GAS Journal of Engineering and Technology (GASJET)
Volume 3 | Issue 4, 2026 Homepage: https://gaspublishers.com/gasjet-home/ ISSN: 3048-5800

An IOT-Driven Fall Detection System Using Bi-Directional LSTM for Safety in Elderly

Aisha Umar Faruq, Prof. N. Agwu Nnanna & Prof Marion Adebiyi

Department of Information Technology, Faculty of Computing, Nile University of Nigeria

Received: *10.03.2026* | ***Accepted:*** *31.03.2026* | ***Published:*** *01.04.2026*

***Corresponding Authors:** Aisha Umar Faruq

DOI: *10.5281/zenodo.19358116*

Abstract **Original Research Article**

Life expectancy is rising quickly along with the world's population, especially in wealthier nations. Health care systems are facing serious challenges due to the growing proportion and total number of older persons in the population. As they age and become weaker, elderly people are falling more frequently, which makes it harder for them to remain stable all the time. Therefore, the goal of this research is to create a more intelligent and effective Deep Learning model for fall detection in older populations. In this study, a multi-modal dataset from barometer, magnetometer, accelerometer, and gyroscope motion signals we fused together to improve model generalizability and robustness. The data pre-processing involved data fusion, label processing, feature selection, and data transformation. We implemented three (3) different models which include Random Forest, Bi-LSTM and CNN-LSTM. The Bi-LSTM model performed best with an accuracy, precision, recall and F1-score of 97% for all the performance metrics while Random Forest and CNN-LSTM achieved 89% and 83% respectively for all the performance metrics. Bi-LSTM model performance can be attributed to the adequate data preprocessing and its capability to learn and preprocess sequential data in both backward and forward directions as opposed to the Random Forest model with a very low adaptability to sequential data, while the CNN-LSTM have the capability to learn from sequential data, its strength lies in image sequence dataset. This study can be adapted for the improvement in the field of fall detection.

Keyword: Fall, LSTM, Safety, Machine learning, IOT.

CHAPTER ONE
INTRODUCTION

1.1 Background of the study

Life expectancy is rising significantly along with the world's population, especially in developed nations (Gaya-Morey et al., 2024). Health care systems are facing enormous challenges as the proportion and total number of older individuals in the population are increasing (Denkovski et al., 2023). Falls among the elderly continue to increase in frequency as people age and become weaker, making it harder for them to maintain stability at all times (Sundaram et al., 2023). Among older adults, falls are a major source of injury and mortality (Luo et al., 2024). Falls can frequently result in severe outcomes, including disability and even death, especially for vulnerable groups like the elderly, young children, and expectant mothers. Falls are the second most common cause of unintentional injury deaths globally, according to the World Health Organization (WHO) (Qi et al., 2023). Data shows that 50% of people over the age of 81 and 30% of people over the age of 65 encounter falls annually, which could be harmful. Falls account for over 45% of all nursing home admissions due to the high morbidity rate (Kulurkar et al., 2023). The likelihood of falls increases exponentially, as a result of age-related biological changes, which contributes to a high frequency of falls and fall-related injuries in ageing populations. The number of injuries brought on by falls is anticipated to increase by 100% by 2030 if preventive measures are not immediately implemented (Igual et al., 2013a). Fall detection is crucial to lowering the risk of consequences since it enables early management and minimizes further health issues (Denkovski et al., 2023). Therefore, accurate and timely identification is important in these situations. Furthermore, long-term studies have correctly recognized falls as a risk to older people's health (Baik & Shin, 2024).

Fall detection approaches have advanced significantly over time by utilizing a variety of sensor modalities and integrating cutting-edge machine learning algorithms and signal processing (Gong et al., 2023). Depending on the detection tools used, current fall detection research can be separated into studies on contact sensors and non-contact sensors (Baik & Shin, 2024). The three main methods are vision-based systems, ambient sensors, and wearable sensors. Accelerometers and gyroscopes are two common wearable sensors that use motion pattern analysis to identify falls (Maray et al., 2023). In addition to being costly, infrared and camera-based sensors have problems with patient privacy and the stationary nature of the system because they capture audiovisual signals. Because of these drawbacks, wearable sensor-based fall detection provides a less expensive way to identify falls based on one or more wearable sensors that are affixed to the user's body or clothing so they can be carried around (Yhdego et al., 2023). The health status of elderly patients using motion-assistive devices is frequently monitored by analysing signals from wearable sensors installed on the body (Bright & Coventry, 2013). Usually, these sensors produce complicated hip motion data that are challenging to decipher without professional assistance. For the sensor data to be meaningfully characterized, a computationally effective fall-detection modeling method is needed. The research community has been drawn to machine learning and deep learning techniques for human fall recognition, primarily because of their increased accuracy. Nonetheless, there are still certain drawbacks (Usmani et al., 2021), primarily related to their model training procedure and the processing power required (Moutsis et al., 2023).

Fall detection techniques use threshold-based, machine learning-based, or deep learning-based methods to analyze motion data and differentiate between falls and Activity of Daily Living (ADLs) (J. Liu et al., 2023). Threshold-based techniques differentiate between falls and ADLs using cut-off values applied to sensor inputs (Blunda et al., 2020), (Ferreira De Sousa et al., 2022). Nevertheless, "false alarm fatigue" may result from these methods' frequently high false alarm rates (Rastogi & Singh, 2021). However, it is difficult to model the various elements that contribute to falls using machine learning-based techniques like traditional classifiers using manually created features (Le et al., 2022), (Son et al., 2022). These variables may be situational (activities causing the fall), external (lighting, obstructions), or internal (age, limited mobility, illness). Fall feature extraction is very difficult and frequently necessitates a multidisciplinary approach. Interest in the potential uses of artificial neural network in biology and medicine has increased as computer technology has advanced dramatically (Patel & Goyal, 2007).

1.2 Statement of the problem

The most common outcomes of fall injuries are fractures, loss of independence, and even death. As people age, their bodies undergo many physical changes that make them more vulnerable to falls. One of the most important factors influencing the severity of a fall is timing; elderly fallers often cannot get up on their own, and prolonged lying down can cause dehydration, hypothermia, pressure sores, and bronchopneumonia. Over 20% of patients admitted to hospitals after falling had been on the ground for an hour or more, increasing their morbidity rates within six months. Previous research that used traditional machine learning, acoustic arrays, and video surveillance systems for fall detection encountered a number of issues, including misclassification, overfitting, and a shortage of data samples. Therefore, a high precision fall detection model needs to be created in order to solve the issues of accuracy, f1 score, recall (misclassification), data sample size, and overfitting using deep learning.

1.3 Purpose of the study

By utilizing real-world datasets and optimizing model performance for accuracy, responsiveness, and energy efficiency, the proposed approach aims to improve the reliability of fall detection systems, guaranteeing timely alerts and enhanced safety for vulnerable individuals across diverse environments. The goal of this study is to develop a robust fall detection system using advanced time-series analysis and deep neural network techniques.

1.4 Research questions

In other to have a clearer and comprehensive view of the research, the objectives are centered on the following questions:

1. How do existing fall detection model differentiate fall from normal daily living activities to limit the rate of false positive?
2. What are the measures to consider in building a model that can appropriately distinguish fall from normal daily living?
3. Will the suggested approach be sufficient to address the issue raised in the current model for fall detection and daily activities?

1.5 Aim and Objectives

The aim of this research is to develop an intelligent IoT-driven and more efficient Deep Learning model for fall detection in the Elderly. The specific objectives of this research work are :

I. To design and fine-tune machine learning models (Random Forest, CNN-LSTM, and Bi-LSTM) for fall detection.
II. To implement the designed Bi-directional LSTM model.
III. To evaluate the performance of the designed model using accuracy, precision, F1-score and recall.

1.6 Scope and limitation of study

This study's scope is restricted to enhancing the performance of the state-of-the-art fall detection model by utilizing several dataset modalities to increase the model's generalizability.

1.7 Organization of the study

There are five chapters in this study report. An outline of this research is given in the first chapter. It highlights the importance and application of the problem at hand while providing a brief synopsis. The problem statement, research question, investigation's goal and goals, study importance, and study scope are all covered in greater detail in this chapter. Chapter 2 states that this study's technical and scholarly literature is critically examined, and the current research is assessed considering previous investigations. The main topics of the third chapter are strategy and design, the study's models, data preprocessing, and data collection techniques are covered in this chapter; implementation, analysis, and results are covered in Chapter 4. Chapter 5 concludes and offers suggestions for additional research. Additional sources, tables, and pictures are available in this chapter.

1.8 Definition of terms

i. Wearable devices: Devices that are worn on clothing or the human body are referred to as wearable devices. They are made up of a transducer and a target receptor. After identifying the target analyte, a receptor reacts appropriately. The receptor's reaction is subsequently transformed into a usable signal by the transducer (Iqbal et al., 2021).
ii. Gyroscope: Gyroscopes are sensors that can measure a body's angular velocity around one or more axes about an inertial reference frame. They come in varying degrees of precision (Dell'Olio et al., 2023).
iii. Barometer: This is a sensory device that measures changes in altitude and atmospheric pressure (Venkateswaran et al., 2024) .
iv. Artificial Intelligence: Artificial Intelligence, also known as AI is the simulation of human intelligence processes by machines,

especially computer systems (Zhai et al., 2021).

v. Machine learning: ML is a subfield of AI that allows for computers to learn from a given set of data and then make useful prediction predictions. These processes can be classified as supervised and unsupervised learning (El Naqa & Murphy, 2015).

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction to fall

Public health, technology, and medicine have all advanced in the last few decades, and people's understanding of education, nutrition, and health has increased. Because of this, life expectancy has increased globally, and the global population is getting older (Zahedian-Nasab et al., 2021). Falling is one issue that comes with living a longer life, frequent but frequently disregarded courses of harm around the world, lower quality of life, soft tissue injuries, higher mortality, chronic pain, fractures, excessive healthcare costs, functional impairment are all consequences of falls (Karlsson et al., 2013). Fall and bone fragility are two of the most significant risk factors for fractures Pasquetti, (2014), while the two main risk factors for falls in older adults are fear of falling and poor balance (Fabre et al., 2010). Every year, about one-third of adults in the community who are 65 years of age or older and 50–60% of people who live in nursing homes and assisted living facilities fall with likelihood of women falling compared to men (Karlsson et al., 2013). However, there's always a chance that someone could shatter their bones in a fall. Furthermore, it might make the individual feel less independent, more reclusive, and self-conscious (NHS, 2021). The development of fall detection systems is urgently needed due to the aging population. According to Mubashir et al., (2013), there are three main approaches for fall detection systems, depending on whether the data is gathered using vision devices. sensors that are wearable or ambient. Nonetheless, Igual et al., (2013b), separated fall detection into wearable technology and context-aware technologies.

2.1.1 Wearable devices

Wearables offer chances to enhance quality of life in a way that is difficult to accomplish with smartphones alone since they can sense, gather, and upload physiological data around-the-clock (Seneviratne et al., 2017). By lessening the burden on hospitals and delivering more accurate and fast information, wearables have completely transformed the healthcare sector (Iqbal et al., 2021). Gyroscopes, magnetometer, barometer and accelerometers wearables are used in this kind of fall detection classification (Igual et al., 2013a). Wearable technology can be applied to various bodily parts, such as the head, eyes, and wrist. Sensors built into smartphones, smart watches, and other portable devices have lately gained popularity in fall detection systems due to their low cost and broad use. In this study, where we highlight the importance of using various modalities of dataset via sensors for better model generalizability. The modalities of sensors proposed in the study include accelerometer, Gyroscopes, magnetometer, and barometer. Thus, we will discuss briefly about the different sensors.

2.1.1.1 Gyroscopes in fall detection

The investigation of new monitoring systems capable of automatically alerting about falls has attracted a lot of scientific interest over the past ten years due to the significant impact that falls have on the quality of life of the elderly and the financial sustainability of health systems (Casilari et al., 2020). A common sensor for identifying body movements is the gyroscope. A two-axis gyroscope can be affixed to the user's right arm, waist, and chest. By examining the body and thigh angel speeds, they can be able to identify falls (Hsieh et al., 2014).

2.1.1.2 Accelerometer in fall detection

An accelerometer is a type of electromechanical device used to monitor accelerating forces. Acceleration is a vector quantity that characterizes a change in velocity. Three acceleration sensors are arranged orthogonally in a tri-axial accelerometer to gather three-dimensional (x-, y-, and z-axis) data (Chapa et al., 2020). Movement stresses the tiny crystals inside the accelerometer, producing a voltage. The sensor determines the direction and speed of the movement by interpreting the voltage size (Yuan et al., 2023). Uniaxial accelerometers only include the vertical axis while measuring acceleration, whereas triaxial accelerometers effectively combine three single-axis accelerometers in three Euclidean axes (Yang et al., 2021). They are useful for quickly determining the condition of the thing they are connected to. The best perspective of the overall health of the connected object is obtained by keeping an eye on all three axes, particularly at both ends. There is little doubt that information from a single measurement axis can shed light on newly discovered machine flaws and their seriousness.

2.1.1.3 Barometer in fall detection

One of the earliest sensors ever designed is a barometer (Shoaib et al., 2014). In the past, their main applications have been as altimeters for aircraft or as environmental sensors to gauge atmospheric pressure for weather forecasts (Lara & Labrador, 2013). A wide range of novel barometer applications evolved with the introduction of microelectromechanical system (MEMS)-based barometers and its methodical integration into wearables and smartphones (Manivannan et al., 2020).

2.1.2 Context-aware systems

This kind of fall detection classification makes use of environment-deployed sensors. The employment of vision-based devices and environmental sensors such as radar, floor, microphone, pressure, and infrared sensors are some examples. Kinect, cameras, and motion capture devices are examples of context-aware systems Igual et al., (2013b).

2.2 Artificial Neural Network (ANN)

An approach for dealing with challenging pattern-oriented problems of the classification and time-series types is provided by ANN. (Walczak & Cerpa, 2019). As opposed to commonly used parametric statistical methods, neural networks' nonparametric nature allows for the creation of models without the need for prior knowledge of data population distribution or any possible relationships among variables.

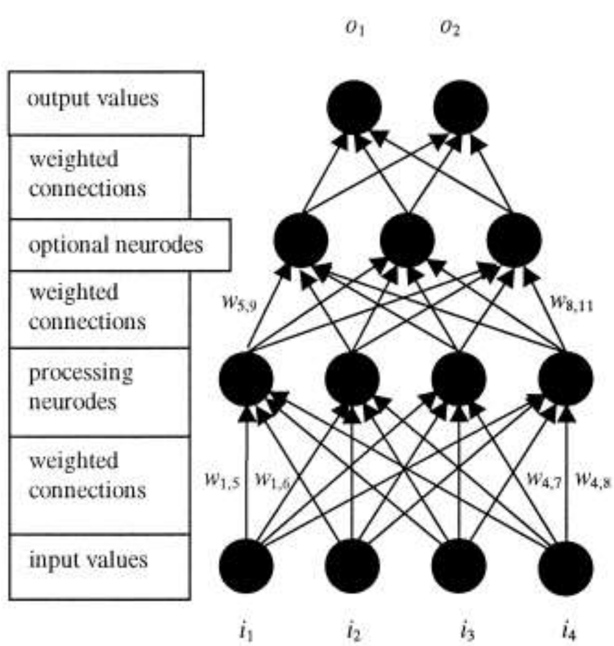

Figure 2. 1 Artificial Neural Network pictorial representation (Walczak & Cerpa, 2019)

2.3 Deep Learning

Machine learning includes a subset called Deep Learning (DL). By learning from enormous volumes of data, DL uses multiple-layer artificial neural networks to precisely replicate how the human brain functions (Bruce Ho, 2016). DL differs from traditional ML in respect to the type of data it uses coupled with how it also learns it. DL eliminates some of the pre-processing that is generally done in ML. DL can automatically ingest, process, and extract characteristics from unstructured data like photographs and text without the need for human professionals (IBM, n.d.). Computer models with several processing layers can learn representations of data with various levels of abstraction with the aid of deep learning. In a variety of domains, such as visual object identification, speech recognition, drug discovery, object detection, and genomics, these techniques have considerably improved the state-of-the-art (LeCun et al., 2015).

2.4 Related works

The detection of fall has been the subject of various studies in the past. These studies include:

A fall detection system was developed in C.-L. Liu et al., (2010), with commercially available technologies. The k-Nearest Neighbor (KNN) classification approach was used to classify the postures based on the ratio and difference of the bounding boxes for the height and width of the human body silhouettes. A fall incident detection system that may be used to identify fall incident events is built using the critical time difference and the KNN classifier. The experiment's findings suggest that it might mitigate the effects of upper-limb movements, and the system's accuracy rate for identifying falls and lying-down incidents is 84.4%. The system's limitation to fall incidents and one everyday activity is a disadvantage.

A threshold-based approach for differentiating between falls and activities of daily living (ADL) is proposed in this study Bourke & Lyons, (2008). The array of fall detection sensors is based on a gyroscope. A bi-axial gyroscope sensor mounted on the trunk, which measures pitch and roll angular velocities, and a threshold-based algorithm were used to distinguish between ADLs performed by elderly subjects and simulated falls performed by young volunteers under supervision onto crash mats. Matlab was used for data analysis in order to calculate the angular accelerations, angular velocities, and trunk angle changes seen during eight distinct fall and ADL types. The findings indicate that, for a total data set of 480 movements, falls can be differentiated from ADL with 100% accuracy.

An address-event vision system that may detect inadvertent falls in senior home care situations was introduced in Fu et al., (2008). The system sounds an alarm when it detects a fall risk. An asynchronous temporal contrast vision sensor with sub-millisecond temporal resolution was used. Compared to a frame-based camera, the sensor transmits fall events with ten times higher temporal resolution and an 84% higher bandwidth efficiency. A straightforward method is used to calculate an instantaneous velocity vector, and fall events are recorded. It was possible to distinguish between fall events and common human behaviors including walking, kneeling, and sitting. Accurate findings can be obtained regardless of the observed person's spatial position in the room or the presence of pets.

An intelligent emergency response system was created in Lee & Mihailidis, (2005) to detect falls within the home. It uses image-focused sensors. Research involving 21 volunteers evaluated the fall-detection component of the system's functionality and efficacy. A mock-up bedroom featuring a bed, chair, and other typical bedroom furnishings was used for the trials. The individuals were told to take several poses. These scenarios must be performed three times in a random order by each participant. These test locations had 315 tasks in all, 126 of which featured fall simulation and 189 of which did not. The algorithm detected a fall in 77% of cases and failed to detect one in 23%. False alarms happened just 5% of the time. The results validate the potential use of a vision-based system to offer safety and security in the homes of senior citizens. However, since human life is at stake, the success rate percentage is extremely low and should be investigated.

A real-time implementation of the fall-detection system design was given in Hazelhoff et al., (2008).

The setup makes use of two fixed cameras that are perpendicular to each other. Principal component analysis was used to determine the direction of the body's main axis and the ratio of the variances in the X and Y directions for each object after the foreground regions of both cameras were eliminated. Using a Gaussian multi-frame classifier, the two aforementioned variables help identify fall events. Additionally, a head-tracking module was added to the system to improve its dependability by rejecting false positives. The efficiency and performance of the system were evaluated for several configurations. The outcomes show that it can identify falls in real time with over 85% accuracy.

In this study Moutsis et al., (2023), a pipeline that uses a threshold-based method to identify falls using information from a three-axis accelerometer was presented. In this manner, a low-complexity system that may be used with any acceleration sensor that gets data at various frequencies was suggested. Furthermore, the study identifies several falls in a time series of sum vector magnitudes, giving the precise time range of the fall, even when the input lengths may vary. This pipeline achieves outstanding performance outcomes on many datasets, with generated specificity of 93.96% and 85.90% and sensitivity of 90.40% and 91.56% on MMsys and KFall, respectively. The performance across different dataset varies with large extent thus questioning the generalizability of the developed model.

In Zaid Salah et al., (2022), a deep learning model for fall detection was created to be implemented on a microcontroller with a small amount of memory. Long range (LoRa) communication technology was used to deploy the microcontroller in a low-power wide-area network. The convolutional neural network (CNN), with 95.55% accuracy, has been found to be the most appropriate lightweight neural network when compared to conventional machine learning techniques. With 61.084 kilobytes of storage needed, the CNN model achieved inference speeds of less than 37.84 ms, suggesting the potential for real-time fall event detection with low-power microcontrollers. A very small model with very low complexity of two layers was used which could affect the performance of the model in generalization. With the small model parameter, there is high probability of the model to overfit.

An automated fall detection system that notifies the caregiver as soon as a fall is detected was used in Li et al., (2012). An acoustic-FADE was developed for this. Acoustic-FADE employs a circular array of microphones to record the sounds in a space. Upon hearing a sound, acoustic-FADE locates its source, amplifies the signal, and classifies it as "fall" or "non-fall." The guided response power with phase transform method, which has been shown to be dependable in loud environments and resistant to reverberation effects, is used to locate the sound source. The beamforming method enhances the signal by using the estimated position of the sound source. Height information is used to improve specificity. The Mel-frequency cepstral coefficient features produced from the enhanced signal are used during the classification process. Using a dataset of 120 falls and 120 non-falls performed by three stunt actors trained to appear elderly under varied environmental settings, the acoustic-FADE achieves 97.0% specificity and 100% sensitivity.

This study Ojetola et al., (2011), employs machine learning, specifically decision trees, to identify four different fall types: left, right, backward, and forward. The accelerometer and gyroscope-based system can distinguish between falls and activities of daily living (ADLs) with 92% recall and 81% precision when applied to experimental data from 8 male individuals. The suggested method's robustness and performance have been further examined in terms of how sensitive it is to the size of the training set and the physical profile of the subjects. However, the precision and accuracy of the fall detection model needs improvement. Thus, suggesting further research in the field of fall detection in elderly persons.

This study Aderinola et al., (2023) investigates a novel fall detection method that makes use of cutting-edge time series algorithms and actual fall data. The suggested approach has an effective runtime and does away with the necessity for human feature engineering. A sizable dataset of actual falls was adopted for the purpose of the study with false alarms and false negatives occurring as infrequently as once every three days demonstrating an average mean of 90.7%. Additionally, the study outperforms current techniques on the FallAllD and SisFall simulated fall datasets. But on FARSEEING dataset, the ResNet model perform better. However, the average F1 score achieved in this study is subjected to improvement as this is related to health where injuries, disabilities, and even fatalities can be experienced.

A single tri-axial accelerometer affixed to the patient's thigh is used by the patient-specific (PS) fall prediction and detection prototype system described in Saadeh et al., (2019), to distinguish between fall events and activities of daily living (ADL). The proposed system has two operating modes for fall prediction (FMFP and SMFD) in order to detect a fall occurrence. Based on the nonlinear support vector machine classifier, the FMFP algorithm identifies a fall risk event and notifies the patient by extracting seven differentiating properties for the pre-fall instance. FMFP has a specificity and sensitivity of 99.1% and 97.8%, whereas SMFD has a specificity and sensitivity of 99.3% and 98.6% for 600 evaluated cases of falls and ADL from 77 individuals.

A fall detection system based on Support Vector Machines (SVM) was introduced in Nguyen et al., (2016). During pre-processing, data from a trial-axis accelerometer system that was captured in different states was smoothed out using a mean filter. Furthermore, features were extracted from the filtered signals using Principal Component Analysis. The tests involve numerous trials with eight different states on a subject, and the results was processed to both identify falling and evaluate the effectiveness of the proposed technique. Falling states were assessed using the threshold and SVM techniques, and the results were compared to determine which strategy yielded the higher suggested performance. Therefore, the threshold approach with low-cost calculation produced a higher average accuracy of 96.3% when compared to the SVM.

In Stampfler et al., (2022), a threshold and machine learning based fall detection were presented. Six of the study's instances used threshold-based detection, while the remaining nine used machine learning techniques such decision trees, k-nearest neighbors, boosting, and neural networks. With reported sensitivities ranging from 60.4 to 99.3% and specificities from 74.6 to 100.0%, all techniques could eventually reach real-time detection. Lastly, several components of data science methodology, such as appropriate test sets for outcomes evaluation, were left out of the studies, raising concerns about whether claimed results would match performance in the real world. Also, performance ranging method of result presentation is a bit awful with the difference between the upper bound and lower bound accuracy value approximately 40%, thus not suitable for real world adoption.

The suggested model for fall detection in this paper by Abro & Jalal, (2024) makes use of multi-modal sensors. A bilateral filter, which is renowned for its monotonic and maximum flat magnitude response in the passband, was used to filter the inertial sensor data to provide smoothness. Important characteristics were obtained, including parseval's energy and the Gaussian Mixture Model (GMM). Additionally, RGB (red-green-blue) movies were used for fall detection, producing features including rectangles, triangles, full-body ridges, and full-body curves. Multimodal fusion was employed to integrate the final results from the inertial sensor and visual data. After optimizing the fused data using the Naive Bayes method, a Multi-layer Perceptron (MLP) classifier was trained for classification. The suggested approach was tested using the UR Fall Detection dataset, and its accuracy of 88% proved its value. The performance of the model needs to be improved.

In the study of Oliveira et al., (2024), an effective machine learning algorithm to identify falls in the elderly using accelerometer data from wearable technology was designed and developed. The study uses General Systems Theory to analyse and integrate data from accelerometers that are already present in wearable technology rather than to create new hardware. It focuses on how software algorithms and hardware data work together to improve the detection of falls in the elderly by interacting with human behaviour. This study achieves a recall of 97.92% using the Multilayer perceptron (MLP) model. Nevertheless, the MLP model is expensive to compute, prone to overfitting, and challenging to adjust hyperparameters.

Based on a thermal sensor and a supervised machine-learning algorithm, the work of Diaz -Ramirez et al., (2024), proposes a non-invasive fall detection system. A thermal sensor was used to collect room temperatures, and the corresponding data labelling was part of the experimental dataset created by students through simulations of both fall and non-fall occurrences. Three popular supervised machine learning models for fall event detection were assessed: a Random Forest, a Convolutional Neural Network, and a Support Vector Machine. The experimental findings show that these models regularly achieve performances above 95% across a range of evaluation measures, indicating their strong ability to differentiate between falls and non-fall events. The dataset used for the training of the model is very small which could affect the generalization of the model.

A vision-based fall detection that identifies falls among patients in emergency rooms was presented in this study by Alshawali et al., (2024). The accuracy of triage fall detection was improved when RF and DL algorithms were combined. 536 pictures of various body positions, including falling, made up the dataset. 94% accuracy, 91% specificity, and 94% sensitivity were attained by the statistical analysis. However, the dataset used in this study is very small which could hardly generalize in the detection of fall in elderly persons.

This study by Al Mudawi et al., (2024) main objective is to investigate the advantages of employing multimodal sensors to improve fall detection systems' accuracy. In order to extract characteristics, the proposed paper integrates skeleton-based modeling of depth sensors with time-frequency features of inertial sensors. A fusion technique is then used to merge these multimodal sensors. The resulting fused data is then subjected to optimization using a modified K-Ary classifier. On the UP-Fall Detection dataset and the UR-Fall Detection dataset, the proposed model's accuracy was 97.97% and 97.89%, respectively.

In this study Shaima R.M et al., (2023), we use machine learning before our gathered fall dataset from accelerometer sensors to identify whether a fall occurred or not. The Support Vector Machine (SVM), Decision Tree, and Naive Bayes supervised machine learning (ML) techniques are used to extract input features from the acceleration data. The findings demonstrate that fall detection accuracy for the Decision Tree, Naïve Bayes and SVM approaches 97%, 91%, and 95%, respectively, with no false alarms. This model lacks generalization as data from a single sensor was used in training of the models. Also, the feature engineering was done manually since a conventional machine learning model was used, however this is prone to error thus, could affect the performance of the developed model.

Wearable sensors were used in Althobaiti, Turke, Katsigiannis, Stamos, & Ramzan,(2020) to monitor individuals who are at risk of falling with minimal external involvement from nursing home or medical professionals. Accelerometer data from 35 healthy subjects were used in this study to record falls and different activities of daily living (ADL). The spatial and frequency domain parameters were obtained in order to train machine learning models to differentiate between falls and no fall events, as well as between falls and other ADLs. Supervised classification trials, which produced an F1-score of 98.41% for distinguishing between fall and no fall events and an F1-score of 88.11% for differentiating between distinct ADLs, demonstrated the efficacy of the proposed method. The performance of the model needs improvement as it can be seen in the F1-score achieved.

Table 2.1 Summary of literature review

S/N	Authors, (year)	Methodology/ Contributions	Results	Limitations
1.	Lee & Mihailidis, (2005)	It uses image-focused sensors. Research involving 21	The algorithm detected a fall in 77% of cases and failed to detect	The vision based is quite expensive and it could be affected by occlusion and there

		volunteers evaluated the fall-detection component of the system's functionality and efficacy.	one in 23%. False alarms happened just 5% of the time.	must not be a blind spot around the area where it is been used.
2.	Bourke & Lyons, (2008)	Matlab was used for data analysis to calculate the angular accelerations, angular velocities, and trunk angle changes seen during eight distinct fall and ADL types.	100% accuracy	There is high tendency that the model overfit considering the number of samples used for the training of the model.
3.	Hazelhoff et al. (2008)	The setup makes use of two fixed cameras that are perpendicular to each other. Principal component analysis was used to determine the direction of the body's main.	The outcomes show that it can identify falls in real time with over 85% accuracy.	The vision-based technology is highly costly, susceptible to occlusion, and requires that there be no blind spots in the vicinity of its application.

4.	Fu et al., (2008)	An asynchronous temporal contrast vision sensor with sub-millisecond temporal resolution was used.	Compared to a frame-based camera, the sensor transmits fall events with ten times higher temporal resolution and an 84% higher bandwidth efficiency.	The vision based is quite expensive and it could be affected by occlusion and there must not be a blind spot around the area where it is been used.
5.	Liu et al. (2010)	The KNN classification approach was used to classify the postures based on the ratio and difference of the bounding boxes for the height and width of the body silhouettes.	The system's accuracy rate for identifying falls and lying-down incidents is 84.4	The system's limitation to fall incidents and one everyday activity is a disadvantage.
6.	Ojetola el al., (2011)	Employs machine learning, specifically decision trees, to identify four different fall types: left, right, backward, and forward	92% recall and 81% precision.	However, the precision and accuracy of the fall detection model needs improvement. Thus, suggesting further research in the field of fall detection in elderly persons.

7.	Li el al., (2012)	Acoustic-FADE employs a circular array of microphones to record the sounds in a space. Upon hearing a sound, acoustic-FADE locates its source, amplifies the signal, and classifies it as "fall" or "non-fall.	The acoustic-FADE achieves 97.0% specificity and 100% sensitivity.	Using a dataset of 120 falls and 120 non-falls performed by three stunt actors trained to appear elderly under varied environmental settings. The dataset is small for adequate detection and generalization.
8.	Nguyen et al., (2016)	Data from a trial-axis accelerator system was smoothed out using a mean filter. While features were extracted from the filtered signals using Principal Component Analysis	An accuracy of 96.3% was achieved.	The very small dataset was used in this study.
9.	Fang el al., (2018)	Accelerometer data from 35 healthy subjects were used in this study to record falls and different activities of daily living (ADL).	An F1-score of 98.41% for distinguishing between fall and no fall events and an F1-score of 88.11% for differentiating	The performance of the model needs improvement as it can be seen in the F1-score achieved.

			between distinct ADLs was achieved.	
10.	Saadeh et al., (2019)	The proposed system has two operating modes for fall prediction (FMFP and SMFD) in order to detect a fall occurrence.	FMFP has a specificity and sensitivity of 99.1% and 97.8%, whereas SMFD has a specificity and sensitivity of 99.3% and 98.6%.	A total number of 100 associated falls and ADL which is very small for generalization.
11.	Althobaiti, Turke, Katsigiannis, Stamos, & Ramzan, (2020)	The spatial and frequency domain parameters were obtained in order to train machine learning models to differentiate between falls and no fall events.	An F1-score of 98.41% for distinguishing between fall and no fall events and an F1-score of 88.11% for differentiating between distinct ADLs.	The performance of the model needs improvement as it can be seen in the F1-score achieved.
12.	Stampfler *et al.* (2022)	A threshold and machine learning based fall detection were presented. Six of the study's instances used	Sensitivities ranging from 60.4 to 99.3% and specificities from 74.6 to 100.0%.	The difference between the upper bound and lower bound accuracy value approximately 40%, thus not suitable for real world adoption.

		threshold-based detection, while the remaining nine used machine learning techniques.		
13.	Stampfler *et al. (*2022)	A threshold and machine learning based fall detection were presented. Six of the study's instances used threshold-based detection, while the remaining nine used machine learning techniques.	Sensitivities ranging from 60.4 to 99.3% and specificities from 74.6% to 100%.	The difference between the upper bound and lower bound accuracy value approximately 40% , thus not suitable for real world adaptation.
14.	Aderinola et al., (2023)	A novel fall detection method that makes use of cutting-edge time series algorithms and actual fall data. The suggested approach has an effective runtime and does away with the necessity for human feature engineering.	Average F1 score of 90.7%.	The average F1 score achieved in this study is subjected to improvement as this is related to health where injuries, disabilities, and even fatalities can be experienced.

15.	Moutsis et al., (2023)	A pipeline that uses a threshold-based method to identify falls using information from a three-axis accelerometer was presented.	Specificity of 93.96% and 85.90% and sensitivity of 90.40% and 91.56% on MMsys and Kfall, respectively.	The performance across different dataset varies with large extent thus questioning the generalizability of the developed model.
16.	Alshalawi et al. (2024)	A vision-based fall detection that identifies falls among patients in emergency rooms was presented. The accuracy of triage fall detection was improved when RF and DL algorithms were combined.	94% accuracy, 91% specificity, and 94% sensitivity were attained by the statistical analysis.	However, the dataset used in this study is very small which could hardly generalize in the detection of fall in elderly persons.

17.	Abro & Jalal (2024)	A bilateral filter, which is renowned for its monotonic and maximum flat magnitude response in the passband, was used to filter the inertial sensor data to provide smoothness.	Accuracy of 88%	The performance of the model needs to be improved demonstrated by the accuracy.
18.	Oliveira et al., (2024)	An effective machine learning algorithm to identify falls in the elderly using accelerometer data from wearable technology was designed and developed.	A recall of 97.92% was achieved.	The MLP model is expensive to compute, prone to overfitting, and challenging to adjust hyperparameters.
19.	Ramirez et al., (2024)	Based on a thermal sensor and a supervised machine-learning algorithm, the work proposes a non-invasive fall detection system	The experimental findings show that these models regularly achieve performances above 95%.	The dataset used for the training of the model is very small which could affect the generalization of the model.

20.	Al Mudawi et al., (2024)	The proposed paper integrates skeleton-based modelling of depth sensors with time-frequency features of inertial sensors. A fusion technique is then used to merge these multimodal sensors	On the UP-Fall Detection dataset and the UR-Fall Detection dataset, the proposed model's accuracy was 97.97% and 97.89%, respectively.	The dataset used for training the model is very small.

2.6 Research gap

Even though fall detection research has advanced significantly, there are still a number of crucial gaps that need to be filled in order to achieve dependable, practical performance. Many current systems rely on threshold-based or vision-based techniques, which have limited flexibility to different situations, significant false alarm rates, and privacy issues. Although deep learning and machine learning techniques have increased accuracy, most of them rely on handcrafted features, tiny sample sizes, or simulated datasets, which leads to poor generalization to actual fall events. Additionally, many studies are inadequate for low-power wearable IoT devices because they are unable to efficiently integrate multimodal sensor data or optimize models for computational efficiency.

Additionally, the comparability and robustness of results are limited by the absence of standardized datasets and uniform validation techniques among investigations. In order to guarantee high accuracy, low latency, and adaptability across various user populations and environments, a lightweight, data-driven, and generalizable fall detection framework that makes use of real-world multimodal sensor data, sophisticated time-series modeling, and deep learning is essential.

CHAPTER THREE
RESEARCH METHODOLOGY

This section of the study outlines several strategies, techniques, and procedures that would be used to accomplish the predefined goals and objectives, including the design framework that acted as the research's compass. This section describes how falls identified using both traditional machine are learning models and artificial neural networks, which can replicate detection with high levels of accuracy, recall, F1 score, and precision. The description of the dataset, data pre-processing, model development, and model evaluation are all highlighted and explained. The Figure 3.1 show the design framework employed in this study.

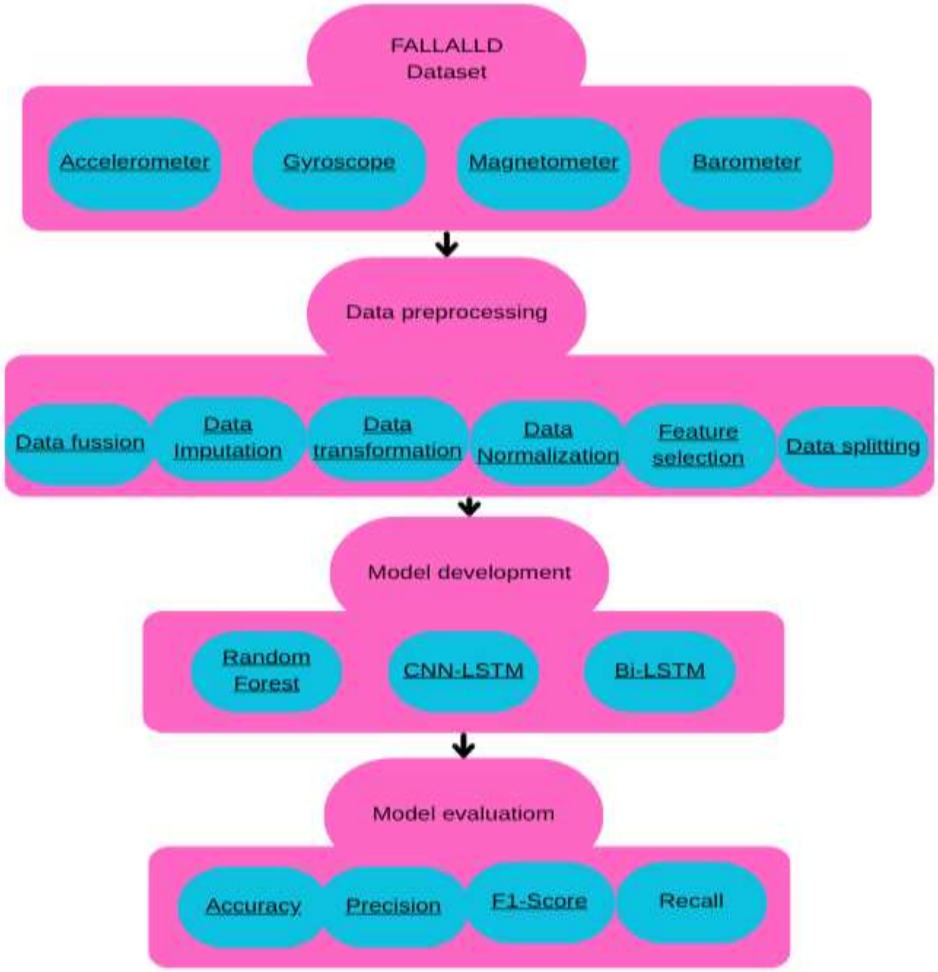

Figure 3.1 Fall Detection System Framework.

3.2: Dataset description

FallAllD is a sizable publicly available dataset of 15 people simulating everyday activities and human falls. In addition to targeting machine learning-based fall detection, which requires a larger number of data samples to prevent over-fitting and thereby enhance the generalization capability of trained machines, the dataset methodically covers all potential fall types. Finally, falls that are followed by recovery are regarded as positive samples rather than negative ones. This dataset is made up of 26420 files that were gathered using three data loggers that the subjects wore around their necks, wrists, and waists. A barometer, magnetometer, accelerometer, and gyroscope were used to record the motion signals to fit the possible applications in fall detection. This dataset gets around the shortcomings of the cutting-edge datasets. Activities of Daily Living (ADLs) and falls are covered in a wide range by FallAllD dataset with detectors worn at the neck, waist, and wrist are considered.

The link to the dataset is highlighted in Table 3.1 while Table 3.2. Highlighted and describe each feature present in the dataset.

Table 3. 1 Dataset acquisition repository

Dataset name/Repository	Link to the dataset
FallAllD: A Comprehensive Dataset of Human Falls and Activities of Daily Living	http://ieee-dataport.org/open-access/fallalld-comprehensive-dataset-human-falls-and-activities-daily-living

Table 3. 2 Dataset acquisition repository

S/no	Feature name	Description	Data type
1	SubjectID	This represents an individual used to carry out the data extraction.	Integer
2	Device	This identify the position in which the device is placed in the body.	String
3	ActivityID	This represents the unique identification for each activity present in the dataset.	Integers
4	TrialNo	This represents the number of times the experiment was confirmed.	Integer
5	Accelerometer	This represents the reading from the accelerometer device i.e. rate of change of velocity.	Object
6	Gyroscope	This represents the reading from the gyroscope device i.e. angular velocity.	Object
7	Magnetometer	This represents the reading from the magnetometer device i.e., the magnetic field.	Object
8	Barometer	This represents the reading from the barometer device i.e. atmospheric pressure.	Object

3.2 Data preprocessing

3.2.1 Data Fusion

The obtained secondary dataset was in .dat data format. There was a total of 26420 in total in which 6604 data samples were extracted from each of the devices, accelerometer, gyroscope, barometer, and magnetometer. The dataset was however fused together to represent a data point having readings from four devices thus giving more validity to the activity. The dataset was then converted into csv format for further processing to aid in the use of the dataset for model development. The csv pictorial representation of the dataset is displayed in Figure 3.1.

	SubjectID	Device	ActivityID	TrialNo	Acc	Gyr	Mag	Bar
0	1	Neck	13	1	[[4155, 55, -82], [4157, 55, -126], [4157, 50,...	[[1613, 205, 77], [1626, 202, 74], [1634, 199,...	[[3543, -780, 1995], [3537, -823, 1958], [3520...	[[1013.193859563175, 23.71569457530976], [1013...
1	1	Neck	13	2	[[4002, -50, 156], [4002, -38, 172], [4003, -2...	[[-57, 58, 18], [-64, 57, 15], [-66, 51, 16], ...	[[3701, 401, 1960], [3746, 365, 2026], [3816, ...	[[1013.214612231942, 24.11322125434876], [1013...
2	1	Neck	13	3	[[3983, 40, -335], [3984, 38, -324], [3987, 34...	[[1651, 208, 63], [1652, 203, 60], [1656, 198,...	[[3849, -680, 1685], [3834, -707, 1717], [3857...	[[1013.25438772214, 24.49373892784119], [1013....

Figure 3.2 CSV data representation for single person and event.

3.2.2 Data Imputation

Missing cell in the dataset causes training failure when attempting to build machine learning models. The dataset was checked against missing data, however no missing data was found in the dataset.

3.2.3 Label processing

The dataset label was saved in a different file named "activity_info.pkl". This file contains different fall types and ADLs which include the cyclic, transient phases of cyclic, transient and some other ADLs represented with numerical numbers from 1 to 135 where 1 to 44 represent different activity daily livings and 101 to 135 represent different fall scenarios while 44 to 100 were removed from the dataset as they represent unknown activities. The target column was created by representing the activities from 1 – 44 with 0 and 44 – 100 with 1. This is because we are dealing with two classification problem which involve ADL or fall scenarios.

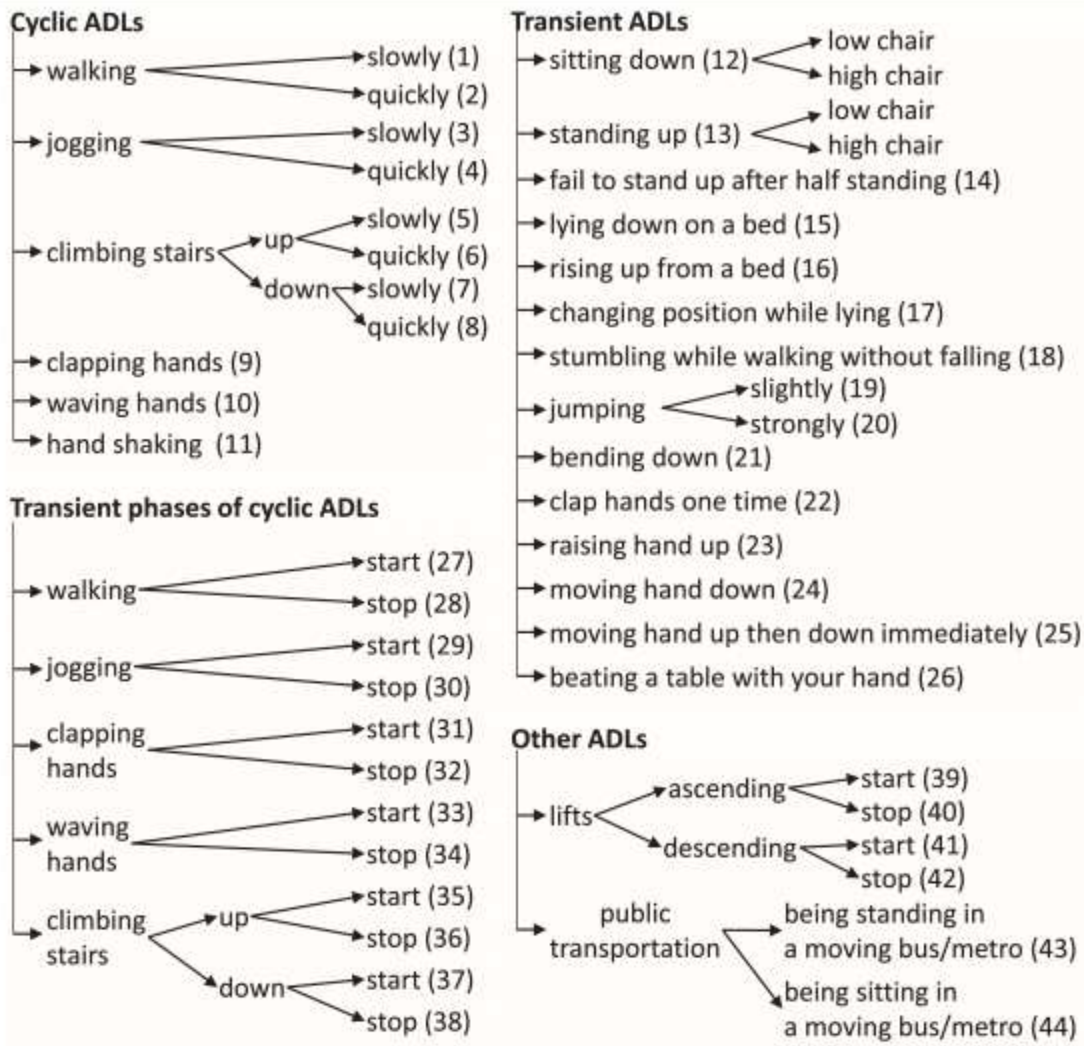

Figure 3.3 ADLs in the dataset (Saleh et al., 2021).

3.2.3 Data transformation

There were 8 features present in the dataset excluding the target column. These features have two different data types which include object and integer. The object columns in the dataset are then transformed into an integer as our dataset need to be vectorized to enable the training of machine learning model.

3.2.4 Data Normalization

Data normalization is one of the methods used for processing data in machine learning. The early convergence of machine learning models is very crucial to obtain an optimized model with fast training and inferencing ability. The columns values in the data are then normalized between 0 and 1 for faster computation to reduce time complexity.

3.2.5 Feature selection

Selection of the most important features are carried out during our data preprocessing. The subjectID and TrialNo column were dropped as these features does not add meaningful insight towards prediction of fall or activity of daily living. The ActivityID column was preprocessed to be dataset target by binarizing the integer values in the column.

3.2.6 Data Splitting

When creating machine learning models, data partitioning is crucial since it helps with training the model and evaluating its performance. As a result, we divided the data collection into training and testing datasets in a 7:3 ratio. However, while developing deep neural network models, the significance of validation cannot be diminished. As a result, we utilize a validation split of 0.2, meaning that during the training process, 20% of the data set designated for model training should be used for model validation. This would allow us to keep an eye out for over-fitting and under-fitting in the training model.

3.2.7 Model Architecture

Here, the model definition for abstract risk model were converted and created computer code for it. A variety of computer languages may be used for implementation, depending on the required IT Infrastructure, thus python programming language was employed. The Random Forest, Bi-LSTM and CNN-LSTM model was implemented.

3.2.7.1 Random forest

A randomly sampled random vector with the same distribution is used to build each tree in a random forest, creating an ensemble of tree-structured classifiers. Although the method is more resistant to overfitting, it produces error rates that are comparable to Adaboost. It is resistant to noise as well (Breiman, 2001). The Strong Law of Large Numbers prevents random forests from overfitting as the number of trees grows, guaranteeing the generalization error's convergence. The strength of each individual tree and the correlation among them determine how accurate a random forest is. To improve accuracy and decrease correlation across trees, a random selection of characteristics is used to build trees in the forest at each split. Internal estimates are used to track correlation, error, and strength offering information about the model's performance. All things considered, random forests are an effective ensemble learning technique that raises the accuracy of classification and regression by utilizing numerous random feature selection, internal error estimates, and decision trees (Breiman, 2001).

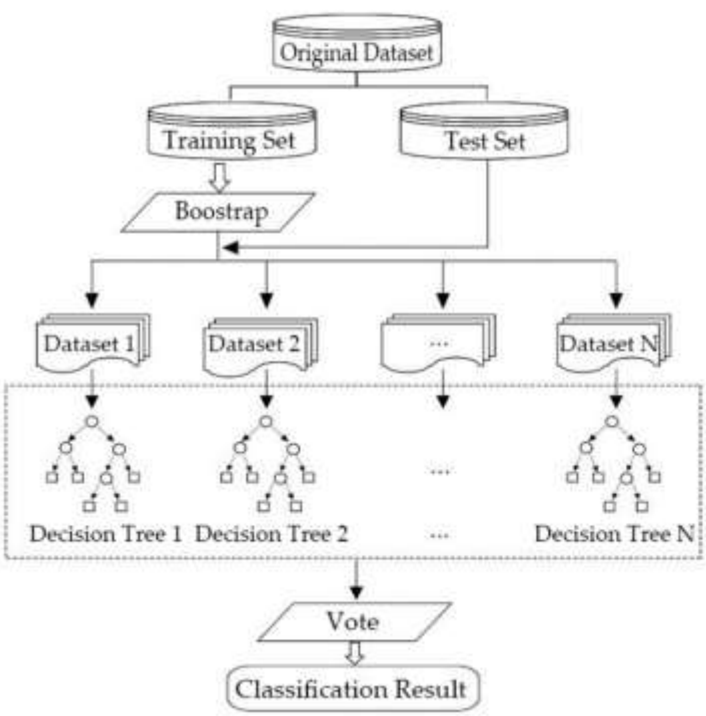

Figure 3.3 Random Forest model Architecture (Chen et al., 2019).

3.2.7.2 Bi-LSTM

A customized version of the LSTM architecture called Bidirectional Long Short-Term Memory (Bi-LSTM) processes input data twice: once in a forward pass (past-to-future) and once in a backward pass (future-to-past). This improves a model's capacity to learn long-term dependencies. The Bi-LSTM can more accurately represent a dataset's underlying context by combining data from both directions. Additionally, in order to manage the added complexity of adjusting its parameters, the Bi-LSTM usually analyses data in smaller batch chunks that are about half the size of those used by ordinary LSTMs because it must handle the data in both orders.

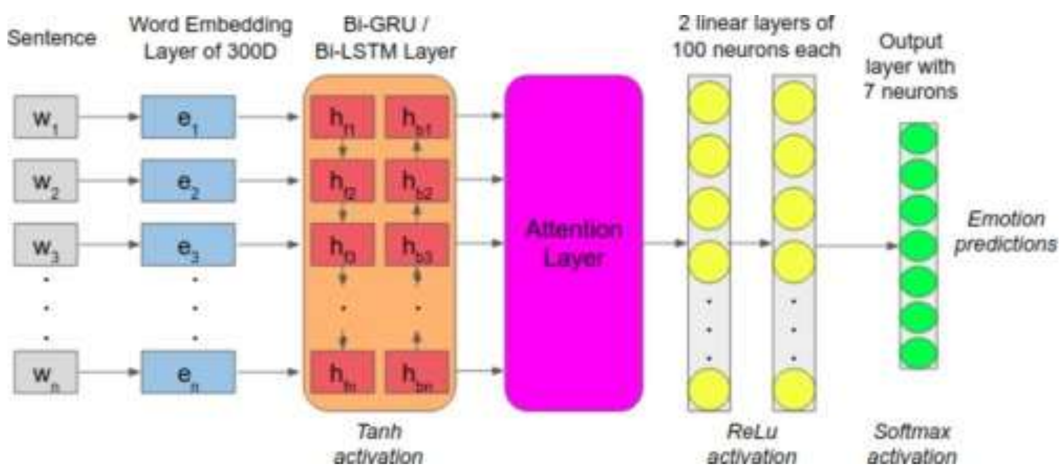

Figure 3.4 Bi-directional LSTM architecture (Ghosh et al., 2023)

3.2.7.3 CNN-LSTM

The suggested hybrid architecture combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to take advantage of both robust sequence learning and spatial feature extraction for fall or daily living activity prediction. To extract local signal patterns from multi-channel sensor data, a single CNN layer uses 64 one-dimensional convolutional filters with ReLU activation. This is followed by a one-dimensional max pooling layer, which lowers computational complexity while identifying task-dependent features. In order to identify long-term temporal correlations and hidden patterns in the data, these extracted features are subsequently processed sequentially by several LSTM layers using a specific gating system made up of input, forget, and output gates. The multivariate data is divided into fixed-length windows of 30-time cycles in order to maximize this sequence learning. To anticipate the final RUL value, the LSTM output is finally sent through dense (completely connected) layers to a neural network classification layer that uses a cross-entropy loss function.

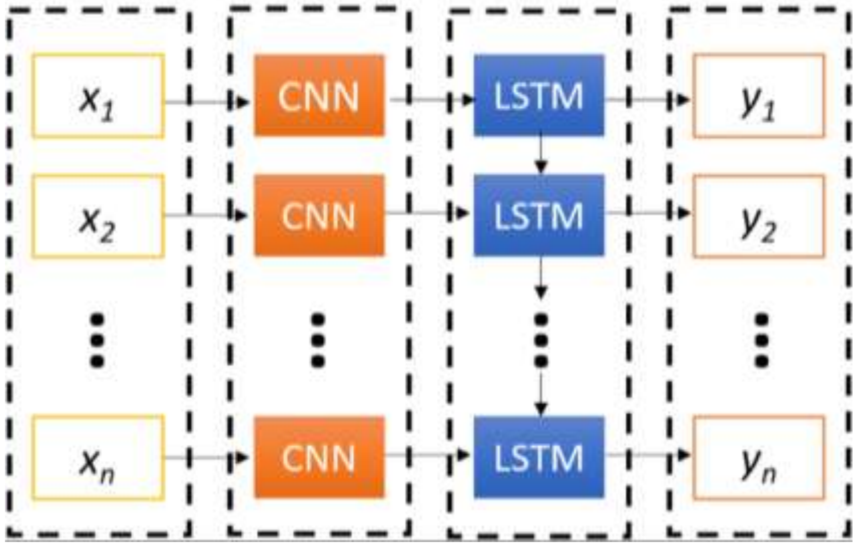

Figure 3.5 CNN-LSTM model architecture

3.2.8 Model Learning

The development of fall and ADL detection, as well as the use of the hyper-parameters, were discussed here.

3.2.8.1 SoftMax

We utilized the SoftMax function to the model's final fully connected layer because the problem at hand is categorization rather than regression. A vector of integers can be transformed into a vector of probabilities using the mathematical technique known as SoftMax, in which the likelihood of each value is inversely proportional to its relative size in the vector.

3.2.8.2 Loss function

The degree to which the algorithm's output has deviated from the desired result can be determined using a mathematical formula called the loss function. It is a method for assessing how accurately the algorithm captures the data. In this study, the difference between the produced and anticipated outputs was measured using categorical cross-entropy loss.

3.2.8.3 Optimizer

The goal is to lessen the loss that happens during learning. In a supervised learning context, the model receives data samples and associated results. When a model produces an output, it first compares it to the intended output and uses the difference to try to produce the generated output as closely to the intended output as feasible. An optimization function is employed to achieve this. However, we employed Adam optimization due to its benefits, which include dealing with sparse gradients, not requiring a stationary objective, insensitivity of parameter update magnitudes to gradient rescaling, and step sizes being roughly constrained by the step size hyperparameter.

3.2.8.4 Batch Size

The batch size is the quantity of dataset samples that will be input into the model simultaneously during a training cycle. The batch size affects the model's capacity to generalize. However, after extensive testing, a batch size of 32 was used to train the categorization model.

3.2.8.5 Learning rate

The rate of learning account for fluctuations in the predicted output. This hyperparameter determines how much the model weights will change each time they are updated during training. Choosing the appropriate learning rate can be difficult since a value that is too high could lead the training process to become unstable and continue until it hits the local minimum. Therefore, after much fine-tuning, we found that a learning rate of 0.002 was ideal for the model's first training before making any additional changes.

3.2.8.6 Epoch

An epoch is the total number of iterations applied during the training of a machine learning model using all the training data. Inappropriate epoch selection may result in either an extremely good or poor fit. In this case, the model training method included 30 epochs.

3.3 Performance metrics

Once the model has been trained with the appropriate hyperparameters to accelerate the pace of convergence, it must be evaluated using recognized evaluation criteria. Four primary performance metrics which include Recall, Precision, Accuracy, and F1 Score were employed in this research study.

3.3.1 Recall

The percentage of correctly forecasted favorable outcomes that occurred. False negatives cases that ought to have been marked for inclusion but weren't taken into account Calculating recall is as follows: The proportion of favourable outcomes that were accurately predicted and actually happened. Cases that should have been tagged for inclusion but weren't are considered false negatives (Karabiber, 2022). Recall can be calculated as follows:

3.3.2 Precision

Precision may be determined as; shows the proportion of optimistic forecasts that came true. Cases that were incorrectly identified for inclusion, or false positives, are considered (Karabiber, 2022). Precision can be calculated as:

3.3.3 Accuracy

A model's accuracy metric can be used to gauge its performance in every class. It is beneficial when every class is given equal weight. The ratio of accurately anticipated occurrences to all predicted events is referred to as accuracy (Gad, 2020). It can be calculated using the formular below:

3.3.4 F1 Score

This measurement is a harmonic average of precision and recall levels.

3.4 Hardware Resources and Specification

Since hundreds of data points are employed in our work, the GPU package is recommended for the study since it requires numerous computations, from feature extraction to loss reduction to the bare minimum, while guaranteeing that the model's weight values converge. To conduct the studies, Google Colab resources were employed. The Google Colab resources have the following specifications:

1. 13 Gig RAM
2. 12 Gig of GPU
3. NVIDIA Tesla K80

3.5 Software Resources

The model was implemented, tested, and visualized using the open-source Python programming languages. The Jupiter Notebook IDE was used for code implementation, Karas, TensorFlow, NumPy, pandas, scikit-learn, matplotlib libraries were used for data preprocessing, model implementation, training, validation and model testing.

3.6 Summary

We spoke about the research design and technique in this chapter. Since this is where the research's uniqueness rests, special attention was paid to the data description and preprocessing. Additionally, the model used were highlighted, the performance evaluation metrics, as well as the hardware and software requirement for the experiments.

CHAPTER FOUR
IMPLEMENTATION AND RESULT

4.1 Introduction

Before discussing the results, we discuss how our research concept and methodology were thoroughly implemented in this chapter. Lastly, as detailed in section 2.2 of chapter 2, we tabulated our data and compared them with results from relevant studies.

4.2 Implementation

The dataset used for this study, FallAllD is a sizable publicly available dataset of 15 people simulating everyday activities and human falls. This dataset contains data of about 26420 files in the .dat format. Each of these files contains the details of each of the experiment carried out during the data gathering. Each experiment consist of a user ID, which anonymize the information of the user, the activityID, which give a description of the activity carried out in the particular experiment, The TrailNo, giving the details of the number of times the particular experiment was carried out, the device gives the part of the body the device was wore, and each of the device name gives the sensor reading from each of the devices. We have 26420 files and each experiment was carried out four different times to allow reading from each of the sensor devices (Accelerometer, Gyroscope, Barometer, and Magnetometer). Thus dividing 26420 by 4 gives a total of 6605 rows as seen in our codebase. The preprocessing of files was done so that each experiment represents a row in the dataset and each sensor reading from the four devices under the same experiment represent the column for that particular experiment. The dataset was split using a ratio of 7:3 for training and testing of the model respectively. Then we proceeded with the development of our models which include LSTM, Bi-LSTM, and CNN-LSTM. The model was implemented using python programming language using deep learning framework, TensorFlow. All the implementation was carried out on google cloud platform known as Colab. The google colab provided us with the graphics processing unit and random-access memory needed to train our model and process our dataset respectively. Other python libraries used for the implementation of our study are NumPy and pandas which are for numerical computations as our sensors reading are in array as opposed to the single value data input attributed to CSV files. We preprocess this array of numbers into numerical values in tensors to enable the machine model to extract the hidden and important features present in the sensor's readings. Then we make use of matplotlib library for plotting of our training and validation outcome. After training our models we then implement the performance metric algorithm to enable us to evaluate the performance of the trained model. Lastly, we drew an inference from comparing the performance of the developed model to the existing model design for fall detection in elderly people.

4.2.1 Importation of Modules

All the model's and data necessary libraries are first imported at the beginning of the implementation as shown in Figure 4.2. Most of the libraries used to create models, modify hyper parameters, and train models come from TensorFlow and Scikit learn. The libraries used for data loading and preprocessing are the NumPy and Pandas while Matplotlib is used for plotting of model training and validation outcome.

```
import pandas as pd
import numpy as np
import pickle
#import tensorflow as tf
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.ticker import FuncFormatter
from itertools import cycle
import pickle
import sys
import time
from typing import List, Tuple
from imblearn.over_sampling import SMOTE

from sklearn import metrics
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import train_test_split
import sklearn.preprocessing
import tensorflow.keras
from tensorflow.keras.layers import Dense, Dropout, Activation
from tensorflow.keras.layers import LSTM, Bidirectional, BatchNormalization, Convolution1D, MaxPooling1D
from tensorflow.keras.models import Sequential
```

Figure 4.1 Code snippet for module importations.

4.2.2 Dataset Preprocessing

The secondary dataset that was acquired was in .dat data file format. 6604 data samples were taken from each of the accelerometer, gyroscope, barometer, and magnetometer devices, for a total of 26420. However, the dataset was combined to create a data point with readings from four devices, which increased the activity's validity.

	SubjectID	Device	ActivityID	TrialNo	Acc	Gyr	Mag	Bar
0	1	Neck	13	1	[[4155, 55, -62], [4157, 55, -126], [4157, 50...	[[1613, 205, 77], [1626, 202, 74], [1634, 199...	[[3543, -780, 1995], [3537, -823, 1958], [3530...	[[1013.15385963175, 23.71569457530976], [1013...
1	1	Neck	13	2	[[4002, -50, 156], [4002, -38, 172], [4003, -2...	[[-57, 58, 18], [-64, 57, 15], [-66, 51, 16], ...	[[3701, 401, 1960], [3746, 365, 2026], [3816, ...	[[1013.214612231942, 24.11322125434876], [1013...
2	1	Neck	13	3	[[3983, 40, -335], [3984, 38, -324], [3987, 34...	[[1651, 208, 63], [1652, 203, 60], [1656, 198...	[[3849, -680, 1685], [3834, -707, 1717], [3857...	[[1013.25408772214, 24.49373892784119], [1013...
3	1	Neck	13	4	[[3959, 165, 197], [3953, 165, 196], [3951, 16...	[[-70, 73, 2], [-67, 80, 4], [-70, 86, 5], [-7...	[[3565, -465, -187], [3543, -460, -176], [3553...	[[1013.167161107663, 24.72971366882324], [1013...

(a)

	acc0_x	acc0_y	acc0_z	acc1_x	acc1_y	acc1_z	acc2_x	acc2_y	acc2_z	acc3_x	...	bar195_2	bar196_1	bar
0	3895.0	-1134.0	1045.0	3894.0	-1133.0	1048.0	3891.0	-1131.0	1055.0	5463.0	...	29.075362	9.366342	29.0
1	-772.0	168.0	-1420.0	1011.0	346.0	-1610.0	1208.0	459.0	-1744.0	4137.0	...	31.112864	11.645918	31.1
2	857.0	599.0	665.0	860.0	594.0	651.0	863.0	582.0	649.0	453.0	...	26.319014	7.471224	26.3
3	4795.0	-696.0	-1299.0	4909.0	-795.0	-1314.0	5008.0	-845.0	-1261.0	4391.0	...	23.053419	12.311413	23.0
4	2440.0	3268.0	678.0	2432.0	3315.0	691.0	2427.0	3347.0	698.0	4117.0	...	26.052927	11.558382	26.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6600	3844.0	554.0	-1076.0	3841.0	550.0	-1075.0	3840.0	550.0	-1074.0	3843.0	...	29.756557	10.441573	29.7
6601	711.0	-3829.0	-1411.0	709.0	-3827.0	-1414.0	712.0	-3829.0	-1415.0	3955.0	...	27.584792	8.241062	27.5
6602	3709.0	-1839.0	-286.0	3705.0	-1837.0	-287.0	3698.0	-1834.0	-287.0	3961.0	...	28.486964	25.601689	28.4
6603	30.0	1134.0	638.0	63.0	1117.0	622.0	101.0	1100.0	630.0	485.0	...	27.530754	26.854329	27.5

(b)

Figure 4.2 (a) The array dataset as obtained from sensors reading (b) The preprocessed dataset for use in machine learning model

4.2.3 Model Implementation

As mentioned earlier, our model was implemented using TensorFlow library. The model was built with so many functions which include the LSTM layer. Convolutional layer, Bi-LSTM layer, pooling, and normalization. Figure 4.3 shows the screenshot of the codebase of our model implementation.

```
[ ]: batch_size = 32
     model = Sequential()
     model.add(Convolution1D(64, kernel_size=32, padding="same",
               activation="relu", input_shape=(455, 1)))
     model.add(MaxPooling1D(pool_size=(5)))
     model.add(BatchNormalization())
     model.add(Bidirectional(LSTM(64, return_sequences=False)))
     model.add(Reshape((128, 1), input_shape=(128, )))

     model.add(MaxPooling1D(pool_size=(5)))
     model.add(BatchNormalization())
     model.add(Bidirectional(LSTM(128, return_sequences=False)))

     model.add(Dropout(0.5))
     model.add(Dense(2))
     model.add(Activation('softmax'))
     model.compile(loss='categorical_crossentropy',
                   optimizer='adam', metrics=['accuracy'])

[ ]: model.summary(line_length=100)

     Model: "sequential_13"
     ____________________________________________________________________________________________________
     Layer (type)                                Output Shape                              Param #
     ====================================================================================================
     conv1d_10 (Conv1D)                          (None, 455, 64)                           2112
```

Figure 4.3 Code snippet for model implementation.

4.3 Results

In this section of the research studies, we analyze and explain the findings from our experiments which include the development of machine learning models to classify problem of the fall or no fall detection in elderly people. Three models were implemented which include the conventional machine learning model (Random Forest Classifier) and deep neural network (Bi-LSTM and CNN-LSTM).

4.4.1 Random Forest Classifier

The random forest conventional model was trained using our fall and activities of daily living dataset to detect when an older person falls or they are basically performing normal daily activities like sitting, sleeping etc. Figure 4.4 shows the classification report and confusion matrix obtained after testing the developed model.

	precision	recall	f1-score	support
0	0.89	0.89	0.89	2903
1	0.89	0.89	0.89	2957
accuracy			0.89	5860
macro avg	0.89	0.89	0.89	5860
weighted avg	0.89	0.89	0.89	5860

Figure 4.4 Classification report for Random Forest in fall/ADL detection

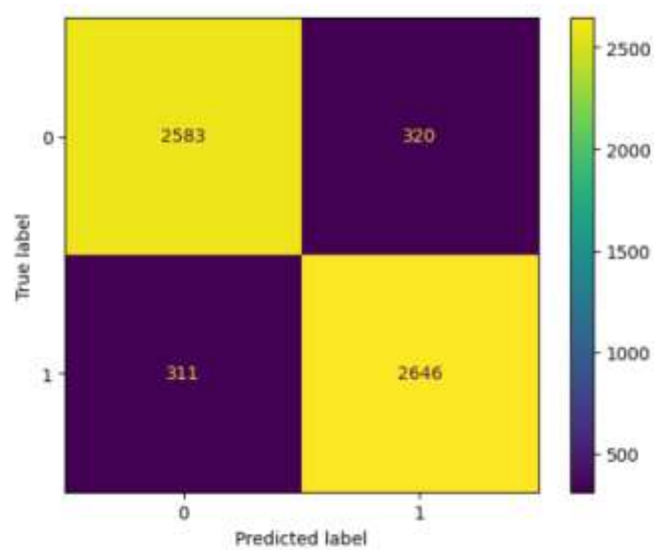

Figure 4.5 Confusion matrix for Random Forest in fall/ADL detection

From the result in Figure 4.4 and Figure 4.5 a total of 5860 samples was used to test the performance of the developed model, and the model was able to detect fall and ADL 89% of times. Since the data is from sensor devices such that it represents a time series data, it is quite understandable the level of performance reached by the conventional model, thus an indication that a robust model is needed to be able to detect the ADL and fall scenarios in a good number of times i.e. low False positive and False negatives.

4.4.2 Bi-LSTM

The Bi-directional Long Short-Term Memory was adopted for the development of a model that can adequately identify Falls and ADL scenarios with very low false positive and false negatives. Figure 4.6 depicts the training and validation performance of the Bi-LSTM model. From the plots, it can be seen that the training loss and validation loss progressively decrease, the same to the training and validation accuracy, thus ruling out case of model over-fitting during our model training.

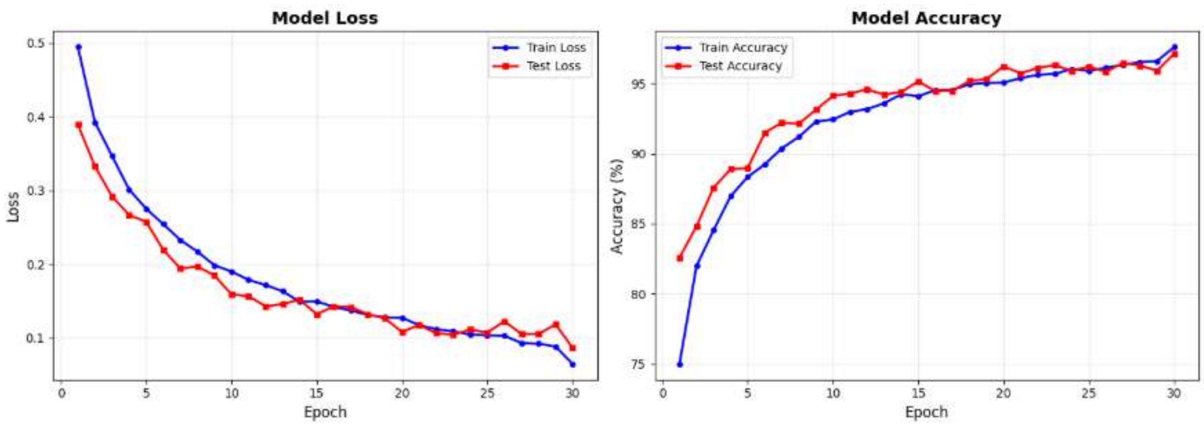

Figure 4.6 Loss and accuracy plot for Bi-LSTM model

Figure 4.7 and Figure 4.8 clearly show the confusion matrix and classification report obtained after testing our developed model with the state-of-the-art performance evaluation metrics. From the figures, it can be seen that the model was tested with 5851 number of samples, and it was able to detect Fall or ADL 97% of the times which clearly show an improvement when compared to the conventional machine learning model (Random Forest).

	precision	recall	f1-score	support
0	0.9731	0.9698	0.9715	2951
1	0.9694	0.9728	0.9711	2900
accuracy			0.9713	5851
macro avg	0.9713	0.9713	0.9713	5851
weighted avg	0.9713	0.9713	0.9713	5851

Figure 4.7 Classification report for Bi-LSTM in fall/ADL detection

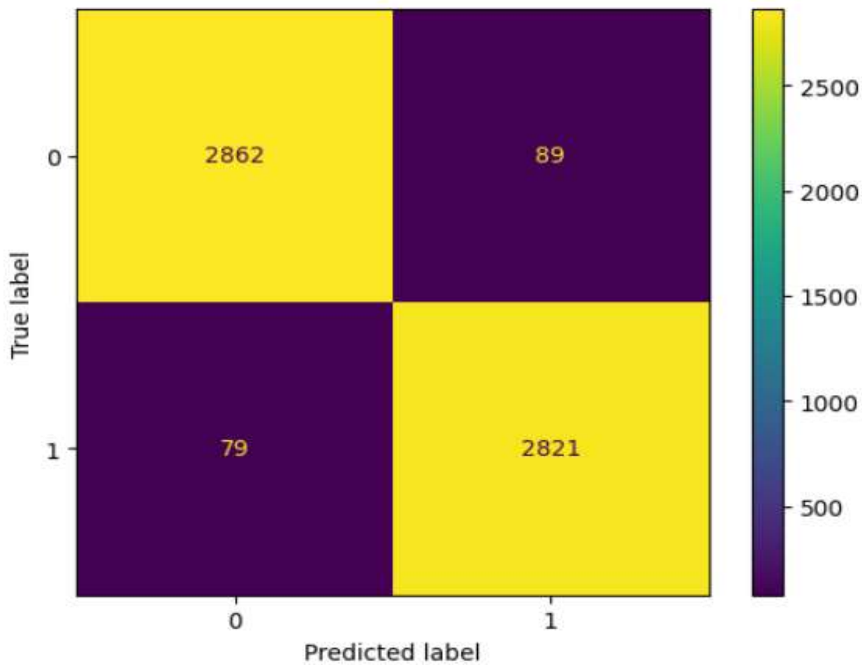

Figure 4.8 Confusion matrix for Bi-LSTM in fall/ADL detection

Also, in figure 4.9 we show the Receiver Operating Characteristics (ROC) curve which show the rate of the true positive to the false positive. As it is seen in the curve, the Area Under the Curve (AOC) is 0.99, thus demonstrating a good model performance.

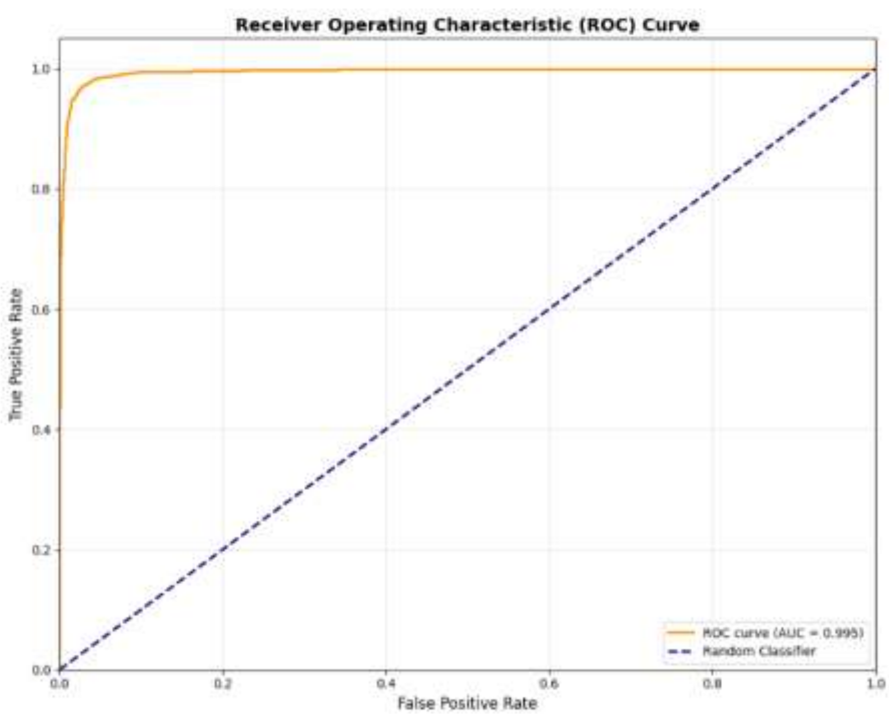

Figure 4.9 ROC-Curve for the Bi-LSTM model

4.4.3 CNN-LSTM

The Convolutional Neural Network - Long Short-Term Memory was adopted for the development of a model that can adequately identify Falls and ADL scenarios with very low false positive and false negatives. Figure 4.6 depicts the training and validation performance of the CNN-LSTM model. From the plots, the training loss and validation loss progressively decrease, the same to the training and validation accuracy, thus ruling out case of model over-fitting during our model training.

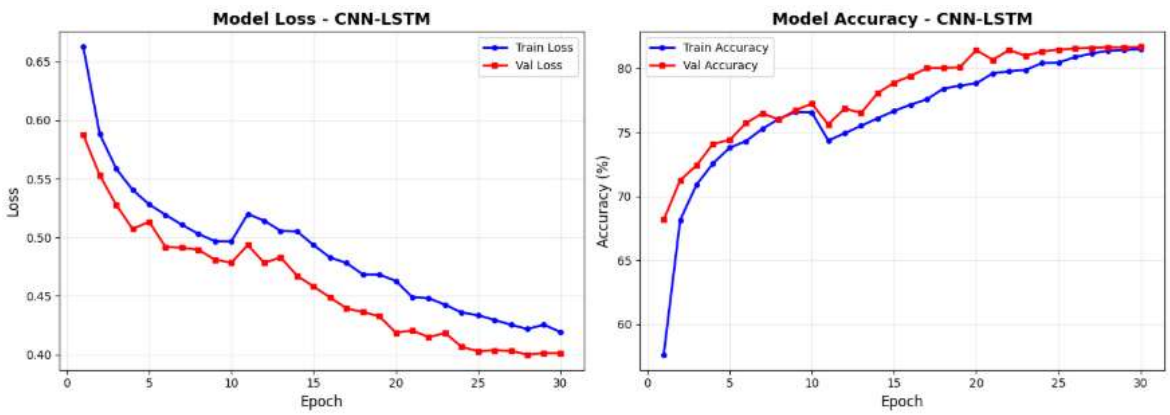

Figure 4.6 Loss and accuracy plot for CNN-LSTM model

Figure 4.7 and Figure 4.8 clearly show the confusion matrix and classification report obtained after testing our developed model with the state-of-the-art performance evaluation metrics. From the figures, it can be seen that the model was tested with 5851 number of samples, and it was able to detect Fall or ADL 82% of the time. However, the low performance obtained can be attributed to the use of convolutional neural network which is basically reserved for image dataset and in our case, we have vector data format that doesn't require convolution and pooling.

	precision	recall	f1-score	support
0	0.8399	0.8138	0.8267	3018
1	0.8073	0.8342	0.8206	2823
accuracy			0.8237	5841
macro avg	0.8236	0.8240	0.8236	5841
weighted avg	0.8242	0.8237	0.8237	5841

Figure 4.7 Classification report for CNN-LSTM in fall/ADL detection

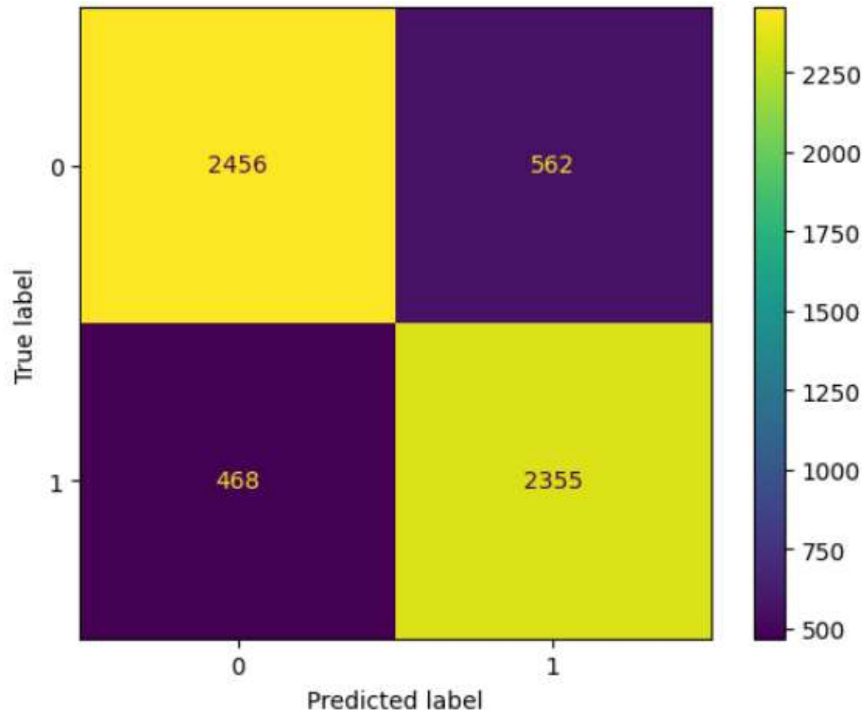

Figure 4.8 Confusion matrix for CNN-LSTM in fall/ADL detection

Also, in figure 4.9 we show the Receiver Operating Characteristics (ROC) curve which show the rate of the true positive to the false positive. As it is seen in the curve, the Area under the Curve (AOC) is 0.90, thus demonstrating a good model performance.

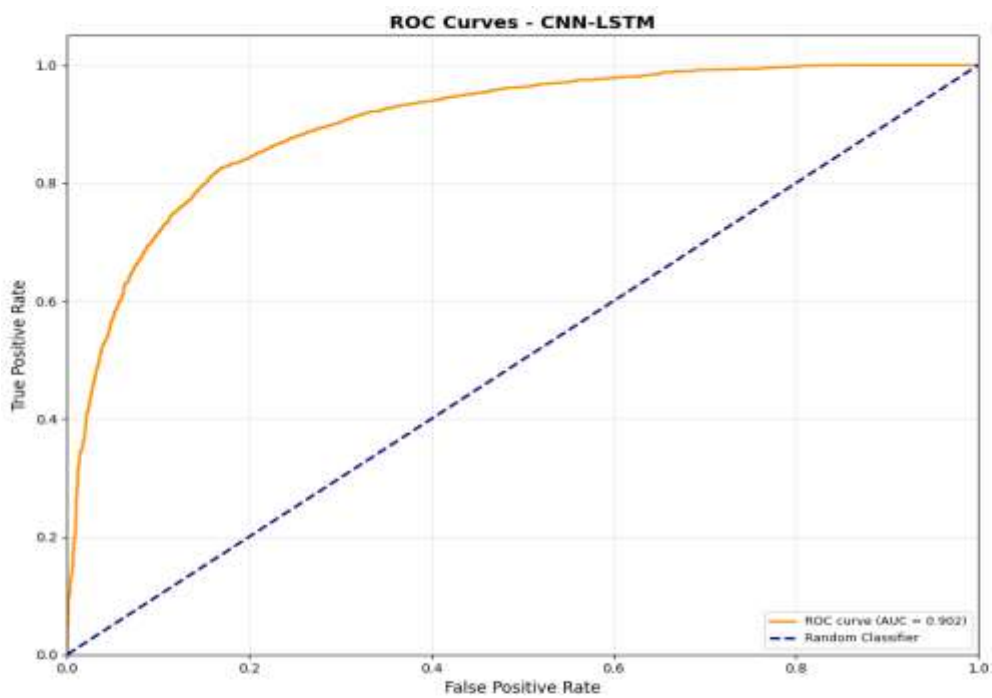

Figure 4.9 ROC-Curve for the CNN-LSTM model in fall/ADL detection

Figure 4.10 shows the summary of the results of our developed model (Random Forest, Bi-LSTM and CNN-LSTM) which includes the comparison of their accuracy, precision, F1-Score and recall.

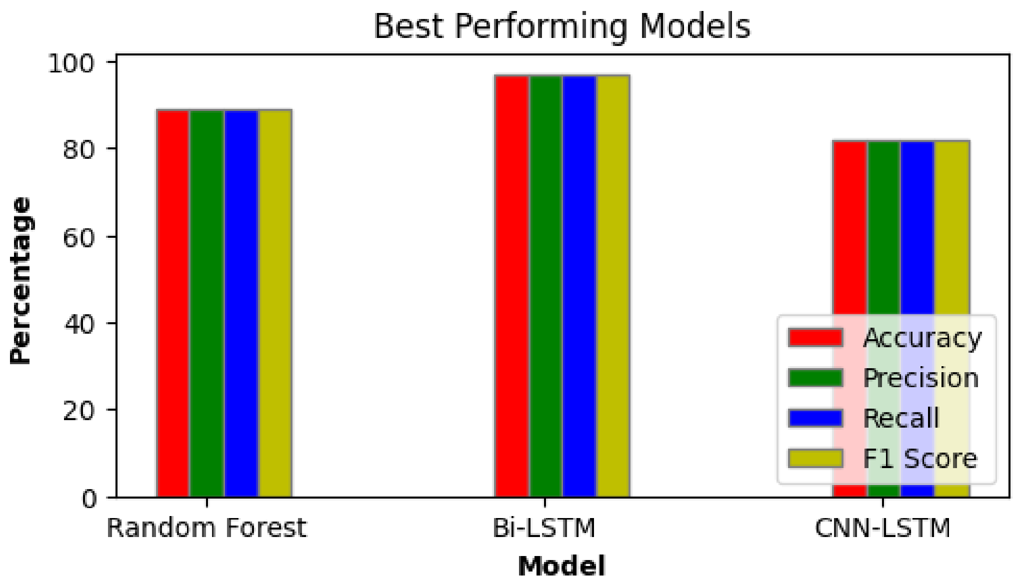

Figure 4.10 Performance comparison of the developed model in fall/ADL detection

4.4 Model Comparison

In this section we compare the outcome of our proposed methodology with the existing methods for the detection of ADL and fall in elderly people.

Table 4.1 Comparison of models for ADL and fall in elderly people

S/n	Paper	Accuracy (%)	F1 Score (%)	Precision (%)	Recall (%)
1	Ojetola et al., (2011)	-	-	81.00	92.00
2	Alshalawi et al., (2024).	94.00	91.00	94.00	-
3	Abro & Jalal, (2024)	88.00			
4	Oliveira et al., (2024)	-	-	-	97.92

5	Ramírez et al., (2024)	95.00	-	-	-
8	Random Forest	89.00	89.00	89.00	89.00
9	**Bi-LSTM (proposed)**	**97.00**	**97.00**	**97.00**	**97.00**
10	CNN-LSTM	82.00	82.00	82.00	82.00

4.5 Discussion of Results

It can be noted from Table 4.1 that our proposed model achieves the best performance when compared to other work from literature. In the work of Ojetola et al., (2011) a low precision of 81% and recall of 92% was achieved which shows the necessity of improve research in the field of fall detection. Also, Alshalawi et al., (2024) achieved F1 score of 91%, precision of 94% and accuracy of 94%, though the result performance can be high, but the model was trained and validated on small dataset, thus raising a question about the credibility of the model. Abro & Jalal, (2024) further work on fall detection led to a low accuracy of 88% and Oliveira et al., (2024) achieved a high recall value of 97.82 and Ramírez et al., (2024) with accuracy of 95% but a single dataset modality (Accelerometer data) was used to carry out their research, this means that the model will fail to perform when data from barometer, gyroscope and magnetometer are fed into the model for prediction. However, in our proposed study we achieved a good performance of about 97% for all the performance metrics while using four modalities of dataset to improve the model generalizability and robustness.

Also, we demonstrated the important of using the characteristics or features of dataset for model selection. From the study, three (3) different models were selected which include Random Forest, Bi-LSTM and CNN-LSTM. The Bi-LSTM model performed better because of its capability to learn and preprocess sequential data in both backward and forward directions as opposed to the Random Forest model adaptability to sequential data. The CNN-LSTM have the capability to learn from sequential data but its strength lies in image sequence data.

CHAPTER FIVE
CONCLUSION, LIMITATION AND RECOMMENDATION

5.1 Conclusion

As the world's population grows, life expectancy is also increasing dramatically, particularly in wealthy countries. The proportion and overall number of older people in the population are rising, posing significant difficulties to health care systems. Elderly individuals are falling more frequently as they get older and weaker, which makes it more difficult for them to always stay stable. According to data, 30% of those over 65 and 50% of those over 81 experience potentially dangerous falls every year. Because of the high morbidity rate, falls account for more than 45% of all nursing home admissions. Thus, this study aims to develop an intelligent and more efficient Deep Learning model for fall detection in elderly populations. In this study, a dataset made up of 26420 files that were gathered using three data loggers wore around the subject necks, wrists, and waists which include a barometer, magnetometer, accelerometer, and gyroscope motion signals to fit the possible applications in fall detection. This dataset was preprocessed which included data fusion, label processing, feature selection and transformation of data. Three (3) different models were selected which include Random Forest, Bi-LSTM and CNN-LSTM. The Bi-LSTM model performed better with 97% for all the performance metrics. This performance can be attributed to its capability to learn and preprocess sequential data in both backward and forward directions as opposed to the Random Forest model with a very low adaptability to sequential data. The CNN-LSTM have the capability to learn from sequential data, but its strength lies in image sequence dataset. This study can be adapted for the improvement in the field of fall detection.

5.2 Limitation

1. The period to carry out this research study was limited which restrained us to go beyond this novelty of the produced research work.
2. Limited access to very large datasets in the field of fall detection.

5.3 Recommendation

1. We also note that Bi-LSTM model extracts redundant features from dataset during training which can affect the overall performance of the model developed. It is recommended that more sophisticated models should be explored when carrying out future research.
2. To reduce the bias caused by the SMOTE data balancing technique, a new dataset with a sufficient number of samples should be sought for.

DEDICATION

To my parents, thank you for your endless support, love, and for pretending to understand what this project is about. Thank you for always asking if I'm "done yet" and for providing an endless supply of support to fuel my work.

To my siblings, As the last born, I've always had the advantage of watching you all make mistakes first so thank you for the life lessons! Your teasing toughened me up, your unsolicited advice guided me, and your leftovers nourished me.

To my friends, for distracting me just enough to keep me sane but not enough to ruin everything.

I couldn't have done it without you all literally, because you wouldn't let me work in peace.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to everyone who supported me throughout the course of this work. I'm especially thankful to my supervisor, family and friends.

LIST OF ABREVIATIONS

ML – Machine learning

CNN – Convolutional Neural Network

ADL – Activities of Daily Living

LSTM – Long Short-Term Memory

Bi-LSTM - Bi-directional Long Short-Term Memory

CNN – LSTM - Convolutional Neural Network - Long Short-Term Memory

SMOTE – Synthetic Minority Over Sampling Technique

References

Abro, I. A., & Jalal, A. (2024). Multi-Modal Sensors Fusion for Fall Detection and Action Recognition in Indoor Environment. *2024 3rd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, 1–6. https://doi.org/10.1109/ETECTE63967.2024.10823705

Aderinola, T. B., Palmerini, L., D'Ascanio, I., Chiari, L., Klenk, J., Becker, C., Caulfield, B., & Ifrim, G. (2023). *Accurate and Efficient Real-World Fall Detection Using Time Series Techniques*.

Al Mudawi, N., Batool, M., Alazeb, A., Alqahtani, Y., Almujally, N. A., Algarni, A., Jalal, A., & Liu, H. (2024). A robust multimodal detection system: Physical exercise monitoring in long-term care environments. *Frontiers in Bioengineering and Biotechnology*, *12*, 1398291. https://doi.org/10.3389/fbioe.2024.1398291

Alshalawi, R., Qadrouh, M., & Alhalabi, W. (2024). *The Effect of using Dimensionality Reduction Compared with Type of Algorithm on Detecting Patient Fall: Triage Case Study*. In Review. https://doi.org/10.21203/rs.3.rs-3950357/v1

Althobaiti, Turke, Katsigiannis, Stamos, & Ramzan, N. (2020). *ShimFall&ADL: Triaxial accelerometer fall and activities of daily living detection dataset*. https://doi.org/10.5281/zenodo.3901285

Baik, J.-Y., & Shin, H.-C. (2024). Fall Detection Using FMCW Radar to Reduce Detection Errors for the Elderly. *Journal of Electromagnetic Engineering and Science*, *24*(1), 78–88. https://doi.org/10.26866/jees.2024.1.r.207

Blunda, L. L., Gutierrez-Madronal, L., Wagner, M. F., & Medina-Bulo, I. (2020). A Wearable Fall Detection System Based on Body Area Networks. *IEEE Access*, *8*, 193060–193074. https://doi.org/10.1109/ACCESS.2020.3032497

Bourke, A. K., & Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics*, *30*(1), 84–90. https://doi.org/10.1016/j.medengphy.2006.12.001

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Bright, A. K., & Coventry, L. (2013). Assistive technology for older adults: Psychological and socio-emotional design requirements. *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, 1–4. https://doi.org/10.1145/2504335.2504344

Bruce Ho. (2016). Deep learning: Image & video recognition. *International Journal of Semantic Computing*, *10*(03), 417–439.

Casilari, E., Álvarez-Marco, M., & García-Lagos, F. (2020). A Study of the Use of Gyroscope Measurements in Wearable Fall Detection Systems. *Symmetry*, *12*(4), 649. https://doi.org/10.3390/sym12040649

Chapa, J. M., Maschat, K., Iwersen, M., Baumgartner, J., & Drillich, M. (2020). Accelerometer systems as tools for health and welfare assessment in cattle and pigs – A review. *Behavioural Processes*, *181*(June), 104262. https://doi.org/10.1016/j.beproc.2020.104262

Chen, J., Yu, J., Wen, J., Zhang, C., Yin, Z., Wu, J., & Yao, S. (2019). Pre-evacuation Time Estimation Based Emergency Evacuation Simulation in Urban Residential Communities. *International Journal of Environmental Research and Public Health*, *16*(23), 4599. https://doi.org/10.3390/ijerph16234599

Dell'Olio, F., Natale, T., Wang, Y.-C., & Hung, Y.-J. (2023). Miniaturization of Interferometric Optical Gyroscopes: A Review. *IEEE Sensors Journal*, *23*(24), 29948–29968. https://doi.org/10.1109/JSEN.2023.3327217

Denkovski, S., Khan, S. S., & Mihailidis, A. (2023). *Temporal Shift—Multi-Objective Loss Function for Improved Anomaly Fall Detection*. *222*.

Díaz-Ramírez, A., Diaz-Escobar, J., Quintero-Rosas, V., & Moncada-Sánchez, R. (2024). Classification of Fall Events in the Elderly Using a Thermal Sensor and Machine Learning Techniques. *Computación y Sistemas*, *28*(4). https://doi.org/10.13053/cys-28-4-4809

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology* (pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1

Fabre, J. M., Ellis, R., Kosma, M., & Wood, R. H. (2010). Falls risk factors and a compendium of falls risk screening instruments. *Journal of Geriatric Physical Therapy (2001)*, *33*(4), 184–197.

Ferreira De Sousa, F. A. S., Escriba, C., Avina Bravo, E. G., Brossa, V., Fourniols, J.-Y., & Rossi, C. (2022). Wearable Pre-Impact Fall Detection System Based on 3D Accelerometer and Subject's Height. *IEEE Sensors Journal*, *22*(2), 1738–1745. https://doi.org/10.1109/JSEN.2021.3131037

Fu, Z., Delbruck, T., Lichtsteiner, P., & Culurciello, E. (2008). An Address-Event Fall Detector for Assisted Living Applications. *Biomedical Circuits and Systems, IEEE Transactions On*, *2*, 88–96. https://doi.org/10.1109/TBCAS.2008.924448

Gad, A. F. (2020). *Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall*. https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/

Gaya-Morey, F. X., Manresa-Yee, C., & Buades-Rubio, J. M. (2024). Deep Learning for Computer Vision based Activity Recognition and Fall Detection of the Elderly: A Systematic Review. *Applied Intelligence*, *54*(19), 8982–9007. https://doi.org/10.1007/s10489-024-05645-1

Ghosh, S., Ekbal, A., & Bhattacharyya, P. (2023). Natural language processing and sentiment analysis: Perspectives from computational intelligence. In *Computational Intelligence Applications for Text and Sentiment Data Analysis* (pp. 17–47). Elsevier. https://doi.org/10.1016/B978-0-32-390535-0.00007-0

Gong, L., Mccall, S., & Yu, M. (2023). Enhancing privacy with optical element design for fall detection. *Electronics Letters*, *59*(20), e12995. https://doi.org/10.1049/ell2.12995

Hazelhoff, L., Han, J., & With, P. (2008). Video-Based Fall Detection in the Home Using Principal Component Analysis. *Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, 298–309. https://doi.org/10.1007/978-3-540-88458-3_27

Hsieh, S.-L., Chen, C.-C., Wu, S.-H., & Yue, T.-W. (2014). A wrist -worn fall detection system using accelerometers and gyroscopes. *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, 518–523. https://doi.org/10.1109/ICNSC.2014.6819680

IBM. (n.d.). *"What is deep learning?," IBM*. IBM Cloud Education,. Retrieved February 2, 2023, from https://www.ibm.com/cloud/learn/deep-learning.

Igual, R., Medrano, C., & Plaza, I. (2013a). Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, *12*(1), 66. https://doi.org/10.1186/1475-925X-12-66

Igual, R., Medrano, C., & Plaza, I. (2013b). Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, *12*(1), 66. https://doi.org/10.1186/1475-925X-12-66

Iqbal, S. M. A., Mahgoub, I., Du, E., Leavitt, M. A., & Asghar, W. (2021). Advances in healthcare wearable devices. *Npj Flexible Electronics*, *5*(1), 9. https://doi.org/10.1038/s41528-021-00107-x

Karabiber, F. (2022). *Precision and Recall*. https://www.learndatasci.com/glossary/precision-and-recall/

Karlsson, M. K., Magnusson, H., Von Schewelov, T., & Rosengren, B. E. (2013). Prevention of falls in the elderly—A review. *Osteoporosis International*, *24*(3), 747–762. https://doi.org/10.1007/s00198-012-2256-7

Kulurkar, P., Dixit, C. K., Bharathi, V. C., Monikavishnuvarthini, A., Dhakne, A., & Preethi, P. (2023). AI based elderly fall prediction system using wearable sensors: A smart home-care technology with IOT. *Measurement: Sensors*, *25*, 100614. https://doi.org/10.1016/j.measen.2022.100614

Lara, O. D., & Labrador, M. A. (2013). A Survey on Human Activity Recognition using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, *15*(3), 1192–1209. https://doi.org/10.1109/SURV.2012.110112.00192

Le, H.-L., Nguyen, D.-N., Nguyen, T.-H., & Nguyen, H.-N. (2022). A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System. *Electronics*, *11*(7), 1030. https://doi.org/10.3390/electronics11071030

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Lee, T., & Mihailidis, A. (2005). An Intelligent Emergency Response System: Preliminary Development and Testing of Automated Fall Detection. *Journal of Telemedicine and Telecare*, *11*, 194–198. https://doi.org/10.1258/1357633054068946

Li, Y., Ho, K. C., & Popescu, M. (2012). A Microphone Array System for Automatic Fall Detection. *IEEE Transactions on Biomedical Engineering*, *59*(5), 1291–1301. https://doi.org/10.1109/TBME.2012.2186449

Liu, C.-L., Lee, C.-H., & Lin, P.-M. (2010). A Fall Detection System using k-Nearest Neighbor Classifier. *Expert Syst. Appl.*, *37*, 7174–7181. https://doi.org/10.1016/j.eswa.2010.04.014

Liu, J., Li, X., Huang, S., Chao, R., Cao, Z., Wang, S., Wang, A., & Liu, L. (2023). A review of wearable sensors based fall-related recognition systems. *Engineering Applications of Artificial Intelligence*, *121*, 105993. https://doi.org/10.1016/j.engappai.2023.105993

Luo, Z., Jia, S., Niu, H., Zhao, Y., Zeng, X., & Dong, G. (2024). Elderly Fall Detection Algorithm Based on Improved YOLOv5s. *Information Technology and Control*, *53*(2), 601–618. https://doi.org/10.5755/j01.itc.53.2.36336

Manivannan, A., Chin, W. C. B., Barrat, A., & Bouffanais, R. (2020). On the Challenges and Potential of Using Barometric Sensors to Track Human Activity. *Sensors*, *20*(23), 6786. https://doi.org/10.3390/s20236786

Maray, N., Ngu, A. H., Ni, J., Debnath, M., & Wang, L. (2023). Transfer Learning on Small Datasets for Improved Fall Detection. *Sensors*, *23*(3), 1105. https://doi.org/10.3390/s23031105

Moutsis, S. N., Tsintotas, K. A., & Gasteratos, A. (2023). PIPTO: Precise Inertial-Based Pipeline for Threshold-Based Fall Detection Using Three-Axis Accelerometers. *Sensors*, *23*(18), 7951. https://doi.org/10.3390/s23187951

Mubashir, M., Shao, L., & Seed, L. (2013). A survey on fall detection: Principles and approaches. *Neurocomputing*, *100*, 144–152. https://doi.org/10.1016/j.neucom.2011.09.037

Nguyen, T. H., Pham, T. P., Ngo, C. Q., & Nguyen, T. T. (2016). A SVM Algorithm for Investigation of Tri-Accelerometer Based Falling Data. *American Journal of Signal Processing*, *6*(2), 56–65. https://doi.org/10.5923/j.ajsp.20160602.03

NHS. (2021). *Falls*. https://www.nhs.uk/conditions/falls/

Ojetola, O., Gaura, E. I., & Brusey, J. (2011). Fall Detection with Wearable Sensors—Safe (Smart Fall Detection). *2011 Seventh International Conference on Intelligent Environments*, 318–321. https://doi.org/10.1109/IE.2011.38

Oliveira, C. P. D., Colombo, C. D. S., & Nunes, D. J. V. (2024). Machine Learning Applied To Fall Detection in the Elderly. *Proceedings of the 20th Brazilian Symposium on Information Systems*, 1–9. https://doi.org/10.1145/3658271.3658330

Pasquetti, P. (2014). Pathogenesis and treatment of falls in elderly. *Clinical Cases in Mineral and Bone Metabolism*. https://doi.org/10.11138/ccmbm/2014.11.3.222

Patel, J., & Goyal, R. (2007). Applications of Artificial Neural Networks in Medical Science. *Current Clinical Pharmacology*, *2*(3), 217–226. https://doi.org/10.2174/157488407781668811

Qi, P., Chiaro, D., & Piccialli, F. (2023). FL-FD: Federated learning-based fall detection with multimodal data fusion. *Information Fusion*, *99*, 101890. https://doi.org/10.1016/j.inffus.2023.101890

Rastogi, S., & Singh, J. (2021). A systematic review on machine learning for fall detection system. *Computational Intelligence*, *37*(2), 951–974. https://doi.org/10.1111/coin.12441

Saadeh, W., Butt, S. A., & Altaf, M. A. Bin. (2019). A Patient-Specific Single Sensor IoT-Based Wearable Fall Prediction and Detection System. *IEEE Transactions on Neural Systems and Rehabilitation Engineering : A Publication of the IEEE Engineering in Medicine and Biology Society*, *27*(5), 995–1003. https://doi.org/10.1109/TNSRE.2019.2911602

Saleh, M., Abbas, M., & Le Jeannès, R. B. (2021). FallAllD: An Open Dataset of Human Falls and Activities of Daily Living for Classical and Deep Learning Applications. *IEEE Sensors Journal*, *21*(2), 1849–1858. https://doi.org/10.1109/JSEN.2020.3018335

Seneviratne, S., Hu, Y., Nguyen, T., Lan, G., Khalifa, S., Thilakarathna, K., Hassan, M., & Seneviratne, A. (2017). A Survey of Wearable Devices and Challenges. *IEEE Communications Surveys & Tutorials*, *19*(4), 2573–2620. https://doi.org/10.1109/COMST.2017.2731979

Shaima R.M, E., Rudzidatul Akmam, D., & Nur Izdihar Muhd, A. (2023). *Fall Detection and Monitoring using Machine Learning: A Comparative Study*. *14*(2), 6.

Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2014). Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors*, *14*(6), 10146–10176. https://doi.org/10.3390/s140610146

Son, H., Lim, J. W., Park, S., Park, B., Han, J., Kim, H. B., Lee, M. C., Jang, K.-J., Kim, G., & Chung, J. H. (2022). A Machine Learning Approach for the Classification of Falls and Activities of Daily Living in Agricultural Workers. *IEEE Access*, *10*, 77418–

77431. https://doi.org/10.1109/ACCESS.2022.3190618

Stampfler, T., Elgendi, M., Fletcher, R. R., & Menon, C. (2022). Fall detection using accelerometer-based smartphones: Where do we go from here? *Frontiers in Public Health*, *10*, 996021. https://doi.org/10.3389/fpubh.2022.996021

Sundaram, B. M., Rajalakshmi, B., Mandal, R. K., Nair, S., & Choudhary, S. S. (2023). Fall Detection Among Elderly Using Deep Learning. *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 554–558. https://doi.org/10.1109/IITCEE57236.2023.10090887

Usmani, S., Saboor, A., Haris, M., Khan, M. A., & Park, H. (2021). Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors*, *21*(15), 5134. https://doi.org/10.3390/s21155134

Venkateswaran, R., Babu, B. N. V. M., Parameswari, M., VamsiLala, P. N. V., Ambigaipriya, S., & Keerthana, N. V. (2024). Fall Detection in Elderly People Using Compact Wearable Sensors with AI. *2024 13th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 369–372. https://doi.org/10.1109/SMART63812.2024.10882586

Walczak, S., & Cerpa, N. (2019). *Artificial Neural Networks In Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction*. 40–53.

Yang, X., Zhao, Y., Street, G. M., Huang, Y., Filip To, S. D., & Purswell, J. L. (2021). Classification of broiler behaviours using triaxial accelerometer and machine learning. *Animal*, *15*(7), 100269. https://doi.org/10.1016/j.animal.2021.100269

Yhdego, H., Paolini, C., & Audette, M. (2023). Toward Real-Time, Robust Wearable Sensor Fall Detection Using Deep Learning Methods: A Feasibility Study. *Applied Sciences*, *13*(8), 4988. https://doi.org/10.3390/app13084988

Yuan, B., Tang, Z., Zhang, P., & Lv, F. (2023). Thermal Calibration of Triaxial Accelerometer for Tilt Measurement. *Sensors*, *23*(4), 2105. https://doi.org/10.3390/s23042105

Zahedian-Nasab, N., Jaberi, A., Shirazi, F., & Kavousipor, S. (2021). Effect of virtual reality exercises on balance and fall in elderly people with fall risk: A randomized controlled trial. *BMC Geriatrics*, *21*(1), 509. https://doi.org/10.1186/s12877-021-02462-w

Zaid Salah, O., Selvaperumal, S. K., & Abdulla, R. (2022). Accelerometer-based elderly fall detection system using edge artificial intelligence architecture. *International Journal of Electrical and Computer Engineering (IJECE)*, *12*(4), 4430. https://doi.org/10.11591/ijece.v12i4.pp4430-4438

Zhai, X., Chu, X., Chai, C. S., Jong, M. S. Y., Istenic, A., Spector, M., Liu, J.-B., Yuan, J., & Li, Y. (2021). A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity*, *2021*(1), 8812542. https://doi.org/10.1155/2021/8812542