source streaming engine called Apache Flink for streaming data analytics, and a long short-term memory (LSTM) network model for fall classification. A fall detection accuracy of 95.8% is observed using 30 LSTM hidden units in each layer on the MobiAct dataset. Sprute et al. [51] propose an accelerometer based smart home solution for detecting falls called Smart Fall. An affordable, low energy requiring wearable device is customized along with a receiver component which acts as a mediator between the smart home environment and the wearable sensor device. Bluetooth Low Energy (BLE) protocol is used to establish wireless connection between the wearable device and the receiver. OpenHAB integration platform is used to connect home appliances in a platform independent way. Vector Sum of Acceleration data (VSA) from the accelerometer is utilized as a threshold for fall detection. Finally, a sensitivity of 91% and a specificity of 100% is obtained in performance evaluation.

Hsieh et al. [52] propose an action-used based system to detect falls using a deep neural network to calculate framework with feedback of features. A feedback model for optical flow assists the system. In addition, a feature feedback mechanism scheme (FFMS) is proposed to assign the feature of the convolutional layers with the most suitable object recognition models. A basic ML method with the convolutional layer to consider the optical feedback is generated yielding an efficient computing cost. An IoT based smart home architecture comprising of RGB cameras is used to collect human activity information. Luo et al. [53] design and implement an FDS called SenseFall using optical flow analysis to highlight changes in thermal energy of each sub-region as an important spatio-temporal feature. Pyroelectric infrared (PIR) sensors are installed to uncover the change of the thermal flux within the 3D object space cells. Hierarchal classifier like a two-layer HMM is used to distinguish human activities from falls. Kianoush et al. [54] propose a real time body localization and device-free RF-based fall detection system that continuously analyzes the RF signals generated by industry-compliant radio devices working in the 2.4 GHz ISM band and facilitating machine-to-machine communication functions. Obstructions caused in the RF signal propagation by multipath phenomenon and subject induced diffraction are utilized for body localization while an HMM is used for detecting falls. In addition, a sensor fusion toolcombining device-free RF-based sensing system within an industrial image sensors framework is proposed. The effectiveness of the approach is proven by the sensitivity and accuracy measurements in the performance evaluation.

Table 2.1 summarizes the advantages and disadvantages of fall detection methods discussed in sections 2.1 and 2.2.

Approach Used	**Strengths**	**Shortcomings**
Vision based fall detection	3D posture and scene analysis, inactivity monitoring, shape modeling,	Invasion of privacy, interference and noise in data, burdensome syncing of

	spatio-temporal motion analysis, occlusion sensitivity	devices, difficult setup of devices
Ambience based fall detection	Safeguards privacy, robust occlusion sensitivity	Expensive equipment, detection dependent on short proximity range
Wearable sensors based fall detection	low costs, small size, light weight, low power consumption, portability, ease of use, protection of privacy, robust occlusion	Intrusive approach, sensors to be worn at all times
Fusion based fall detection	Robust measurements, accurate detections, high performance	Difficult setup of equipment, complex syncing between devices
IoT based fall detection	High success rates for precision, accuracy and gain, accessibility with real-time patient monitoring	Threat of data security, compromise of privacy, strict global healthcare regulations

Table 2.1 Strengths and Shortcomings of different fall detection approaches

2.3 Classification Algorithms on Sensor Approaches

According to the literature review done in this research, there are two categories of fall detection approaches irrespective of the type of sensors used to accumulate the information data. The categories includefall detection relying on thresholds and use of ML algorithms for fall detection.

2.3.1 Threshold Based Fall Detection

Classification thresholds often called as decision thresholds are a common criterion used in categorizing events and setting labels. A general rule of thumb is to use predefined value of threshold(s) and classify events through comparison with real time sensor data. If the value of observed sensor data is higher than the set threshold for that sensor type, then the activity is classified as a “fall”. Many researches use a combination of thresholds and associations rules among them to reach a final decision. Mohammadi et al. [55] propose threshold-based classification rules to reach a threshold that minimizes classification errors and derive its asymptotic distribution.

The researches,[28] [31] [34] [38] [40] [43] [44] utilize threshold-base criteria for respective fall detection techniques. Thresholds are dependent on the method used, physical attributes of the subject e.g. the subject height since rules applied for short people can’t be same for taller people and the categories of falls considered (falling laterally,falling backward, falling forward) etc. Dai et al. [57] introduce a smartphone android based FDS called PerFallD which utilizes embedded smartphone sensors like accelerometer and gyroscope to detect falls and send alerts if subject doesn’t respond within a pre-specified time frame. Thresholds based on angular velocity and accelerationidentify falls. Different thresholds had to be set per location for experiments conducted at multiple sensor placements on the body.

One major drawback of using thresholds for fall detection is the inability to correctly distinguish higher accelerations generated during falls from similar accelerations patterns produced in activities likesitting down abruptly, standing upright abruptly, climbing up , hopping, jumping etc. Suchoverlap of patterns produces uncertainty in threshold-based approaches as described in [56]. Figure 2.7depicts acceleration data gathered from 10 subjects while performing ADLs and falls as explained in [56].

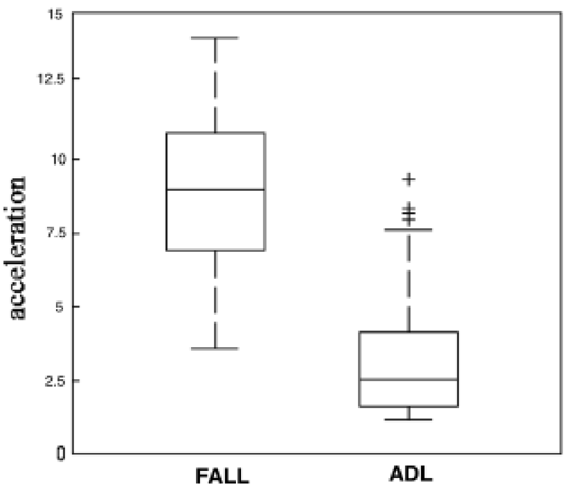


Figure 2.7 Fall vs ADL acceleration data

2.3.2 Machine Learning Based Fall Detection

Machine learning (ML) is a subset of Artificial Intelligence (AI) which incorporates the study of algorithms and statistical models to make systems learn automatically and executeparticular tasks without explicit instructions. Essentially a mathematical model is built by machine learning algorithms on sample data to execute decision making and make predictions in absence of explicit instructions. ML has a wide range of applications from data analysis to computer vision, email filtering, predictive business analytics etc. Healthcare systems have shown considerable advancements using ML algorithms at the heart of systems. A common model for fall detection using machine learning is depicted in Figure 2.8 and explained in detail below.

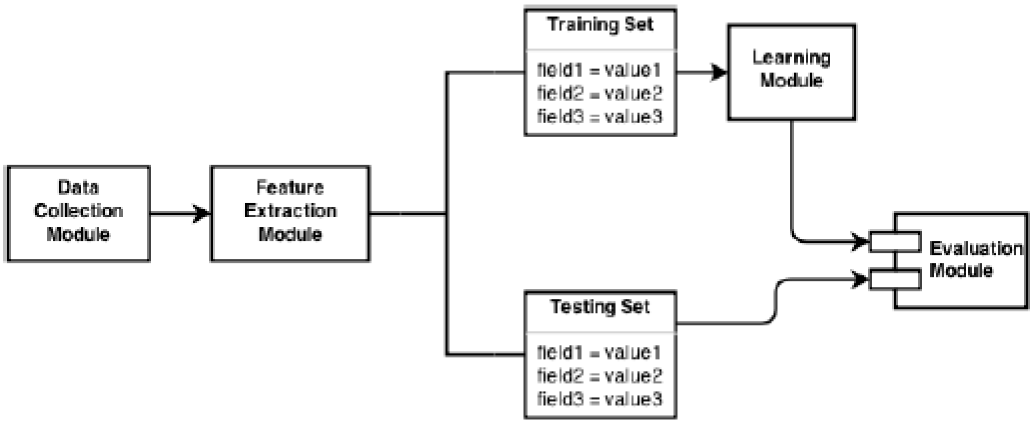


Figure 2.8 A basic ML model for fall detection [58]

- **Data Collection**

First and foremost, specific variables from the subject of interest's motion information is recorded in a systematic fashion ensuring validity, reliability and accuracy of the accumulated data. In the fall detection context, speed of a fall, acceleration coordinates, durations of inactivity, angular velocity, patterns of movements etc. are the commonly recorded variables. Noise and other meaningless redundancies are eliminated from the collected data to output valuable information that follows a specific format.

- **Feature Extraction**

Dimensionality reduction also called feature extraction is the procedure to identify relevant features and characteristics from the collected raw data. Feature extraction has a direct relation with dataset's descriptive power. Hence increase in number of features makes a dataset more expressive. Feature extraction is accomplished in two stages; feature construction and feature selection.

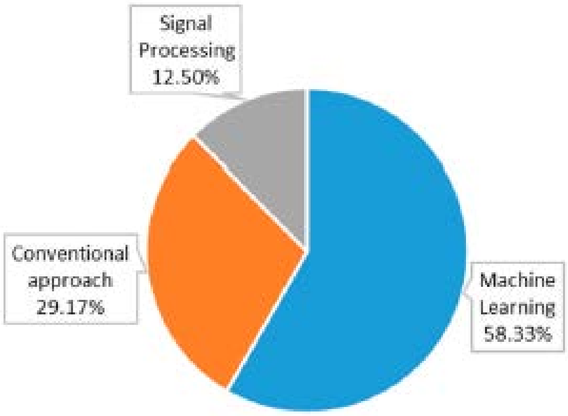
- **Learning Module**

  Learning mechanism focus on finding important and well-meaning relationships within data and analyzing the process to extract this information. The learning mechanisms are categorized as supervised and unsupervised, where the former utilizes labeled data to generate a predictive model and the later uses unlabeled data to create a recognition model. The most relevant supervised machine learning algorithms include Decision Trees (DTs), Naive Bayes, K-Nearest Neighbour (KNNs) and Support Vector Machines (SVMs).

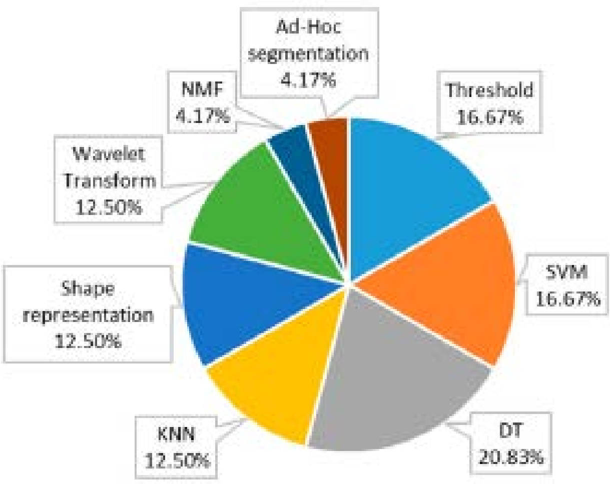
- **Model Evaluation**

  An essential module in the learning phase is the model evaluation that utilizes performance indicators to systematically assess the efficiency of the inferred structure and comparison of different learning techniques. Cross-validation is the most appropriate choice for measurement of performance in fall detection scenario. The comparison of the performance of a classifier on a specific dataset is done with the help of statistical tests along with cross-validation. An array known as confusion matrix stores the classification results.

  In ML based fall detection approach, various categories of falls and ADL patterns are trained by a learning algorithm and then events are classified categorically by an evaluation algorithm. Wang et al. [59] propose an unobtrusive approach named WiFall where physical layer Channel State Information (CSI) is used as the indicator of activities. WiFall delivers 90% precision with a false alarm rate of 15 percent on average using a one-class SVM classifier in all testing scenarios. Whereas when using Random Forest algorithm, 94% fall detection precisions with 13 percent false alarm are achieved. Nukala et al. [60] propose real-time automatic fall detection by applying a custom Wireless Gait Analysis Sensor (WGAS) with a with a simple but very fast Back Propagation Artificial Neural Network (BP ANN) and with an SVM classifier. BP ANN yields a high specificity of 97.4%, and 100% sensitivity, whereas SVM outputs specificity of 100%, and sensitivity of 94.1%. The SVM demonstrates somewhat reduced sensitivity in comparison to BP ANN. The reason is the use RBF Kernel as an activation function whereas the BP ANN activation is the tan sigmoid (Tanh) function producing better recognition accuracy. Özdemir et al. [61] propose fall detection using accelerometer, gyroscope and magnetometer at 6 different body locations. Several machine learning algorithms were used for detecting falls including the artificial neural networks (ANNs), k-nearest neighbor (k-NN) classifier,Bayesian decision making (BDM),support vector machines (SVM),least squares method (LSM) anddynamic time warping (DTW). Sliding window technique with a 5 second time duration is used to partition the data before the pre-processing step. The KNN classifier and LSM conclude the best results at above 99% sensitivity, accuracy and specificity. Stone et al. [62] propose a decision tree based two-stage fall detection system using Microsoft Kinect on data collected from homes of elderly for a period of nine years. The first stage of the proposed mechanism involvessegmentation of ground events from vertical state time series data determined by track analysis of the subjects, characterizing the subject’s vertical state in individual depth image frames. The second stage involves using an ensemble of decision trees to compute the confidence of a fall preceding a ground event. The method makes comparisons with five different fall detection algorithms and produces promising results. Machine learning based fall detection algorithms have become the new mainstream approach over conventional signal processing and threshold-based applications since the advancement of sensors to deliver detailed human activity recognition information. Figure 2.9 presents the categories of most cited algorithms (as mentioned in [63]) used for fall detection since year 2014.



(a)



(b)

Figure 2.9: Categories of algorithms employed for FDS: (a) major categories of algorithm employed since 2014, (b) detailed categories of algorithms employed since 2014

Table 2.2 summarizes fall detection methods proposed using machine learning algorithms.

Ref No.	Year of Publication	Mechanism Deployed	Algorithm	Accuracy (%)	Specificity (%)	Sensitivity (%)	Precision (%)
[60]	2014	Fall detectionusing a custom Wireless Gait Analysis Sensor	BP ANN SVM	NS	97.4 100	100 94.1	NS
[61]	2014	Fall detectionusing accelerometer, gyroscope and magnetometer at 6 different body locations	KNN LSM SVM BDM DTW ANN	99	99	99	NS
[96]	2015	Fall detectionusing accelerometer	TBM + KNN	90.0	97.0	83.0	NS
[59]	2017	Fall detection using physical layer Channel State Information as the indicator of activities	SVM RF	NS	NS	NS	90 94

[93]	2018	Fall detection using accelerometer, gyroscope and magnetometer	KNN SVM LDA DT LR	99.0 97.4 96.4 95.9 97.4	97.9 97.0 93.8 93.7 96.9	100 97.9 99.0 98.0 97.9	NS
[94]	2018	Fall detection using accelerometer in an embedded system	DT	91.7	97.2	91.7	NS
[95]	2019	Fall detection using accelerometer	SVM	99.9	99.44	99.5	NS

Table 2.2 Fall detection techniques using ML algorithms

2.4 Deep Learning based Fall Detection

Recent years have seen an advance in application of Deep Learning techniques for natural language processing, image recognition, speech recognition, human activity recognition etc. DL techniques have demonstrated to deliver remarkable results in comparison to extracting features manually.

Nogas et al. [64] treatthe predicament of distinguishing between ADLs and falls as an anomaly detection problem. The approach is based on rare occurrences of falls and utilize convolutional LSTM autoencoder (ConvLSTM-AE) tolearn spatio-temporal features from activities of daily life (ADLs). The proposed method uses ADLs for training of ConvLSTM-AE. During testing, the unseen falls are identified by the reconstruction error. A frame-based anomaly score called the cross-context anomaly score is calculatedwith fall as the class of interest tocompute area under the curve (AUC) of the ROC. During testing, the anomaly score is then utilized to detect an unseen fall. The performance evaluation depicts that Deep Autoencoder was outperformed by Convolutional Autoencoder (CAE), and ConvLSTM-AE outperformed CAE, indicating that incorporating spatial, as well as temporal information prove effective in detecting unseen falls.

Musci et al. [65] prsent an online FDS based on the publicly available SisFall dataset using Recurrent Neural Network (RNN) as a classifier. Raw data is preprocessed by a fully connected layer succeeded by stacked LSTM layers and a fully connected layer at the end. Additional batch normalization and dropout layers configured with a weighted cross entropy loss function are inserted in the neural network. The model attains 97.16% and 94.14% accuracy for falls and ADLs respectively.Mauldin et al. [66] propose an Android app called SmartFall which utilizes accelerometer data gathered from a smartwatch to identify incidents of fall. In order to avoid the latency experienced while securing data privacy and interfacing with cloud server, SmartFall application running on a paired smartphone carries out the computation. Performance evaluations carried out in comparison to SVM and Naïve Bayes depicts outperformance of the deep learning models based on CNN and RNN due to the ability of the DL model to spontaneously learn subtle features from the unprocessed data. On the other hand, SVM and NB models are limited to learn only from a manually specified range of features. Wisesa et al. [67] propose to use RNN as a tool for analyzing falls from collected sensor data. The research uses the publicly available UMA FALL ADL dataset from Universidad de Málaga. LSTMs are used as a variant of RNN for the purpose of experiment . The results deduce a good classification between falls and ADLs by only using X-axis data of the accelerometer sensor attached to the waists of the subjects. The average accuracies from X-axis accelerometer data for training and validation are 91.43% and 92.31% respectively.

Table 2.3 summarizes fall detection techniques using deep learning methods.

Ref No.	**Year of Publication**	**Mechanism Deployed**	**Algorithm**	**ROC**	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
[99]	2017	Vision based fall detection and alert system	3D-CNN 3D-CNN with data augmentation	NS	78.7 96.9	NS	NS
[64]	2018	Fall detection treated as an anomaly detection problem	DAE CAE Deconv. CAE Upsampling ConvLSTM-AE	0.64 0.70 0.75 0.83	NS	NS	NS
[97]	2018	Fall detection based on SisFall dataset using accelerometer	RNN (LSTM)	NS	95.51	92.7	94.1
[65]	2018	Fall detection based on SisFall dataset	RNN	NS	97.16 (falls) 94.14 (ADLs)	NS	NS

[67]	2019	Fall detection based on UMA fall dataset using only X-axis data of the accelerometer sensor	RNN	NS	92.31	NS	NS
[98]	2019	Fall detection based on SisFall dataset	One LSTM layer Two LSTM layers One GRU layer Two GRU layers	NS	96.3 96.1 96.4 96.7	88.2 90.2 88.2 87.5	96.4 97.1 96.3 96.8

Table 2.3 Fall detection methods using deep learning techniques

2.5 Treatment of Missing Values

Missing value arise when data collection doesn't provide data as intended due tofaulty measurements, manual data entry and equipment failures etc. It is common to observe missing values in most of the sensory sources used. Invariably, missing values may emerge as outliers or wrong entries which need to be deleted before intended analysis and are difficult to process. Luengo et al. [74] highlight the key issues that are associated with the missing data problem including; loss in efficiency, complications in data analysis and biased results produced from differences between complete and incomplete datasets. The underlying effects of missing values observed in data makes their treatment an imperative task.

Treatment of missing data is a mature field with considerable amount of research present in the scientific community. Donald B. Rubin [75] describes the phenomena of missing values and provides statistical literature on the mechanisms that cause missingness when making inferences about the data distribution. The simplest treatment of missing values is the reduction of dataset by eliminating missing values where observed. Kantardzic et al. [76] propose the elimination of samples (rows) whereas Lakshminarayan et al. [77] propose elimination of attributes (columns) where missing values are observed. Elimination of all samples is called case analysis. This treatment is conditioned on availability of a large dataset and a relatively small percentage of missingness in the samples. Removing attributes with missing entries during analysis is not appropriate treatment because it creates making inferences about the attributes difficult.

Imputation is a method utilized for replacement of missing values. It usually includes basic techniques like replacement of missing value with the last observed value, replacement with mean or medians and single regression replacement. Most recent values are replicated in the case of last value replacement when missing values resurface. A similar version of value replacement is the hot deck imputation which uses details from similar observations to impute missing data. Andridge et al. [78] review the hot deck imputation methods for non-responses in surveys and highlights key issues that are observed like use of covariate information when dealing with adjustment cell method limiting the amount of auxiliary information that can be effectively utilized. Another important issue is obtaining reliable inferences from imputed data achieved through hot-deck. Mean imputation is commonly used for numeric attributes where missing values are imputed with the mean of attributes. For categorical attributes, the most common attribute is used for replacement. Variations of mean imputation method include imputation of missing values with the mean of a given class, imputation of missing values with the median of a given class for a more robust approach since the presence of outliers affect the imputation with means. Kantardzic et al. [76] apply various methods of mean imputation in their research. An advanced method of singular imputation is the use of regression analysis to use relevant information variables in order to predict the values of the missing responses. Missing values are measured only once in single regression technique for imputation which can lead to biasing of standard errors since there is no method to check the accuracy of imputed values resembling true values.

An extension of single regression imputation is the multiple imputation technique where missing values are estimated multiple times to produce one final result. There are three basic steps to multiple imputation technique: (1) replacement of missing data, (2) independent statistical analysis of each resulting independent dataset, (3) combining the results of imputations. Schaffer et al. [78] review the core ideas of multiple imputation elaboratively discussing the software tools for data analysis and treatment. The proposed method is tested on a benchmark dataset from the Adolescent Alcohol Prevention Trial (Hansen & Graham, 199 I). Grzymala et al. [79] propose the closest fit algorithm for imputation of missing data by replacing a absent attribute value with an available value of the same attribute from another similar case. The KNN method looks most similar to closest fit method but the KNN method looks for "k" no of most relevant attributes. This method is used for classification, where"k" closest data set points to an input vector are examined and assigned the object to the class that has the majority of points among these "k". Batista et al. [80] use the K Nearest Neighbor algorithm for imputation. Several distance measures to compute the nearest neighbors are analyzed such as Manhattan, Euclidean and Pearson functions.

The research concludes that the ideal value of k is significant to guarantee high performance.

Neural Networks comprise of a class of predictive modeling system which operates by adjusting parameters iteratively. This technique has lately been utilized as a predictive model for imputing missing values. Gupta et al. [81] propose using multilayered networks and backpropagation algorithm for imputing missing values by dividing the dataset into complete and incomplete subsets. Training of the neural network is done on the complete dataset and the trained network is used to calculate missing values of the incomplete set.

CHAPTER: 3

METHODOLOGY

This section gives a detailed description of the methods applied for the purpose of this research study in the order of application: dataset information, proposed solution and experimental design setup. The resulting outcomes of the proposed study are analyzed in the next section.

3.1 Dataset Information

The promising growth in the field of research for fall detection and HAR using ML approaches make the selection of an appropriate dataset for the training and validation of the model a crucial key task.

The few publicly available datasets deem insufficient to be used for testing fresh approaches due to lack of detailed information, limitations in activities performed and absence of targeted objective population. Another drawback observed in most publicly available datasets is the method of recording human activities. In most cases, inbuilt smartphone sensors are used to record motion. An underlying drawback of using smartphones for detecting falls could be the loose connection between the device and the subject's body depending upon the placement of the smartphone. This could lead to the invalidity of produced results.

This research utilizes wearable sensor-based approach for detection of falls since approaches based on vision and ambience have restricted scope and feasibility for some environments. Wearable sensor technology is unarguably the most employed method for reliable fall detection. An alternative to external sensors, wearable sensors are mounted on the subject's body. Wearable sensors collecting important data related to the patient's body movement make up for an efficient solution for fall detection with their light weight, low costs, small size, low power utilization, portability, user-friendliness and attractive features. Commonly used wearable sensors are gyroscope, accelerometer, heart rate sensor, magnetometer etc. which can be attached at multiple positions on the human body.

The research uses few basic requirements for the analysis of wearable sensor datasets: public availability of raw data, detailed documentation of performed activities, incorporation of both ADLs and falls in the activities performed and the dataset to be reported in a peer-reviewed paper. The datasets mentioned in the Table 3.1 meet the mentioned premises. Among the mentioned datasets, SisFall dataset is chosen as the most appropriate choice of dataset for the purpose of this research since it contains the largest amount of data and heterogeneity in ADLs and subjects. The other datasets exclude the elderly population and have limited diversity in context of performed activities and number of subjects.

The SisFall dataset is unique in its extensive research and focus on independent elderly people who are considered as the objective population in fall detection systems. A survey washeldfor 15 adults of 60 years of age and abovefor the psycho-physic program of the Universidad de Antioquia (for the duration of July till August 2014, inColombia), and 17 retirement homes (for the duration of October 2014 till January 2015, in Colombia). It serves as a helping aid for the selection of suitable activities to be performed by the subjects.

Dataset	No. of Subjects	No. of type of ADLs performed	No. of type of Falls performed	Sensing Device
MobiFall [68]	24 (22 to 42 years old)	9	4	Smartphone
tFall [69]	10 (20 to 42 years old)	7 days of continuous ADL readings. The experiments were not labeled by activity.	8	Smartphone
DLR [70]	16 (23 to 50 years old)	6	1	Wearable sensors
Project gravity [71]	3 (ages 22, 26, and 32)	7	12	Smartphone
UMAfall [72]	17 (18 to 55 years old)	8	3	Wearable sensors
SisFall [73]	23 (19 to 75 years old)	19	15	Wearable sensors

Table 3.1 List of publicly available datasets considered for fall detection solution

The dataset was generated with the collaborative volunteering of 38 subjects categorized as young adults and elderly subjects. The elderly class included 15 volunteers out of which 8 were male and 7 were female subjects whereas the younger setcomprised of 23 subjects including 11 males and 12 females. Table 3.2 depicts the basic information about each group whereas detailed facts for every individual is presented in Table 3.3

	Age	Gender	Weight (in kg)	Height (in m)
Young Subjects	19 - 30	M	59 - 82	1.65 - 1.84
	19 - 30	F	41 - 64	1.50 - 1.69
Senior Subjects	60 - 71	M	56 - 103	1.63 - 1.71
	62 - 75	F	50 - 71	1.49 - 1.69

Table 3.2 Age, height, weight of the participating subjects

Subject	Age	Gender	Weight (in kg)	Height (in m)
E01	71	M	102	171
E02	75	F	57	150
E03	62	F	51	150
E04	63	F	59	160
E05	63	M	72	165
E06	60	M	79	163
E07	65	M	76	168
E08	68	F	72	163
E09	66	M	65	167
E10	64	F	66	156
E11	66	F	63	169
E12	69	M	56	164
E13	65	M	72	171
E14	67	M	58	163
E15	64	F	50	150
A01	26	F	53	165
A02	23	M	58	176
A03	19	F	48	156
A04	23	M	73	170
A05	22	M	69	172
A06	21	M	58	169
A07	21	F	63	156
A08	21	F	42	149
A09	24	M	64	165
A10	21	M	67	177
A11	19	M	81	170
A12	25	F	47	153
A13	22	F	55	157
A14	27	F	46	160
A15	25	F	52	160
A16	20	F	61	169
A17	23	M	75	182
A18	23	M	73	181
A19	30	M	76	170
A20	30	F	42	150
A21	30	M	68	183
A22	19	F	51	158
A23	24	F	48	156

Table 3.3 Specifics of individual participants in the SisFall dataset

A custom made sensing deviceembedded with a Freescale MMA8451Q accelerometer,an Analog Devices (Norwood, Massachusetts, USA) ADXL345 accelerometer,a Kinets MKL25Z128VLK4 microcontroller (NPX, Austin, Texas, USA), an ITG3200 gyroscope, an SD card for recording, and a 1000 mA/h generic battery is used to accumulate data recordings. The custom sensor was mounted to the waist of the participants with the help of a belt as shown in Figure 3.1. This position assists in precise differentiation among activities when using accelerometer. The sensor’s orientation (see Figure 3.1)is characterized as the right side of the subject aligned with positive x-axis, the direction of gravity aligned with the positive y-axis and forward direction aligned with the positive z-axis. All trials were conductedusing a200 Hzsampling frequency.

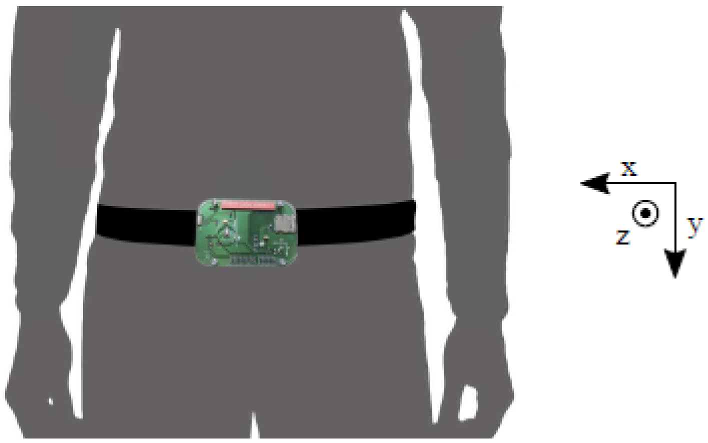


Figure 3.1 Waist mounted custom made embedded device used for data acquisition comprising of a gyroscope and two accelerometers

All the protocols followed for performing falls and ADLs have been approved by medical professionals and physicians specialized in fields of athletics. The Supplementary Materials contain videos for each category of activities performed by the subjects, depicting the precisesettings during recording. Snapshots taken from two of these recordings are shown in Figure 3.2 and Figure 3.3 in order to depict a fall and ADL, respectively.

Figure 3.2 Fall F01: Forward fall when walking triggered by a slip performed by a subject

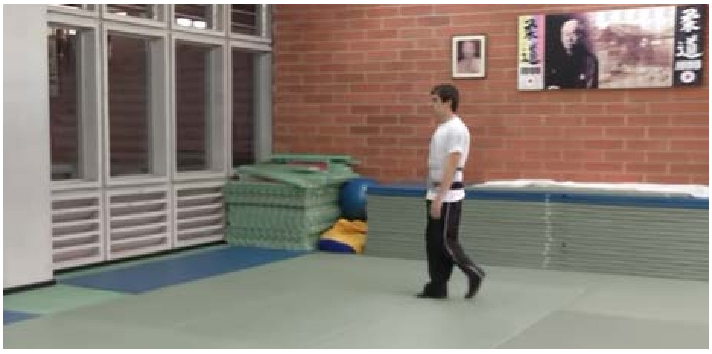


Figure 3.3 ADL D01: Walking slowly performed by a subject

The falls included in the dataset are mentioned in Table 3.4 along with description, duration and number of trials performed.

Activity Description	**Activity Code**	**Trial Period**	**Trials**
Falling forward when walking triggered by a slip	F01	15s	5
Falling backwards when walking triggered by a slip	F02	15s	5
Falling laterallywhen walking triggered by a slip	F03	15s	5
Falling forward when walking triggered by a trip	F04	15s	5
Falling forward when jogging triggered by a trip	F05	15s	5
Falling Verticallywhen walking caused by fainting	F06	15s	5
Falling when walking, with use of hands in a table to dampen fall, caused by fainting	F07	15s	5
Falling forward while trying to get up	F08	15s	5
Falling laterally while trying to get up	F09	15s	5
Falling forward whilesitting down	F10	15s	5
Falling backwards whilesitting down	F11	15s	5

Falling laterally whilesitting down	F12	15s	5
Falling forward when sitting, triggered by fainting or falling asleep	F13	15s	5
Falling backwards when sitting, triggered by fainting or falling asleep	F14	15s	5
Falling laterally when sitting, triggered by fainting or falling asleep	F15	15s	5

Table 3.4 Falls performed in the dataset

The selection of activities chosen for ADLs are based on ordinary movements, movements resembling falls in terms of acceleration waveforms and movements generating false positives due to high acceleration. The ADLs performed in the SisFall dataset are mentioned in Table 3.5.

Activity Description	Activity Code	Trial Period	Trials
Walking slowly	D01	100s	1
Walking quickly	D02	100s	1
Jogging slowly	D03	100s	1
Jogging quickly	D04	100s	1
Walking upstairs and downstairs slowly	D05	25s	5
Walking upstairs and downstairs quickly	D06	25s	5
Slowly sit in a half height chair, wait a moment, and up slowly	D07	12s	5
Quickly sit in a half height chair, wait a moment, and up quickly	D08	12s	5
Slowly sit in a low height chair, wait a moment, and up slowly	D09	12s	5
Quickly sit in a low height chair, wait a moment, and up quickly	D10	12s	5
Sitting a moment, trying to get up, and collapse into a chair	D11	12s	5
Sitting a moment, lying slowly, wait a moment, and sit again	D12	12s	5
Sitting a moment, lying quickly, wait a moment, and sit again	D13	12s	5
Being on one’s back change to lateral position, wait a moment, and change to one’s back	D14	12s	5
Standing, slowly bending at knees, and getting up	D15	12s	5
Standing, slowly bending without bending knees, and getting up	D16	12s	5
Standing, get into a car, remain seated and get out of the car	D17	25s	5
Stumble while walking	D18	12s	5
Gently jump without falling (trying to reach a high object)	D19	12s	5

Table 3.5 ADLs performed in the dataset

3.2 Proposed Solution

The focus of this research is to propose a noise tolerant fall detection that detects the occurring falls with precision and accuracy while handling the missing values observed in wearable sensor data. The adopted method is based on Machine Learning and, in particular, on a Deep Learning approach. The proposed fall detection solution attempts to achieve the already challenging task of human activity recognition along with the added obstacles of dealing with noisy, incomplete data.

The overall system design is visually represented in Figure 3.4, the details of which are explained stepwise.

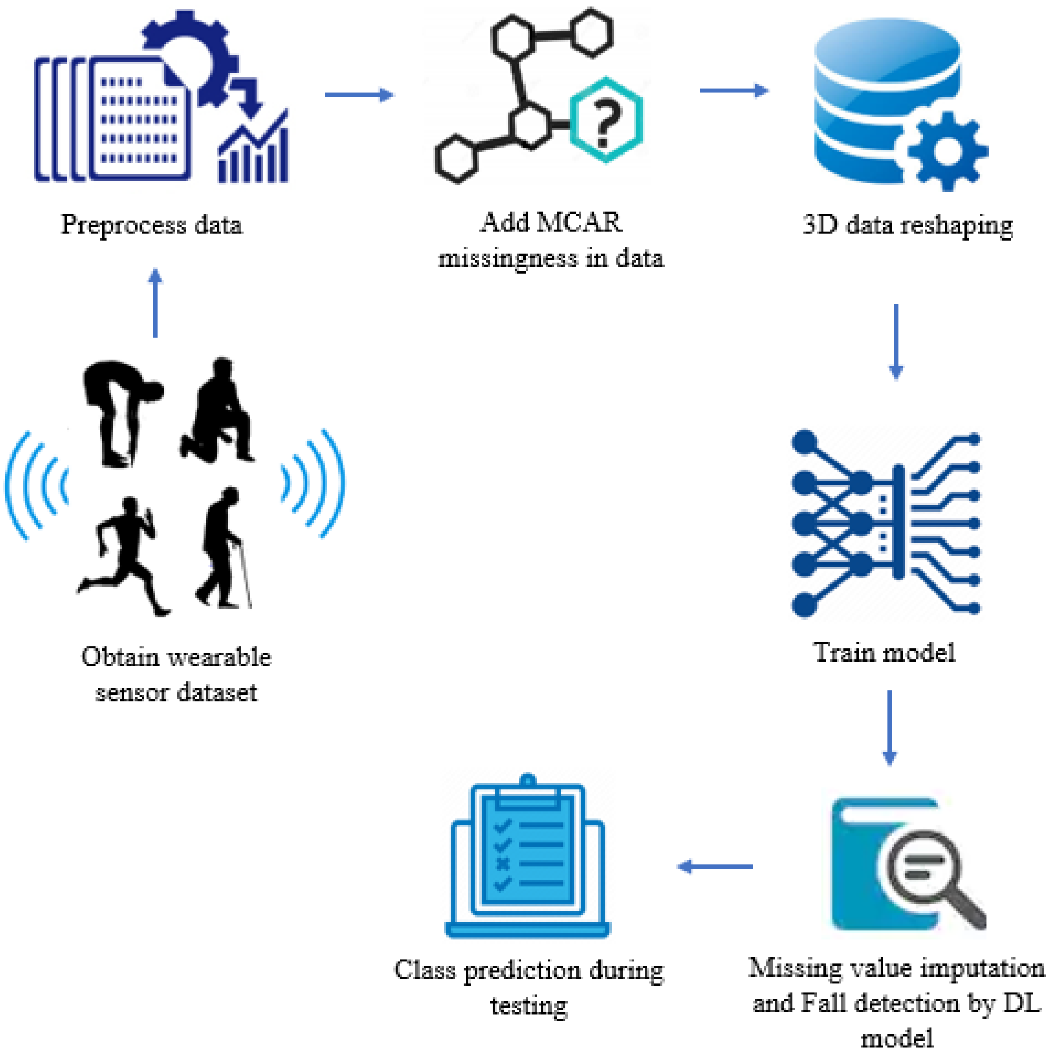


Figure 3.4 Overall System Design

3.2.1 Dataset

As mentioned previously, the dataset obtained for this research is SisFall [73] dataset which comprises of 23 participants performing15 categories of falls and 19 different ADLs.The elderly group had been exempted from performing falls for safety and medical reasons. The research uses a subset of the original dataset, focusing on the activities performed by 5 young adults, particularly with codes: SA01, SA02, SA03, SA04, SA05. All these subjects performed 19 different ADLs and 15 categories of falls carried out over multiple trials. Each participant has 154 csv files, each file corresponding to a typical activity and a particular trial. The total number of data files for five subjects sum up to (5 x 154) 770. The exclusion of the elderly subjects from selection was merely based on their inability to perform falls. If included, the proposed fall detection solution would give biased results based on imbalance of activities (ADLs vs falls) in the data.

3.2.2 Preprocessing

Data preprocessing involves the transformation applied to collected raw data before feeding it to machine learning algorithms. Data preprocessing is a key step that reshapes data into desired clean formats that can be feasible for analysis later.

The SisFall dataset contains inequal number of performed ADLs and falls along with variable duration and number of trials per activity. This nonuniformity in the data can lead to biased learning during the training and validation phases resulting in inaccurate picture for identifying falls. In order to uniformize the collected data generated from each source, durations of execution for all the activities are analyzed and the minimum duration for an activity i.e. (12 seconds) is chosen as the standard. As a result, all activity durations are truncated to 12 seconds leading to 2400 records per file. The first 12 seconds of performance of each activity are chosen. The sampling frequency is 200 Hz. This preprocessing step rectifies the imbalance in data and produces a uniform subset of the original dataset.

The next task carried out for the preprocessing of data is data annotation. Data annotation is basically labelling of data to meet the needs of machine learning algorithms. The annotations or preparations involved can be quite varied, but usually it is to replicate whatever task the model is likely to perform. The critical task of data annotation is one of the most reliable ways to increase model performance. The objective of data annotation is to create clean data which when fed to a machine learning model in training phase allows the algorithm to rightfully identify the data and learn from it. The original SisFall dataset files contain tri-axial data from 3 sensors. The annotated version of the dataset created for the selected subjects includes the addition of activity and user ID labels in order to facilitate data analysis later.

The classes of activities created during data labelling include:

- **FALL:** this class characterizes the activity intervals when the subjectsuffers a dangerous state transition leading to a harmfulshift of state, i.e., a fall. All 15 types of falls performed by the participants are subsumed under the umbrella of this class label.
- **ADL:** this class characterizes the activity intervals when the subjectmaintains control of its state and performs tasks without abrupt state transitions which may lead to falls. All 19 types of ADLs performed by the participants are subsumed under the umbrella of this class label.

3.2.3 Missingness

Missingness in data is defined as an absence of response from sensors or data collection sources where a response is expected. Numerous reasons can be the cause of underlying missingness, all of which lead to detrimental effects on quality and authenticity of data. The impact of missing data can be significant on quantitative research, resulting in the following concerns:

- Skewed parameter estimation
- Loss of information
- Reduced statistical insight
- Escalated standard errors
- Impaired generalizability of findings

- **Proportion of Missing Data**

  The percentage of missingness in a dataset is associated with the standard of statistical inference interpreted from data. A benchmark for acceptable proportion of missingness in data as such does not exist in the literature, however, Schaffer et al. [82] assert that a missingness percentage of 5% or less is insignificant. Contrarily, Bennett et al. [83] concludes that a skewed statistical analysis and biased inferenceare expected when the percentage of missing values in data exceed10% . Additional criteria other than the percentage of missing data asses the missingness issue. Tabachnick et al. [84] deduce that the patterns of existence of missing data and their respective mechanisms have greater impact on the findings than the percentage of missingness.

- **Missing Data Mechanisms**

For understanding the concept of missing data mechanisms, the data matrix X is partitioned into incomplete subset with missing values as ($X_{missing}$) and observed subset with complete responses as ($X_{observed}$). Hence the dataset can be represented as:

$$\mathbf{X} = (\mathbf{X}_{\mathbf{missing}}, \mathbf{X}_{\mathbf{observed}}) \quad (3.1)$$

Rubin [75] analyzed the behaviors of missingness in data and categorized each phenomenon mathematically. According to him, missing values exist in the following three mechanisms: Missing at Random (MAR), Missing Completely at Random (MCAR), Missing Not at Random (MNAR)

- **Missing at Random (MAR)**

Missing at random (MAR) is the most realistic assumption for research where data is considered to be missing when the probability of missing values relies on the set of observed responses but is not dependent on the specific missing values themselves. Hence, the MAR conditions persists when the probability of observing missing data relies only upon $X_{observed}$, but not on $X_{missing}$.

MARmissingness can formally be defined with the following assumption: Let M be the matrix of missingness with the same dimensions as X. Matrix M consists of 1s and 0s only, with 1 corresponding to a value being observed and 0 to a value being missed.

Let the distribution of M be given as P (M|Y,ξ), where

ξ = parameter of missingness

then the distribution of M can be modeled as Equation 3.2 and the missingnessmechanism is considered MAR according to Schaffer [85].

$$\mathrm{P}\ (\mathrm{M}|\mathrm{X},\xi) = \mathrm{P}\ (\mathrm{M}|\mathrm{X}_{\mathrm{observed}}\ ,\mathrm{X}_{\mathrm{missing}}\ ,\xi) = \mathrm{P}\ (\mathrm{M}|\mathrm{X}_{\mathrm{observed}}\ ,\xi) \qquad (3.2)$$

Hence, the probability of observing missing values in data is dependent upon the observed values and the parameter of missingness (ξ).

- **Missing Completely at Random (MCAR)**

The MCAR mechanism of missingness in data is observed when the probability of existence of missing values in data is neither dependent upon the observed responses ($X_{observed}$) nor on the missing values that are expected to be obtained ($X_{missing}$). It is an ideal but unreasonable assumption which exists in cases like failure of equipment, technical unsatisfaction, loss of data in transferring etc. The distribution of M can be modeled as:

$$\mathbf{P}\ (\mathbf{M}|\mathbf{X},\boldsymbol{\xi}) = \mathbf{P}\ (\mathbf{M}|\mathbf{X}_{\mathbf{observed}},\mathbf{X}_{\mathbf{missing}}\ ,\boldsymbol{\xi}) = \mathbf{P}\ (\mathbf{M}|\boldsymbol{\xi}) \qquad (3.3)$$

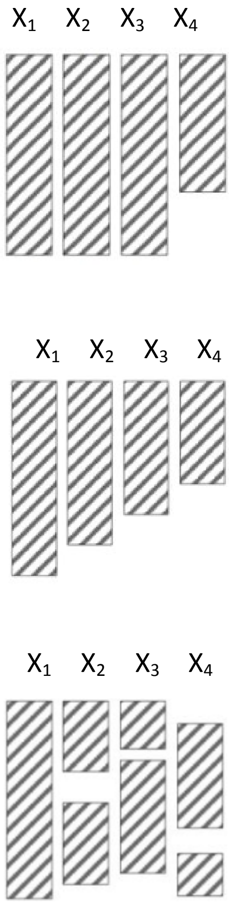
- **Missing Not at Random (MNAR)**

MNAR missingness is observed when the probability of observing missingness in data depend upon the missing values themselves. If the characteristics of data does not satisfy the criteria of MAR/MCAR categories, then data falls into the category of MNAR. For MNAR, the model of missingness needs to be clearly described and integrated in data analysis. This practice helps generate unbiased parameter estimation. This exhausting requirement isn't neededwith MCAR or MAR mechanisms. The cases of MNAR data are challenging. Manual modelling the missingness is the only way to achieve unbiased parameter estimation for MNAR.

- **Patterns of Missingness**

Patterns of missingness in data are categorized as univariate/multivariate, monotone and arbitrary or general. In order to understand each pattern, suppose there are *n* rows and *p* columns in the data matrix X. Columns of the matrix are denoted as, $X_1, X_2, \ldots, X_P$. Each column represents one variable. The details of patterns are given below:

- multivariate if more than one variable is missing.
- Monotone: A missingness pattern is considered monotone when X_{j+1},X_{j+2},...X_p are observed as missing ifthe value at X_j is missed.
- Univariate/Multivariate: A missingness pattern is considered univariate if only one variable is missing.
- Arbitrary/General: When data is missing in a random haphazard manner for any attribute or feature then the mechanism of missingness is defined as arbitrary. This non-specific missingness mechanism is more difficult to cater for than the rest.



(a) Univariate Missing Pattern

(b) Monotone Missing Pattern

c) Arbitrary Missing Pattern

Figure 3.5 Missing data patterns: (a) Univariate missingness (b) Monotone missingness (c) Arbitrary missingness

For the purpose of this research, Multivariate Missing Completely at Random (MCAR) missing data mechanism was considered with various percentages of missingness in the data. Experimental analysis is carried out by adding three different percentages; 20%, 30% and 40% of MCAR missingness in data.

The resulting performances of the proposed fall detection mechanism with different percentages of missingness are compared with the fall detection results obtained by using complete clean data with no noise and missing values.

3.2.4 Data Reshaping

Data reshaping serves as a prerequisite task before feeding the data into the proposed fall detection design. Stacked Bidirectional LSTMs are used at the heart of proposed fall detection scheme which require 3D reshaping of data at the input. The three dimensions of input to each LSTM layer include samples, timesteps and features. A sample represents a single sequence. A batch comprises of one or more samples. A feature comprises of a single observation at a time step.One timestep is a single instant of observation in the sample. Hence, a three dimensional array is required at the input layer to fit the model and make predictions.

The number of timesteps and features need to be defined as the input dimension argument when specifying the LSTM's input layer. The network assumes data to have at least a single sample or more. A basic representation of input shape to LSTM layer is given in Figure 3.6.

A fixed sample size of 770 sequences is considered for all experiments in the research with variable timesteps for each case.

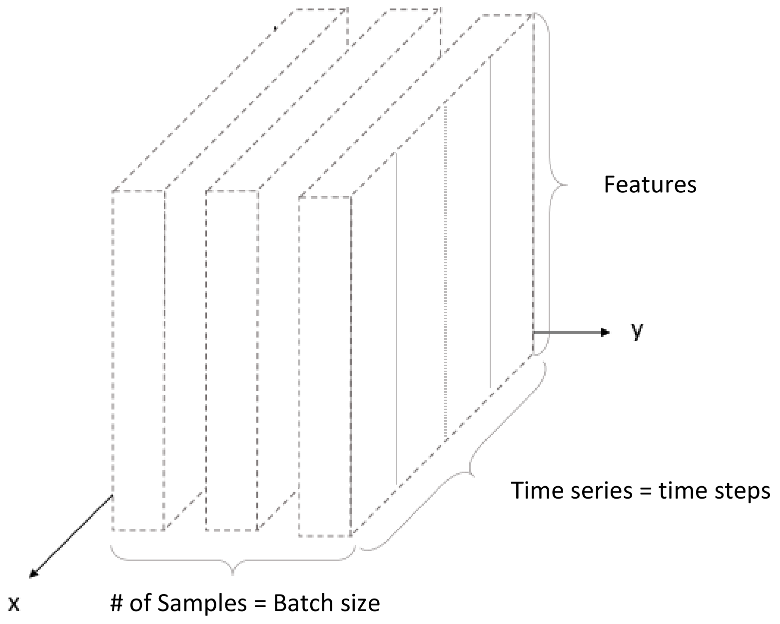


Figure 3.6 3D Input to LSTM layer

A 2D array can only be reshaped into 3D if the following condition for dimension equality holds true:

$$\text{Rows x Columns of 2D array} = \text{Samples x Time steps x Features of 3D array} \quad (3.4)$$

The research uses two different approaches for detecting falls. A combined sensor approach which utilizes data from all types of sensors used for data accumulation to help make decisions about fall detection and a single sensor approach which employs data from individual sensors (accelerometer or gyroscope) to detect falls. The number of features selected for data reshaping are dependent upon the type of approach used. The multisensor fusion approach uses two tri-axial accelerometers and one tri-axial gyroscope making the number of selected features to be 9. The single sensor approach uses one tri-axial sensor at one time making the number of selected features to be 3.

3.2.5 Fall Detection

The focus of this research is to create a deep learning based noise tolerant fall detection mechanism which handles missing values observed in data. The task of detecting falls is handledlike a sequence classification problem with a specifiedsequence length as input to the bidirectional LSTMs.

- **Sequence Classification Problem**

  In predictive modeling, sequence classification is expressed as predicting a class for the sequence of inputs available over space or time. Variable length of sequences, a large vocabulary of input characters and the requirement to learn long-term context or dependencies between symbols in the input sequence makes the sequence classification task challenging to achieve.

  The task of fall detection presented in this research requires the classification of each input sample(sequence) received into categories of either FALL or ADL.

- **Imputation of Missing Values**

  In statistics, imputation is defined as the replacement of unobserved values with substituted values. Imputation of missing values in data is a key task required to fulfill the need of data completeness for the uses of advanced analysis. Conventional methods such as case deletion, regression imputation which replaces missing values with mean, median or mode of data are simply not good enough to handle missing values as these methods can cause bias in the data. A wholesome imputation technique must satisfy a few rules: Estimation without bias, moderate computational complexity and time costs and the ability to retain the original data distribution and relationship among attributes.

Based on their ability to remember long term dependencies from the past and future to make predictions, Bidirectional Long Short-Term Memory RNNs can be utilized to achieve the task of imputation. Here, imputation can be considered as a mapping of input x_t to the missing value observed at x_{t+1}. A dataset with missing values is filled if missing values at time t+1 are replaced with their corresponding predictions.

- **Sensor Based Approaches**

There are two approaches used in this research based on the number and kind of sensors used to accumulate the data. A combinational approach which uses the combination of sensors to gather data coming from different sources. This approach uses data from all three sensors of the customized embedded device to detect falls. While the single sensor approach chooses one sensor at a time to detect falls. This is done in order to understand the behavior of individual sensor data and how it contributes towards fall detection. The sensors considered for this approach are the ITG3200 gyroscope and the MMA8451Q accelerometer. All three percentages of missingness have been considered for the individual sensor case.

A comprehensive overview of Recurrent Neural Networks (RNNs) is required to understand the proposed fall detection mechanism.

- **Recurrent Neural Networks**

RNN is a class of artificial neural network derived from the feed forward networks. However, the distinguishing feature between recurrent and feedforward networks is presence of at least one feedback connection in the recurrent networks. This connection feeds a part of the produced output back to the input. Thus, the activations passaround in a loop enabling the network to learn sequences and perform temporal processing. Hidden state is the most noteworthy feature of RNN which retains a portion of the sequential information. The sequential information is preserved in the hidden state which serves as a memory, remembering all the information that the network has witnessed so far from the preceding timestep. The same function is applied to all the inputs (operating with the same set of parameters) and hidden layers to generate the output. Thisreduces the complexity of parameters, unlike other neural networks.Mathematically, the process of carrying the memory forward can be represented as:

$$\mathbf{h_t} = \boldsymbol{\phi}(\mathbf{W x_t} + \mathbf{U h_{t-1}}) \tag{3.5}$$

Where h_t, the hidden state at timestep t, is a function of the input x_t at timestep t transformed by a weight matrix W. This modified input is appended to the hidden state of the preceding timestep h_{t-1} which is multiplied by the transition matrix U. A logistic sigmoid function or tanh is used as a squashing function, ϕ. The weight matrices act as filters deciding thesignificanceassigned to the current and previous hidden states. The feedback loop very exists at every timestep and adds traces of previous hidden state as well as those that preceded h_{t-1}if memory allows. Basic RNN architecture is presented in Figure 3.7.

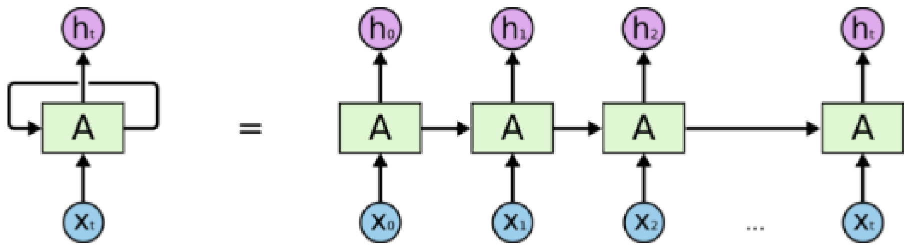


Figure 3.7 An uncoiledRNN [89]

The output given as y_t is linked to the hidden state and inputs after multiplication with a weight matrix V through a non-linear function e.g. softmax function.

$$y_t = softmax\ (h_t V) \tag{3.6}$$

The RNNs do not suffer from the limitation of accepting fixed dimensions for inputs and outputs and are flexible to variable sized inputs and corresponding outputs. The network can accept sequences either for the input and output, or both. The output can be configured as many-to-one or many-to-many. Some basic topologies for the RNN are shown in Figure 3.8.

Backpropagation in feedforward networks refers to the mathematical technique used to calculate derivatives which works as a supervised training algorithm for updating the weights in the network. This direct application of the derivative chain rule helps minimize errors in the network.

RNNsrely on an application of backpropagation when applied to sequence problems like that of time series data called backpropagation through time (BPTT). The BPTT algorithm provides sequential sets of input/output timesteps to the network. It unfolds the network and calculates inaccuracies across each timestep. Finally, the algorithm rolls up the network and updates weights across the network. This entire process is repetitive.

The computational cost of BPTT has a direct relation with the number of timesteps. When the number of timesteps is higher,weight updates become an exhausting process depending upon calculations of derivatives. The weights eventually vanish or explode resulting in noisy model performance and poor learning. A gradient explains the change in all weights with regard to the change in error. Hence, the problem is identified as the vanishing or exploding gradient problem in RNNs.

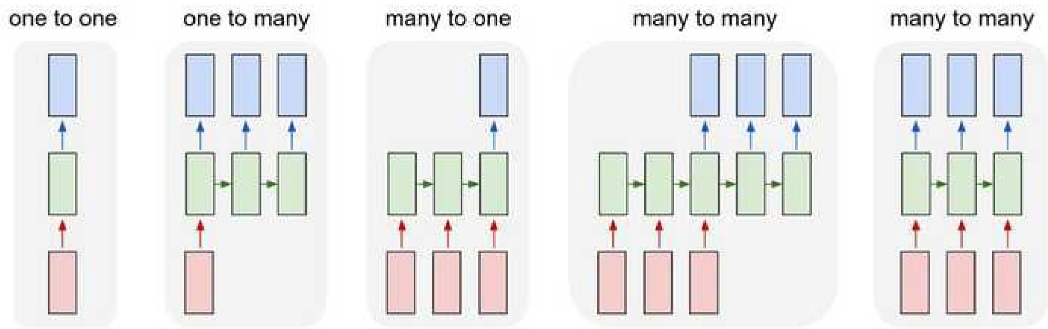


Figure 3.8 Topologies of RNNs. The red represents the inputs, green are the RNNs and the blue represents the outputs. [87]

- **Long Short-Term Memory Networks**

First introduced by Hochreiter & Schmidhuber [88], Long Short-Term Memory networks or LSTMs are a special category of RNNs having the ability to learn long-term dependencies. LSTMs are precisely conceived to cater the long-term dependency issue or the exploding/vanishing gradient problem resulting from BPTT.

All RNNs contain repetitive modules of neural network. The chain like repeating structure in LSTMs is slightly different, consisting of four interactive neural network layers instead of one. The four gates include forget gate, input gate, output gate, and internal hidden state gate represented by *F*, *I* *O*, *G* respectively [89]. See Figure 3.9. These four gates are utilized by LSTMs to perform a specific function of defining a cell state at each timestep. Detailsareappended orwithdrawnfrom the cell state after careful regulation from these gates. The gates comprise a sigmoid neural net layer and a pointwise multiplication operation which output a zero or a one defining how much information to allow pass.

The first step is executed by the forget gate which regulates which information from the previous hidden state, h_{t-1}, to remove or forget from the cell state. The forget gate analyzes h_{t-1} and current data input, x_t, and produces an output between 0 and 1. f_t represents the forget gate in equation 3.7.

$$\mathbf{f_t} = \boldsymbol{\sigma} \left(\mathbf{W_f} \cdot \left[\boldsymbol{h_{t-1}}, \boldsymbol{x_t}\right] + \boldsymbol{b_f}\right) \quad (3.7)$$

The next step is to decide which information to store in the cell state. This is a two-step procedure. Input gate layer decides the values to be updated and a tanh layer generates a vector of new candidate values, C'_t, that could be added to the state. These two are combined to create an update of the state.

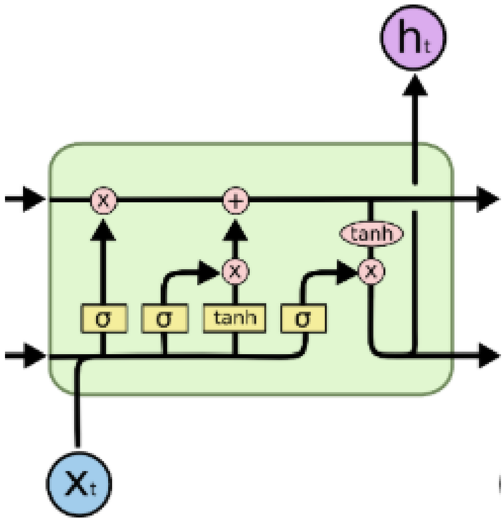


Figure 3.9 The four interacting layers of an LSTM repeating module [89].

$$\mathbf{i_t} = \boldsymbol{\sigma}\,(\mathbf{W_i} \cdot \left[\mathbf{h_{t-1}}, \mathbf{x_t}\right] + \mathbf{b_i}) \qquad (3.8)$$

$$\mathbf{C_t'} = \mathbf{tanh}\,(\mathbf{W_c} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \mathbf{b_i}) \qquad (3.9)$$

In order to update the previous cell state C_{t-1} to current state C_t, C_{t-1} is multipliedto f_t and added to i_t * C'_t.This procedure is mathematically represented in equation 3.10

$$\mathbf{C_t} = \left(\mathbf{ft} \ * \ \mathbf{C_{t-1}}\right) + (\,\mathbf{i_t} * \mathbf{C_t'}\,) \qquad (3.10)$$

Finally, the output is based on a filtered version of the cell state by first applying a sigmoid layer which decides which part of the cell state to output. Next the cell state is passed through a tanh layer to scale values between -1 and 1 and multiplied by the output of the sigmoid gate to only select the decided parts to output. Equations 3.11 and 3.12 represent the procedure described.

$$\mathbf{o_t} = \boldsymbol{\sigma}\,(\mathbf{W_o} \cdot [\mathbf{h_{t-1}}, \mathbf{x_t}] \, + \, \mathbf{b_o}) \qquad (3.11)$$

$\mathbf{h_t = o_t * tanh}\ (\mathbf{C_t})$(3.12)

The output gate decides how much of the current hidden state to expose to the upcoming layer at the next timestep.

- **Bidirectional LSTMs**

Bidirectional LSTMs or BiLSTMs are a widely applied improvement of LSTMs which work better with sequence classification problems. Conventional LSTMs are only able to make use of the previous context but BiLSTMs overcome this shortcoming by processing the data in both directions with two separate hidden layers. The outcomes of the two hidden layers are then fed forwards to the same output layer. This basic concept enables BiLSTMs to access long-range context in both input directions. BiLSTMs utilize both the previous and future context however, the forward pass and backward pass are completely independent of each other.

Bidirectional LSTMs are like training two separate LSTMs on the input sequence given all the timesteps, one that is trained on the input sequence as is and the other on the reversed copy of the input sequence. The input is executed in two ways, one from past to future and the other from future to past. The basic difference from a unidirectional LSTM is the ability of LSTM training in backwards direction to preserve information from the future as well.At any timestep the combined hidden layers preserve information from past and future.

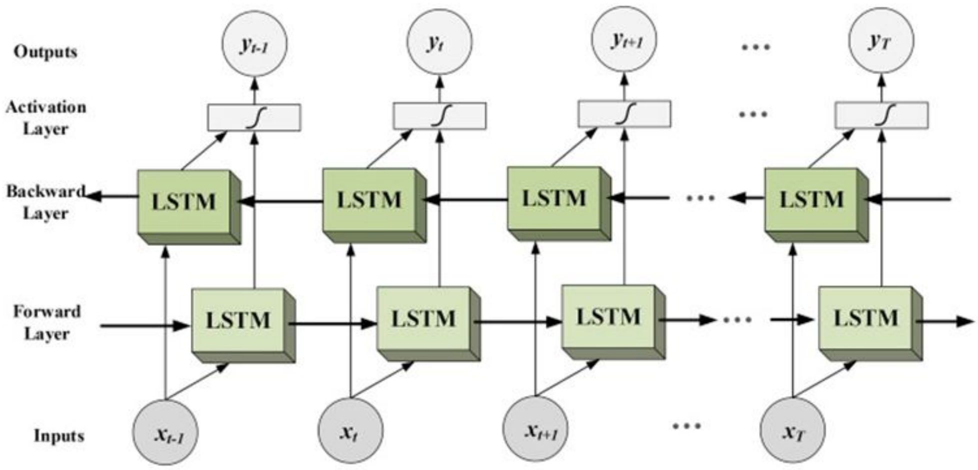
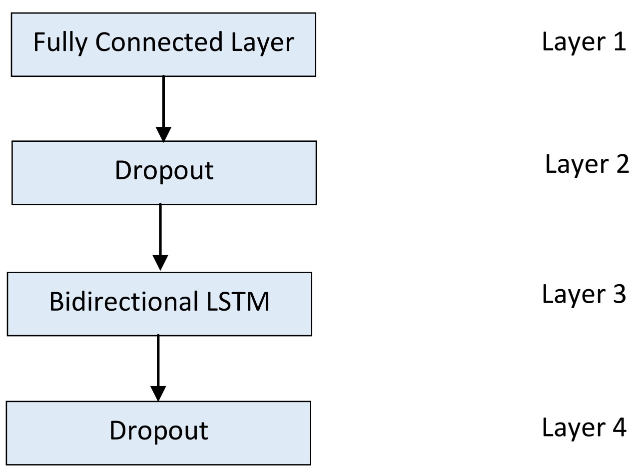


Figure 3.10 Bidirectional LSTMs. x_t are inputs and y_t represent outputs [90]

Based on their ability to retain long term dependencies from both past and future, stacked BiLSTMs in many-to-one configuration are used for the proposed solution of sequence classification in fall detection scenario. The overall representation of proposed fall detection mechanism is represented in Figure 3.11

In addition to the two stacked BiLSTMs, a dropout layers are incorporated to help with the regularization of the neural network. Regularization are the techniques used to prevent the deep neural networks from overfitting the training datasets.



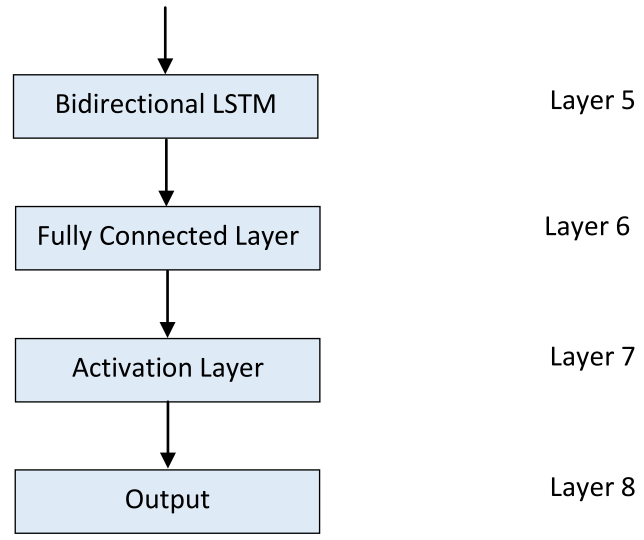
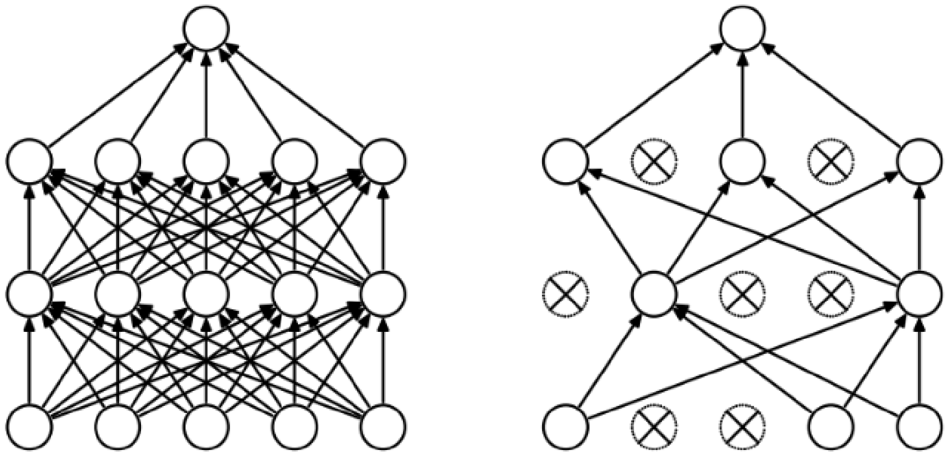


Figure 3.11 Proposed Fall Detection Mechanism

- **Dropout**

Dropout is a regularization technique, where recurrent connections between LSTM units and input are probabilistically excluded from activation and updating of weight during the training of the network. This procedure has the effect of reducing overfitting and improving overall model performance. Network readjustment takes place for every training sample where a set of new neurons are dropped out. While testing, the weights are multiplied by their probability of their associated units' dropout.

The core concept of dropout as proposed by Srivastava el al. [92] is for each hidden unit to learn to work with a randomly chosen sample of other units making it more robust and independent of other hidden units by producing adequate useful features on its own. The probability of input neurons being dropped ranges from 0 to 1, where 0 means no dropout and 1 refers to no connection. See Figure 3.12



(a) (b)

Figure 3.12: (a) Conventional Neural Network, (b) Neural Network with appliedDropout

- **Fully Connected Layer**

The first fully connected layer (layer 1 as shown in Figure 3.11) represents the input layer where all the data preprocessing is done. This layer provides data to the BiLSTMs after passing through dropout regularization layers. The second fully connected layer (layer 6) collects all the output information from the second BiLSTM of the stack (layer 5) and passes it into the activation layer (layer 7) which is responsible for the final classification. The values at the final fully connected layer (layer 6) get multiplied by some weights and are passed through an activation layer. The activation layer aggregates them into a vector which contains one value per every class of the model.

- **Activation Layer**

The purpose of the activation layer at the end of a neural network classification model is to convert the score produced by the neural network into values or classes that can be interpreted by humans. For the purpose of this research, softmax activation is used which helps a multi-class classifier assign instances to one of the classes. The softmax function works like the arg max function which does not return the largest value from the input, but the position of the largest values. Given the scores in the form of vector from the fully connected layer (layer 6), the softmax layer returns the probability of the largest value being the i-th element of the vector.

Algorithm 1 summarizes the proposed mechanism of fall detection which treats missing values in sensor data.

Algorithm 1: Algorithm for Deep Learning based Missing Data Imputation and Fall Detection

---

(1) **Data Preprocessing:**
   (a) Adjustment of activities durations, resampling and data selection.
   (b) Data annotation into classes: FALL / ADL.

(2) **Missing Data Pattern Generation:**
   (a) Creation of MCAR pattern of missingness.
   (b) Creation of noisy datasets with different percentages of MCAR missingness: 20%, 30%, 40%.

(3) **Train/Test Split:**
   (a) Division of datasets for training, testing and validation stages.

(4) **3D Data Reshaping:**

(a) Transformation of training data from 2D (samples, timesteps) to 3D (samples, timesteps, features) datasets.

(5) **Deep Learning Based Fall Detection:**

(a) Creation of neural network based on stacked BiLSTMs, dropout layers, activation and fully connected layers.

(6) **BiLSTM Network Training:**

(a) Hyperparameters Optimization: choice of loss function, optimizer, hidden units, hidden layers, output layer with activations.

(b) Training of Network: Choice of batch size, early stopping percentage and epochs.

(7) **BiLSTM Prediction:**

(a) Sequence Classification:Use of trained data to predict classes (FALL/ADL) during testing.

(8) **Performance Evaluation:**

(a) Model loss and classification accuracy analysis.

(b) Confusion matrix creation.

(c) Effectiveness analysis: calculation of precision, sensitivity and specificity.

3.3 Experimental Setup

The proposed fall detection mechanism has been implemented using the Keras framework, a high-level framework for deep learning for Python programming language. All training procedures have been performed on LENOVO 80MK workstation, equipped with an Intel® Core™ i7-6500U CPU.

The fall detection problem is treated as a sequence classification problem with fixed length sequence input using two BiLSTMs stacked on top of each other. Each BiLSTM layer contains 32 hidden neurons. Batch size is 2048 and dropout probability used in both layers is 0.2. The dropout layers are implemented only during the training phase and removed during testing. Categorical cross entropy is chosen as the loss function. All BiLSTMs are trained using early stopping. A 90%/10% train/test split is used while the train/validation split is kept 80%/20%. A softmax activation function is applied at the output for each input sequence.

CHAPTER: 4

PERFORMANCE EVALUATION

This section gives a detailed description of the results obtained for the two approaches applied on the fall detection problem in presence of missing values. One approach uses a collaborative method of data collection where data from all three sensors is used to provide information for detecting falls. The other approach uses information collected from individual sensors to detect falls. The two approaches also cater the missing data problem where MCAR missingness of various percentages (20%, 30%, 40%) is added to the collected sensor data.

4.1 Performance Evaluation Metrics

Using classification accuracy alone as a performance measure can be confusing if there is an unequal number of observations in each class or if there are more than two classes in the dataset. Computing a confusion matrix gives better understanding of what the classification model is predicting correctly and the types of errors it is making. Confusion matrix, also known as the error matrix, sketches a better visualization of the classification algorithm and helps to identify the confusion between classes. It gives an insight about the kind of errors made by the model. Most evaluation metrics are computed from the confusion matrix. It summarizes the number of correct and incorrect predictions with count values and divides them into each class. This is the basic concept of a confusion matrix. The rows depict actual classes while the columns represent the predicted class outcomes by the classifier. Some terms used to define a confusion matrix include:

- Positive (P) : Observation is positive.
- Negative (N) : Observation is not positive.
- True Positive (TP) : Observation is positive. The prediction is positive.
- False Negative (FN) : Observation is positive, but the prediction is negative.
- True Negative (TN) : Observation is negative.The prediction isnegative.
- False Positive (FP) : Observation is negative,the prediction is positive.

	Predicted Positive	Predicted Negative

Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Figure 4.1 Confusion Matrix Layout

Classification rate or accuracy is the ratio of error-free predictions made to the total number of predictions made by the classification model.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (4.1)$$

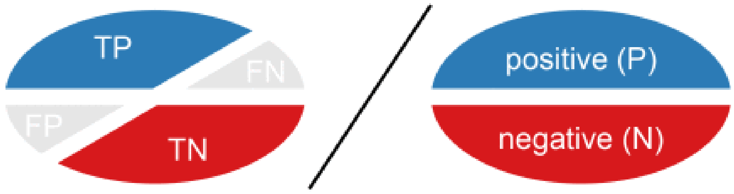


Figure 4.2 Accuracy of Classifier

However, this accuracy can be problematic since it includes both kinds of errors.

Error rate is the ratio of all incorrect predictions to the total number of predictions made by the classification model.

$$\text{Error Rate} = (\text{FP} + \text{FN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (4.2)$$

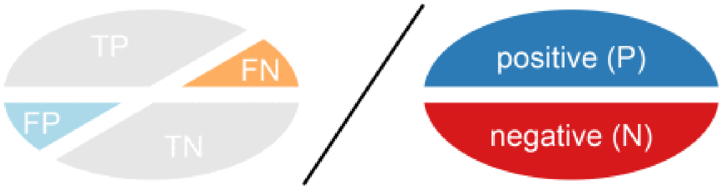


Figure 4.3 Error Rate of Classifier

Another informative measure is sensitivity or recall. Sensitivity is the ratio of number of correct positive predictions to the total number of positives.

$$\text{Sensitivity} = TP / TP + FN \qquad (4.3)$$

Figure 4.4 Sensitivity of Classifier

Specificity is the ratio of the number of correct negative predictions to the total number of negatives.

$$\text{Specificity} = TN / TN + FP \qquad (4.4)$$

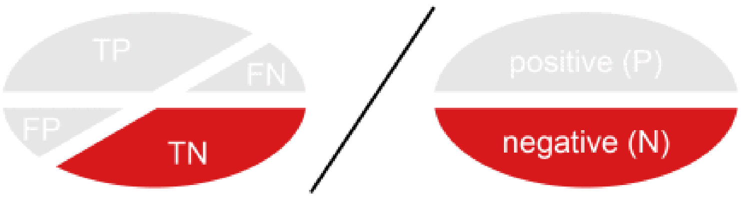


Figure 4.5 Specificity of Classifier

Precision (PREC) is the ratio of the number of error-free positive predictions to the total number of positive predictions. It is also called positive predictive value.

$$\text{Precision} = \text{TP} / \text{TP} + \text{FP} \tag{4.5}$$

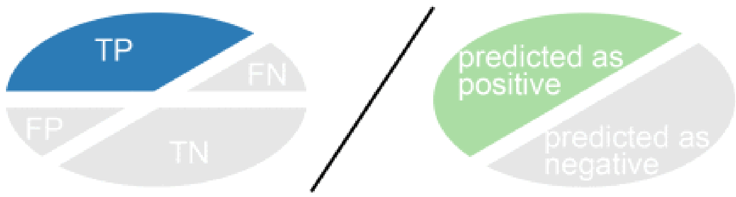


Figure 4.6 Precision of Classifier

4.2 Data Distribution

Examination of data distribution is essential in order to ensure that the proposed sequence classification model doesn’t create bias due to data imbalance. The dataset considered for experiments contains 770 sequences comprising of 395 performed ADLs and 375 falls. Each sequence or sample consists of 2400 instances or observations. The total instances of the considered dataset accumulate to 1,848,000. A train/test split of 90%/10% is adopted for the purpose of this research. This split trains the model on 693 samples and evaluates it on 77 samples. Table 4.1 describes the distribution of the two activities (ADLs and falls) in training and testing datasets.

	Train	**Test**
ADLs	362	33
Falls	331	44
Total	693	77

Table 4.1 Distribution of ADLs and Falls in training and testing datasets

4.3 Performance Evaluation

4.3.1 The Combined Sensors Approach

This section describes the experimental results of using a combination of sensors to detect falls. Three tri-axial sensors (2 accelerometers and 1 gyroscope), working at a sampling frequency of 200 Hz are chosen to provide collected data from subjects executing15 categories of falls and 19 different ADLs. The experiment is performed on 4 different case scenarios: dataset with complete records and no observed missing values, incomplete dataset consisting of 80% of original data and 20% of the data observed as missing values through MCAR mechanism, , incomplete dataset consisting of 70% of original data and 30% of the data observed as missing values through MCAR mechanism and incomplete dataset consisting of 60% of original data and 40% of the data observed as missing values through MCAR mechanism. Extensive hyperparameters tuning and optimization performed on a Keras framework using the Python programming language lead to the results mentioned in Table 4.2. The hyperparameters selected for the proposed fall detection mechanism are described in the experimental setup section.

% of Original Data Observed	% of MCAR Missing Values Observed	Training Accuracy	Test Accuracy	Training Loss	Test Loss
100	0	98.01 %	97.4 %	0.0749	0.1198
80	20	96.39 %	94.81 %	0.1002	0.107
70	30	95.85 %	92.21 %	0.1205	0.2259
60	40	88.81 %	88.31 %	0.2694	0.282

Table 4.2 Results for Combined Sensors Approach: Accuracy and Loss during training and testing phases

Figure 4.7 shows the confusion matrix, for the analyzed four scenarios, resulting from testing the proposed model with categorical cross entropy as the applied loss function and with the optimal choice of other hyperparameters. The model is trained on 693 samples and evaluated on 77, results of which are described in the figure.

Predicted Label

ADL **FALL**

True Label

ADL

FALL

33	0
2	42

(a)

32	1
3	41

(b)

31	2
3	41

(c)

27	6
3	41

(d)

Figure 4.7 Confusion matrix resulting from testing the multisensor fusion approach: (a) 0% missingness observed in data, (b) 20% missingness observed in data, (c) 30% missingness observed in data, (d) 40% missingness observed in data

4.3.2 The Single Sensor Approach

This section describes the experimental results of using individual sensors to detect falls. While the single sensor approach chooses one sensor at a time to detect falls. This is done to analyze the behavior of individual sensor data and how it contributes towards fall detection. The sensors considered for this approach are the ITG3200 gyroscope and the MMA8451Q accelerometer. All three percentages of missingness have been considered for the individual sensor case. The same hyperparameters optimized for the multisensor fusion approach are applied when using the individual sensors.

An accelerometer is a device used to measure static and dynamic accelerations of a body. Measurement of static acceleration which is caused by the gravity helps to calculate the angle the device is tilted at with respect to the earth. Measurement of dynamic acceleration can help analyse the way the device is moving. HAR based on wearable sensors rely heavily on the information provided by accelerometers. For the purpose of this research, MMA8451Q accelerometer readings, collected at a sampling frequency of 200 Hz and a resolution of 14 bits, are used to detect falls for the four scenarios of missing values observation. Table 4.3 gives accuracies and losses for training and testing phases of only using MMA8451Q accelerometer for fall detection.

Figure 4.8 shows the confusion matrix, for the analyzed four scenarios of using accelerometer in the single sensor approach, resulting from testing the proposed model with categorical cross entropy as the applied loss function and with the optimal choice of other hyperparameters. The model is trained on 693 samples and evaluated on 77, results of which are described in the figure.

% of Original Data Observed	% of MCAR Missing Values Observed	Training Accuracy	Test Accuracy	Training Loss	Test Loss
100	0	97.65 %	96.1 %	0.084	0.1224
80	20	94.4 %	93.51 %	0.1571	0.196
70	30	87.73 %	87.01 %	0.3569	0.3765
60	40	82.67 %	81.82 %	0.4084	0.3827

Table 4.3 Results for Single Sensors Approach using Accelerometer: Accuracy and Loss during training and testing phases

A gyroscope sensor or angular velocity sensor is used to measure and maintain the orientation and angular velocity of a body. While accelerometers only have the capability of measuring linear motion, gyroscopes aremore advanced with the ability to measure the tilt and lateral orientation of an object. Hence, analyzing the individual contribution of a gyroscope to detect falls enhances thescope of the research. Table 4.4 gives accuracies and losses for training and testing phases of only using ITG3200 gyroscope for fall detection.

Figure 4.9 shows the confusion matrix, for the analyzed four scenarios of using gyroscope in the single sensor approach, resulting from testing the proposed model with categorical cross entropy as the applied loss function and with the optimal choice of other hyperparameters. The model is trained on 693 samples and evaluated on 77, results of which are described in the figure.

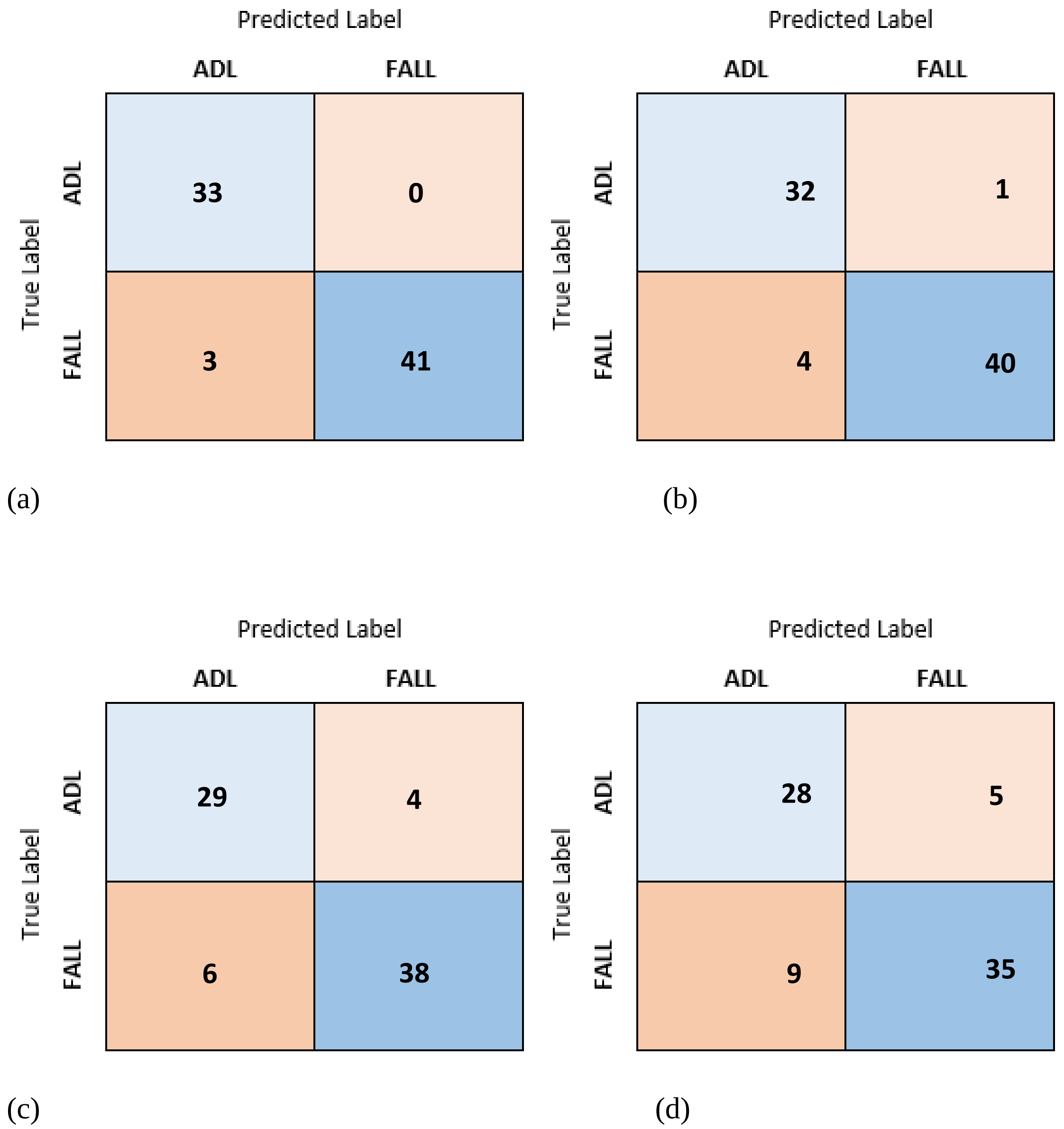


Figure 4.8 Confusion matrix resulting from testing the single sensor approach using accelerometer: (a) 0% missingness observed in data, (b) 20% missingness observed in data, (c) 30% missingness observed in data, (d) 40% missingness observed in data

% of Original Data Observed	% of MCAR Missing Values Observed	Training Accuracy	Test Accuracy	Training Loss	Test Loss
100	0	77.62 %	74.03 %	0.4548	0.4754
80	20	70.04 %	66.23 %	0.5252	0.6135
70	30	64.80 %	62.34 %	0.6276	0.6207
60	40	57.76 %	46.75 %	0.6985	0.7331

Table 4.4 Results for Single Sensors Approach using Gyroscope: Accuracy and Loss during training and testing phases

4.4 Effectiveness Analysis

For the purpose of evaluating the proposed approach, this section discusses the effectiveness analysis by calculating different metrics on the basis of obtained confusion matrices.Since accuracy alone isn't enough to understand the detailed breakdown of performance, other evaluation metrics including error rate, sensitivity, specificity and precision are calculated for the two approaches. This analysis gives a deeper insight of how the model performs during evaluation at testing phase.

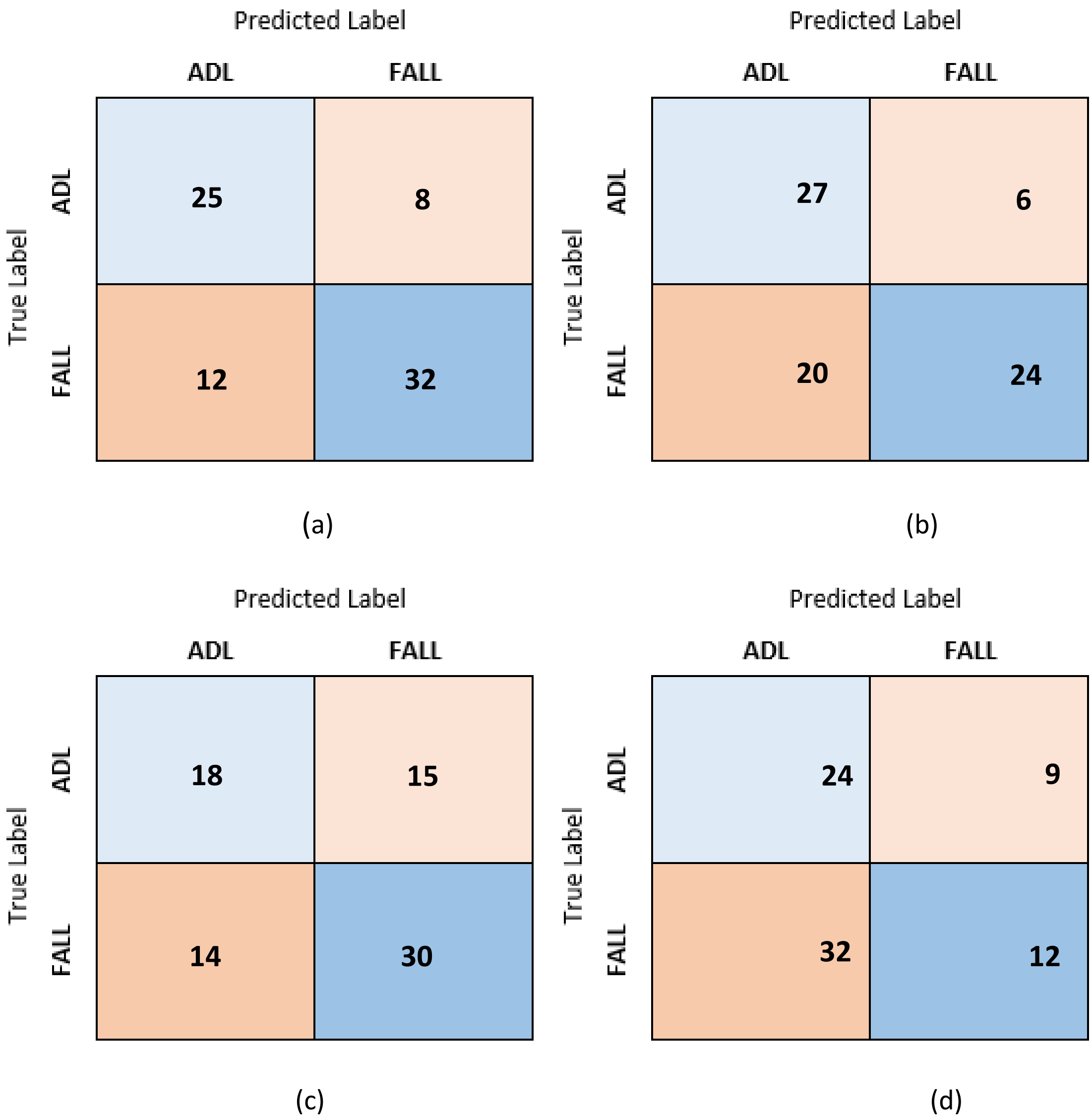


Figure 4.9 Confusion matrix resulting from testing the single sensor approach using gyroscope: (a) 0% missingness observed in data, (b) 20% missingness observed in data, (c) 30% missingness observed in data, (d) 40% missingness observed in data

Table 4.5 describes the results of effectiveness analysis applied on the multisensor fusion approach.

% of Original Data Observed	% of MCAR Missing Values Observed	Error Rate	Sensitivity	Specificity	Precision
100	0	0.0259	100 %	95.45 %	94.28 %
80	20	0.0519	96.97 %	93.18 %	91.43 %
70	30	0.0649	93.93 %	93.18 %	91.17 %
60	40	0.1168	81.81 %	93.18 %	90 %

Table 4.5 Effectiveness analysis for the multisensor fusion approach

Table 4.6 describes the results of effectiveness analysis applied on the single sensor approach using accelerometer for fall detection.

% of Original Data Observed	% of MCAR Missing Values Observed	Error Rate	Sensitivity	Specificity	Precision
100	0	0.039	100 %	93.18 %	91.67 %
80	20	0.065	96.97 %	90.90 %	88.89 %
70	30	0.13	87.87 %	86.36 %	82.85 %
60	40	0.181	84.84 %	79.54 %	75.67 %

Table 4.6 Effectiveness analysis for the single sensor approach using accelerometer

Table 4.7 describes the results of effectiveness analysis applied on the single sensor approach using gyroscope for fall detection.

% of Original Data Observed	% of MCAR Missing Values Observed	Error Rate	Sensitivity	Specificity	Precision
100	0	0.26	75.75 %	72.72 %	67.55 %
80	20	0.338	81.81 %	54.54 %	57.44 %
70	30	0.377	54.54 %	68.18 %	56.25 %
60	40	0.532	72.72 %	27.27 %	42.85 %

Table 4.7 Effectiveness analysis for the single sensor approach using gyroscope

CHAPTER: 5

CONCLUSION

The aim of this research was to apply deep learning techniques to resolve the missing values predicament faced during fall detection and to devise a reliable fall detection mechanism for the applications of IoMT. More specifically, stacked bidirectional LSTMs blocks are used to impute the missing data as well as to classify falls and ADLs.

After a thoughtful analysis and comprehensive literature review of publicly available datasets, SisFall deemed most appropriate for the problem at hand. The choice of bidirectional LSTMs for imputation and sequence classification relies on their ability to retain information and long-term dependencies from past and future. The usage of LSTMs led to the annotation of dataset to make it more suitable for a deep learning-based approach. 3D data reshaping is done to make the dataset cohesive with the dimensionality requirements of LSTMs. In order to overcome the imbalance in dataset, a standard duration window is chosen for all performed activities. The research evaluates different patterns of missingness observed in data and generates MCAR missingness in data in different proportions which is later handled by the proposed fall detection mechanism. Finally, the research proposes two approaches based on the number and type of wearable sensors used to accumulate the data. The end result is the successful distinction between falls and ADLs, the details of which are summarized below.

5.1 Conclusion and Analysis

The comparison between the combined sensors approach and single sensor indicates the superiority of the former where the multisensor fusion approach generates 97.4% accuracy when classifying falls and ADLs for the complete dataset, 94.81% for the best-case scenario of missingness (i.e. 20%) and 88.31% for the worst (i.e. 40%). The effectiveness analysis supports the conclusion with 100%, 96.97% and 81.81% sensitivity for the three cases respectively. The precision for complete dataset, data with 20% missing values and data with 40% missing values are 94.28%, 91.43% and 90% respectively.

When comparing the performances of individual sensors in the single sensor approach, use of accelerometer outperforms that of gyroscope with accuracy for complete dataset, data with 20% missing values and data with 40% missing values to be 96.1%, 93.51% and 81.82% respectively. The effectiveness analysis generates sensitivity of 100%, 96.97% and 84.84% respectively for the three cases using accelerometer. The precision when using accelerometer as the single sensor for complete dataset, data with 20% missing values and data with 40% missing values are 91.67%, 88.89% and 75.67% respectively.

5.2 Future Work

Few possibilities for future work resulting from this research can be:

- Power usage and resources consumption analysis for the proposed fall detection system
- Performance improvement when using individual sensors for data accumulation for imputation and fall detection
- Application of 4^{th} order IIR Butterworth low-pass filter or an averaging filter at the preprocessing stage to reduce signal noise
- Evaluation of the impact of different sampling rates for the sensors

BIBLIOGRAPHY

[1] Roser, M. and Ritchie, H. (2018). Technological progress. Retrieved on January 19, 2018 fromhttps://ourworldindata.org/technological-progress.

[2] Probabilistic Recovery of Incomplete Sensed Data in IoT by Berihun Fekade, Taras Maksymyuk, Member, IEEE, Maryan Kyryk, and Minho Jo, Senior Member, IEEE in IEEE INTERNET OF THINGS JOURNAL, VOL. 5, NO. 4, AUGUST 2018

[3] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: definition, potentials, and societal role of a fast-evolving paradigm," Ad Hoc Networks, vol. 56, pp. 122-140, 2017.

[4] https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

[5] Falls, September 2019, http://www.who.int/en/news-room/fact-sheets/detail/falls.

[6] Mubashir, Muhammad, Ling Shao, and Luke Seed. "A survey on falldetection: Principles and approaches." Neurocomputing 100 (2013):144-152

[7] A Survey on Vision-based Fall Detection by Zhong Zhang, Christopher Conly, and Vassilis Athitsos, PETRA ’15, July 01 - 03 2015, Island of Corfu, Greece

[8] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani., Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Transactions on Biomedical Engineering, 60:427–436, February 2013.

[9] Y. T. Liao, C.-L. Huang, and S.-C. Hsu. Slip and fall event detection using bayesian belief network. Pattern recognition, 45(1):24–32, 2012.

[10] Nguyen, Viet Anh, Thanh Ha Le, and Thuy Thi Nguyen. "Single camera-based Fall detection using Motion and Human shape Features." Proceedings of the Seventh Symposium on Information and Communication Technology. ACM, 2016.

[11] Foroughi, Homa, et al. "An eigenspace-based approach for human fall detection using integrated time motion image and neural network." Signal Processing, 2008. ICSP 2008. 9th International Conference on. IEEE, 2008

[12] A simple vision-based fall detection technique for indoor video Surveillance by Jia-Luen Chua · Yoong Choon Chang ·Wee Keong Lim, SIViP (2015) 9:623–633

[13] Auvinet, Edouard, et al. "Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution." IEEE Transactions on Information Technology in Biomedicine 15.2 (2011): 290-300.

[14] G. Diraco, A. Leone, and P. Siciliano. An active vision system for fall detection and posture recognition in elderly healthcare. In Design, Automation & Test in Europe Conference & Exhibition, pages 1536–1541, March 2010.

[15] a Survey on Vision-based Fall Detection, Zhong Zhang, Christopher Conly, and Vassilis Athitsos

[16] Mastorakis, G.; Makris, D. Fall detection system using Kinect's infrared sensor. J. Real Time Image Process. 2014, 9, 635–646.

[17] Gasparrini, S.; Cippitelli, E.; Spinsante, S.; Gambi, E. A Depth-Based Fall Detection System Using a Kinect® Sensor. Sensors 2014, 14, 2756–2775.

[18] Shengke Wang & Long Chen & Zixi Zhou & Xin Sun & Junyu Dong, Human fall detection in surveillance video based on PCANet, Multimed Tools Appl (2016) 75:11603–11613

[19] Hirata, Y.; Komatsuda, S.; Kosuge, K. Fall prevention control of passive intelligent walker based onhuman model. In Proceedings of the International Conference on Intelligent Robots and Systems,Nice, France, 22–26 September 2008; pp. 1222–1228.

[20] Alwan, M.; Rajendran, P.J.; Kell, S.; Mack, D.; Dalal, S.; Wolfe, M.; Felder, R. A smart

and passive floor-vibration based fall detector for elderly. In Proceedings of the InternationalConference on Information and Communication Technologies, Damascus, Syria, 24–28 April2006; pp. 1003–1007

[21] Sixsmith, A.; Johnson, N.; Whatmore, R. Pyroelectric IR sensor arrays for fall detection in theolder population. J. Phys. 2005, 128, 153–160.

[22] Scott, T. Bed Exit Detection Apparatus. Google Patents, 2000. Available online: http://www.google.com/patents/US6067019 (accessed in January 2020)

[23]Patel, S., Park, H., Bonato, P. et al., A review of wearable sensors and systems with application in rehabilitation, J NeuroEngineering Rehabil (2012) 9: 21. https://doi.org/10.1186/1743-0003-9-21

[24] Toreyin, B. Ugur, et al. "Falling person detection using multisensory signal processing." EURASIP Journal on Advances in Signal Processing 2008 (2008): 29.

[25] Töreyin, B. U., Y. Dedeoğ lu, and A. E. Çetin. "HMM based falling person detection using both audio and video." 2006 , IEEE 14th Signal Processing and Communications ApplicationsConference. Vol. 2006. 2006

[26] Ahmet Yazar, Fatih Erden and A. Enis Cetin, Multi-Sensor Ambient Assisted Living System for Fall Detection, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’14). 2014.

[27] Chen, G.; Huang, C.; Chiang, C. A reliable Fall Detection System Based on Wearable Sensor and Signal Magnitude Area for Elderly Residents. In Aging Friendly Technology for Health and Independence; Lee, Y., Bien, Z.Z., Mokhtari, M., Kim, J.T., Park, M., Kim, J., Lee, H., Khalil, I., Eds.; Springer Berlin Heidelberg: Berlin Heidelberg, Germany, 2010; pp. 267–270.

[28] Chen, D.; Feng, W.; Zhang, Y.; Li, X.; Wang, T. A wearable wireless fall detection system
with accelerators. In Proceedings of the International Conference on Robotics and Biomimetics(ROBIO), Karon Beach, Phuket, 7–11 December 2011; pp. 2259–2263.

[29] Werner, F.; Diermaier, J.; Panek, P.; Schmid, S. Fall detection with distributed floor-mountedaccelerometers: An overview of the development and evaluation of a fall detection system withinthe project eHome. In Proceedings of the 5th International ICST Conference on PervasiveComputing Technologies for Healthcare, Dublin, Ireland, 23–26 May 2011; pp. 354–361.

[30] Lai, Chin-Feng, et al. "Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling." IEEE Sensors Journal 11.3 (2011): 763-770.

[31] Wang, J.; Zhang, Z.; Li, B.; Lee, S.; Sherratt, R.S. An enhanced fall detection system for elderly person monitoring using consumer home networks. IEEE Trans. Consum. Electron. 2014, 60, 23–29.

[32] M. Popescu, Y. Li, M. Skubic, M. Rantz, “An Acoustic Fall Detector System that Uses Sound Height Information to Reduce the False Alarm Rate”, in Proc. 30th Int. IEEE Eng. in Medicine and Bio. Soc. Conference, pp. 4628-4631, Aug. 2008.

[33] Andò, Bruno, et al. "An event polarized paradigm for ADL detection in AAL context." IEEE Transactions on instrumentation and Measurement 64.7 (2015): 1814-1825.

[34] Isabel N. Figueiredo, Carlos Leal, Luís Pinto, Jason Bolito and André Lemos, Exploring smartphone sensors for fall detection, The Journal of Mobile User Experience (2016) 5:2 DOI 10.1186/s13678-016-0004-1

[35] He, Y.; Li, Y. Physical Activity Recognition Utilizing the Built-In Kinematic Sensors of a Smartphone. Int. J. Distrib. Sens. Netw. 2013, doi:10.1155/2013/481580.

[36] Fortino, G.; Gravina, R. Fall-MobileGuard: A smart real-time fall detection system. In Proceedings of the 10th EAI International Conference on Body Area Networks, Sydney, Australia, 28–30 September 2015; pp. 44–50.

[37] Mohammad Ashfak Habib, Mas S. Mohktar , Shahrul Bahyah Kamaruzzaman , Kheng Seang Lim , Tan Maw Pin and Fatimah Ibrahim, Smartphone-Based Solutions for Fall Detection and Prevention: Challenges and Open Issues, Sensors 2014, 14, 7181-7208; doi:10.3390/s140407181

[38] A.K.Bourke, G.M.Lyons , A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor, Medical Engineering & Physics, Volume 30, Issue 1, January 2008, Pages 84-90

[39] Gregory Koshmak, Amy Loutfi, and Maria Linden, Challenges and Issues in Multisensor Fusion Approach for Fall Detection: Review Paper, Hindawi Publishing Corporation Journal of Sensors, Volume 2016, Article ID 6931789, 12 pages, doi:/10.1155/2016/6931789

[40] Y. Li, T. Banerjee, M. Popescu, and M. Skubic, "Improvement of acoustic fall detection using Kinect depth sensing," in Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '13), pp. 6736–6739, IEEE, Osaka, Japan, July 2013.

[41] A. Yazar, F. Erden, and A. E. Cetin, "Multi-sensor ambient assisted living system for fall detection," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '14), pp. 1–3, 2014.

[42] Haobo Li ; Aman Shrestha ; Francesco Fioranelli ; Julien Le Kernec ; Hadi Heidari ; Matteo Pepa ; Enea Cippitelli, Ennio Gambi, Susanna Spinsante, Multisensor data fusion for human activities classification and fall detection, 2017 IEEE Sensors.

[43] De Cillis, F.; de Simio, F.; Guido, F.; Incalzi, R.A.; Setola, R. Fall-detection solution for mobile platforms using accelerometer and gyroscope data. In Proceedings of the 37th Annual

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015.

[44] Ando, B.; Baglio, S.; Lombardo, C.O.; Marletta, V. A multisensor data-fusion approach for ADL and fall classification. IEEE Trans. Instrum. Meas. 2016, 65, 1960–1967.

[45] Pierleoni, P.; Belli, A.; Maurizi, L.; Palma, L.; Pernini, L.; Paniccia, M.; Valenti, S. A wearable fall detector for elderly people based on AHRS and barometric sensor. IEEE Sens. J. 2016, 16, 6733–6744.

[46] Ntanasis, P.; Pippa, E.; Özdemir, A.T.; Barshan, B.; Megalooikonomou, V. Investigation of sensor placement for accurate fall detection. In Lecture Notes of the Institute for Computer Sciences; Perego, P., Andreoni, G., Rizzo, G., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 225–232.

[47] Panagiotis Tsinganos and Athanassios Skodras, On the Comparison of Wearable Sensor Data Fusion to a Single Sensor Machine Learning Technique in Fall Detection, Sensors 2018, 18, 592; doi:10.3390/s18020592

[48] Tuan Nguyen Gia, Igor Tcarenko, Victor K. Sarker, Amir M. Rahmani, Tomi Westerlund,

Pasi Liljeberg, and Hannu Tenhunen, IoT-Based Fall Detection System with Energy

Efficient Sensor Nodes, IEEE Nordic Circuits and Systems (Norcas'16), DOI: 10.1109/NORCHIP.2016.7792890

[49] Diana Yacchirema,, Jara Suárez de Puga, Carlos Palau, Manuel Esteve, Fall detection system for elderly people using IoT and Big Data, The 9th International Conference on Ambient Systems, Networks and Technologies, 2018

[50] Dharmitha Ajerla, Sazia Mahfuz , and Farhana Zulkernine, A Real-Time Patient Monitoring Framework for Fall Detection, Hindawi, Wireless Communications and Mobile Computing, Volume 2019, Article ID 9507938, 13 pages, DOI: 10.1155/2019/9507938

[51] Dennis Sprute, Aljoscha Pörtner, Matthias Koenig, Smart Fall: Accelerometer-Based Fall Detection in a Smart Home Environment, 13th International Conference on Smart Homes and Health Telematics (ICOST), Geneva, Switzerland, June 2015, DOI: 10.1007/978-3-319-19312-0_16

[52] Yi-Zeng Hsieh, Yu-Lin Jeng, Development of Home Intelligent Fall Detection IoT System Based on Feedback Optical Flow Convolutional Neural Network, IEEE Access, Volume 6, 2017, DOI: 10.1109/ACCESS.2017.2771389

[53] Xiaomu Luo, Tong Liu, Jun Liu, Xuemei Guo and Guoli Wang, Design and implementation of a distributed fall detection system based on wireless sensor networks, EURASIP Journal on Wireless Communications and Networking 2012, 2012:118

[54] Sanaz Kianoush , Stefano Savazzi ; Federico Vicentini ; Vittorio Rampa ; Matteo Giussani, Device-Free RF Human Body Fall Detection and Localization in Industrial Workplaces, IEEE Internet of Things Journal, Volume: 4 , Issue: 2 , April 2017, DOI: 10.1109/JIOT.2016.2624800

[55] Mohammadi, Leila & van de Geer, Sara. (2003). On threshold-based classification rules. Lecture Notes-Monograph Series. 42. 10.2307/4356242.

[56] Vallejo, M.; Isaza, C.V.; Lopez, J.D. Artificial neural networks as an alternative to traditional fall detection methods. In Proceedings of the 35th Annual International Conference on Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 3–7 July 2013; pp. 1648–1651

[57] Dai, J.; Bai, X.; Yang, Z.; Shen, Z.; Xuan, D. PerFallD: A pervasive fall detection system using mobile phones. In Proceedings of the 8th International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, Germany, 29 March–2 April, 2010; pp. 292–297.

[58] Yueng Santiago Delahoz and Miguel Angel Labrador, Survey on Fall Detection and Fall Prevention Using Wearable and External Sensors, Sensors 2014, 14, 19806-19842; doi:10.3390/s141019806

[59] Yuxi Wang ; Kaishun Wu ; Lionel M. Ni, WiFall: Device-Free Fall Detection by Wireless Networks, IEEE Transactions on Mobile Computing (Volume: 16 , Issue: 2 , Feb. 1 2017), DOI: 10.1109/TMC.2016.2557792

[60] Nukala, B. , Shibuya, N. , Rodriguez, A. , Tsay, J. , Lopez, J. , Nguyen, T. , Zupancic, S. and Lie, D. (2014) An Efficient and Robust Fall Detection System Using Wireless Gait Analysis Sensor with Artificial Neural Network (ANN) and Support Vector Machine (SVM) Algorithms. Open Journal of Applied Biosensor, 3, 29-39. doi: 10.4236/ojab.2014.34004.

[61] Özdemir, A.T.; Barshan, “Detecting Falls with Wearable Sensors Using Machine Learning Techniques” Sensors 2014, 14, 10691-10708.

[62] Stone, E.E.; Skubic, M. Fall Detection in Homes of Older Adults Using the Microsoft Kinect. IEEE J. Biomed. Health Inform. 2015, 19, 290–301.

[63] Tao Xu, Yun Zhou, and Jing Zhu, New Advances and Challenges of Fall Detection Systems: A Survey, MDPI journal, Applied Sciences. 2018, 8, 418; doi:10.3390/app8030418

[64] Jacob Nogas, Shehroz S. Khan, Alex Mihailidis, Fall Detection from Thermal Camera Using Convolutional LSTM Autoencoder, Conference: 2nd Workshop on AI for Aging, Rehabilitation and Independent Assisted Living at IJCAI, Federated Artificial Intelligence MeetingAt: Stockholm, Sweden, July 2018.

[65] Musci, M.; De Martini, D.; Blago, N.; Facchinetti, T.; Piastra, M. Online Fall Detection using Recurrent Neural Networks. arXiv 2018, arXiv:1804.04976.

[66] Taylor R. Mauldin, Marc E. Canby, Vangelis Metsis, Anne H. H. Ngu, and Coralys Cubero River, SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning, Sensors 2018, 18, 3363; doi:10.3390/s18103363

[67] I Wayan Wiprayoga Wisesa and Genggam Mahardika, Fall detection algorithm based on accelerometer and gyroscope sensor data using Recurrent Neural Networks 2019 IOP Conf. Ser.: EarthEnviron. Sci. 258 012035

[68] Vavoulas, George & Pediaditis, Matthew & Spanakis, Emmanouil & Tsiknakis, Manolis. (2013). The MobiFall dataset: An initial evaluation of fall detection algorithms using smartphones. 13th IEEE International Conference on BioInformatics and BioEngineering, IEEE BIBE 2013. 1-4. 10.1109/BIBE.2013.6701629.

[69] C. Medrano, R. Igual, I. Plaza, and M. Castro, "Detecting falls as novelties in acceleration patterns acquired with smartphones," PLOS ONE, vol. 9, no. 4, pp. 1–9, 04 2014.

[70] Frank, K.; Vera, M.J.; Robertson, P.; Pfeifer, T. Bayesian Recognition of Motion Related Activities with Inertial Sensors. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp), Copenhagen, Denmark, 26–29 September 2010; pp. 445–446

[71] Vilarinho, T.; Farshchian, B.; Bajer, D.G.; Dahl, O.H.; Egge, I.; Hegdal, S.S.; Lones, A.; Slettevold, J.N.; Weggersen, S.M. A combined smartphone and smartwatch fall detection system. In Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015.

[72] Casilari-Pérez, Eduardo & Santoyo Ramón, José & Cano-Garcia, Jose Manuel. (2017). UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection. Procedia Computer Science. 110. 32-39. 10.1016/j.procs.2017.06.110.

[73] Angela Sucerquia, José David López and Jesús Francisco Vargas-Bonilla, SisFall: A Fall and Movement Dataset, Sensors 2017, 17, 198; doi:10.3390/s17010198

[74] Luengo, J., 2011: Missing Values in Data Mining [Online] Available at: http://sci2s.ugr.es/MVDM/index.php

[75] Donald B. Rubin, Inference and missing data, Biometrika 1976, 63, 3, pp. 581-92

[76] Kantardzic M. 2003: Data Mining – Concepts, Models, Methods, and Algorithms, IEEE, pp. 165-176.

[77] Lakshminarayan, K., Harp S. A. & Samad, T., 1999: Imputation of Missing Data in Industrial Databases, Applied Intelligence 11, pp. 259–275. Available at: http://www.ime.unicamp.br/~wanderson/Artigos/imputation_industrial_databases.pdf

[78] Schafer JL, Olsen MK, Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective, Multivariate Behav Res. 1998 Oct 1;33(4):545-71. doi: 10.1207/s15327906mbr3304_5.

[79] Grzymala-Busse J. W., Hu M., 2001: A Comparison of Several Approaches to Missing Attribute Values in Data Mining, Revised Papers from the Second International Conference on Rough Sets and Current Trends in Computing, pp. 378-385, Available at: http://sci2s.ugr.es/keel/pdf/specific/congreso/grzymala_busse_hu01.pdf

[80] G. E. A. P. A. Batista and M. C. Monard A study of k-nearest neighbour as an imputation
method 2002 Frontiers in Artificial Intelligence and Applications 87 pp 251–260

[81] Gupta, A. & Lam, M., 1998: The weight decay backpropagation for generalizations with missing values, Annals of Operations Research 78, pp. 165-187. Available at: http://link.springer.com/article/10.1023%2FA%3A1018945915940

[82] Schafer JL (1999) Multiple imputation: a primer. Stat Methods in Med 8(1):3–15. doi:10.1177/096228029900800102

[83] Bennett DA (2001) How can I deal with missing data in my study? Aust N Z J Public Health 25(5):464–469. doi:10.1111/j.1467-842X.2001.tb00294.x

[84] Tabachnick BG, Fidell LS (2012) Using multivariate statistics, 6th edition. Allyn & Bacon, Needham Heights, MA

[85] Schafer JL (1997) Analysis of incomplete multivariate data. Chapman & Hall/CRC, London

[86] Little RJA, Rubin DB (2002) Statistical analysis with missing data, 2nd edition. Wiley, New York

[87]https://towardsdatascience.com/an-introduction-to-recurrent-neural-networks-for-beginners-664d717adbd

[88] Sepp Hochreiter and Jurgen Schmidhuber, Long Short-Term Memory, Neural Computation, November 1997, https://doi.org/10.1162/neco.1997.9.8.1735

[89] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[90] https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/

[91] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681

[92] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research 15 (2014)

[93] de Quadros, T.; Lazzaretti, A.E.; Schneider, F.K. A movement decomposition and machine learning-based fall detection system using wrist wearable device. IEEE Sens. J. 2018, 18, 5082–5089.

[94] Yacchirema, D.; de Puga, J.S.; Palau, C.; Esteve, M. Fall detection system for elderly people using IoT and big data, Procedia Comput. Sci. 2018, 130, 603–610.

[95] Saleh, M.; Jeannès, R.L.B. Elderly fall detection using wearable sensors: A low cost highly accurate algorithm, IEEE Sens. J. 2019, 19, 3156–3164.

[96] Fortino, G.; Gravina, R. Fall-MobileGuard: A smart real-time fall detection system. In Proceedings ofthe 10th EAI International Conference on Body Area Networks, Sydney, Australia, 28–30 September 2015; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Brussels, Belgium, 2015; pp. 44–50.

[97] Torti, E.; Fontanella, A.; Musci, M.; Blago, N.; Pau, D.; Leporati, F.; Piastra, M. Embedded real-time fall detection with deep learning on wearable devices. In Proceedings of the 2018 2first IEEE Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 29–31 August 2018; pp. 405–412.

[98] Francisco Luna-Perejón, Manuel Jesús Domínguez-Morales and Antón Civit-Balcells, Wearable Fall Detector Using Recurrent Neural Networks, Sensors 2019, 19, 4885; doi:10.3390/s19224885

[99] Seokhyun Hwang, DaeHan Ahn, Homin Park, Taejoon Park, Maximizing Accuracy of Fall Detection and Alert Systems Based on 3D Convolutional Neural Network, IoTDI 2017, April 2017, Pi,sburgh, PA USA