Empowering Elderly Safety: 1D-CNN and IoT-Enabled Fall Detection System

Rahul Modak, Koushik Majumder, Santanu Chatterjee, Rabindra Nath Shaw, and Ankush Ghosh

Abstract The integration of cutting-edge technology, including deep learning, smartphone capabilities, and wearable devices, has sparked a transformative revolution in fall detection systems, offering real-time monitoring and swift response in the event of a fall. This research study presents a fall detection system that harnesses advanced deep learning techniques, particularly 1D convolutional neural networks (CNNs), to achieve remarkable accuracy scores of 91% and 92%. Rigorously evaluated using the Sisfall and UMA Fall datasets, which consist of 9 and 25 features, respectively, obtained through meticulous hand engineering, this system demonstrates its efficacy in detecting falls. The potential of this advanced fall detection system lies in its ability to significantly enhance the safety and well-being of individuals by enabling timely assistance after a fall. By leveraging the power of artificial intelligence and state-of-the-art technology, the system promises to amplify the efficiency of fall detection in real-world scenarios, providing reassurance and peace of mind for both individuals and their caregivers. Particularly beneficial for vulnerable populations like the elderly, this technology holds the promise of mitigating the risk of severe injuries and fatalities resulting from falls. The study's findings underscore the substantial progress that can be achieved in fall detection by seamlessly integrating deep learning, smartphone technology, and wearable devices. This integration paves the way for a future where prompt assistance becomes standard practice, reducing the potential consequences of falls and ultimately improving the quality of life for those at risk. As this research sheds light on the immense benefits of advanced fall detection systems, it serves as a significant step forward in ensuring the safety and welfare of individuals, fostering a safer environment for everyone.

Keywords Deep learning (DL) · Artificial intelligence (AI) · One-dimensional convolutional neural network (1DCNN) · Internet of Things (IoT)

R. Modak · K. Majumder (✉) · S. Chatterjee
Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Kolkata, WB, India
e-mail: koushikzone@yahoo.com

R. N. Shaw · A. Ghosh
University Center for Research and Development (UCRD), Chandigarh University, Chandigarh, India

R. N. Shaw et al. (eds.), *Innovations in Electrical and Electronic Engineering*, Lecture Notes in Electrical Engineering 1115, https://doi.org/10.1007/978-981-99-8661-3_38

1 Introduction

In the ever-evolving realm of medicine, remarkable progress has been made since the turn of the millennium, leading to a notable increase in life expectancy by five years. This momentous advancement has triggered a significant demographic shift, with the elderly population now accounting for 8.5% of the global populace. Projections by the esteemed National Institutes of Health (NIH) indicate that by the year 2050, this proportion will escalate to a staggering 20%. Among the many concerns tied to the geriatric community, the prevalence of falls stands out as a substantial risk factor and the second leading cause of mortality. According to the esteemed World Health Organization (WHO), an estimated 37.3 million accidents occur annually, necessitating medical attention and leading to over 646,000 fatalities. Falls afflict around 30% of individuals aged 65 and above each year, with this percentage surging to 50% for those aged 80 and older. In response to this pressing issue, wearable healthcare applications have emerged as a promising solution, thanks to the advancements witnessed in hardware and operating systems. Particularly, automated fall detection systems (FDSs) have garnered significant interest in academic research circles due to their remarkable ability to identify and promptly report falls, consequently mitigating their impact and consequences on the elderly.

Market projections suggest that automated fall detection systems are poised to capture 60% of the fall detection systems market share between 2019 and 2020, showcasing a compound annual growth rate (CAGR) of approximately 4% from 2019 to 2029. Governments worldwide are investing in research pertaining to fall detection devices to address the substantial portion of healthcare expenditures allocated to fall-related injuries [1, 2]. In conclusion, the advancements in medicine have led to an extended life expectancy, resulting in a demographic shift with a growing elderly population. Falls, being a significant concern for the elderly, are responsible for a considerable number of accidents and fatalities each year. Wearable healthcare applications, particularly automated fall detection systems, have emerged as a promising solution to this issue. Market trends indicate a rising demand for such systems, and governments recognize the need for investing in fall detection device research to mitigate healthcare costs associated with fall-related injuries.

1.1 Fall Risk Factors

The act of falling occurs when an individual encounters difficulty in maintaining their balance and attempts to regain an upright position. While young people possess the physical strength to recover their balance, older individuals face greater challenges due to their weakened physical state. The causes of such collapses can be diverse, and the term "risk factors for falls" encompasses all possible circumstances that might contribute to a fall. In truth, the incidence of falls is the result of a complex interplay of multiple factors. Therefore, understanding the likely risk factors associated with falls among the elderly is crucial. A comprehensive understanding of these risk variables enables the development of more effective strategies to prevent falls. Various factors, including biological, behavioral, demographic, and environmental elements, may contribute to falls. Extensive research has identified a range of potential hazards, which are outlined in Fig. 1. Falling from beds ranks as the second most common cause of fall-related injuries, second only to physiological issues [3]. Risk factors related to behavior are intertwined with people's thoughts, emotions, and daily activities. These factors can be addressed through strategic interventions. For example, if a person experiences trips or falls due to excessive drug or alcohol misuse, their behavior patterns can be altered through appropriate treatment strategies. Environmental risk factors stem from the immediate surroundings of an individual. Cracked sidewalks, uneven surfaces, and inadequate lighting are prominent examples of environmental risk factors. Biological risk factors include an individual's age, gender, and overall physical health. Acute and chronic diseases, diabetes, cardiovascular problems, vision impairments, balance issues, and high or low blood pressure are among the biological risk factors. While age and gender are unalterable biological characteristics, illnesses can be prevented or managed through proper medical treatment, and both physical and mental well-being can be enhanced [2]. Falls are the result of a complex interplay of factors. Risk factors for falls include biological, behavioral, demographic, and environmental elements. Understanding these factors is crucial in developing effective strategies to prevent falls among the elderly. By addressing behavior-related risks through interventions and mitigating environmental hazards, the incidence of falls can be reduced. Additionally, managing and preventing diseases and promoting overall health play a vital role in minimizing the biological risk factors associated with falls.

1.2 Types of Falls Fall

Up until the 1990s, categorizing fall was a significant problem. The largest obstacle was a lack of agreement among researchers. The majority of the classification at that time was based on the causes of falls. Depending on the position preceding a fall, there were three (other categories of falls also shown in (Fig. 2)) main categories of falls:

a. **Fall from Bed**

- At the time of the fall, the person is lying in bed either sleeping or not.
- From bed height to floor height, the body height decreases. The body typically experiences what feels like a free fall motion at that time.
- The body is in a position on the floor that is close to the bed.
- The entire procedure occurs in a series of smaller activities over the course of 1–3 s.

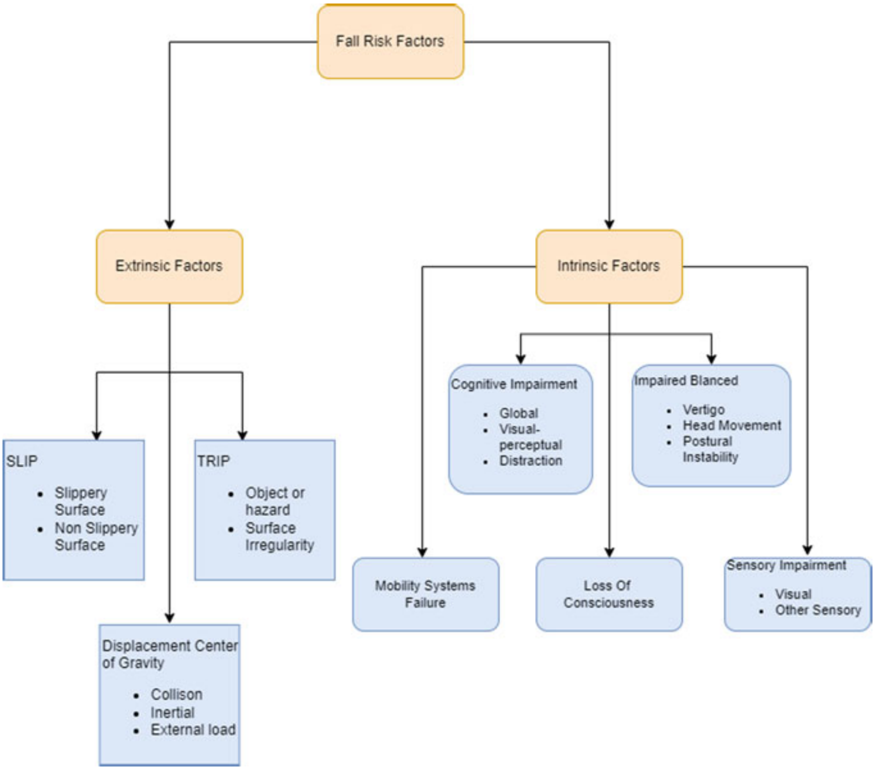


Fig. 1 Fall risk factors

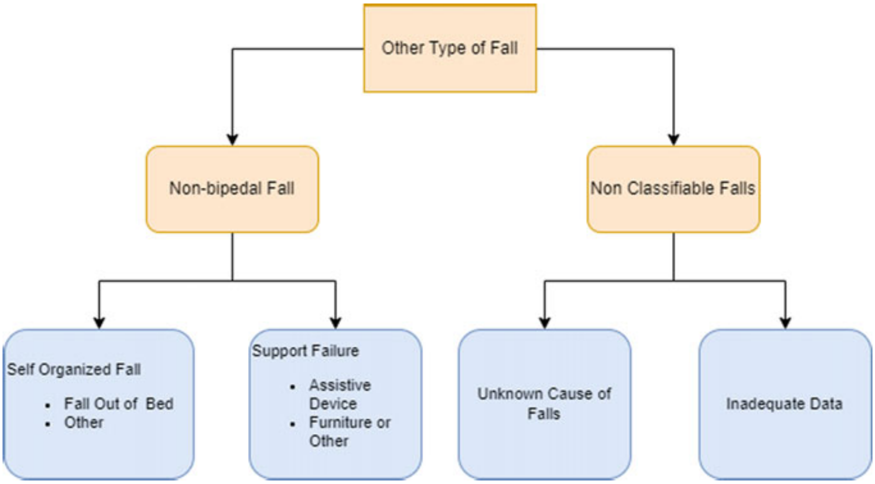


Fig. 2 Other categories of falls

b. **Fall from Sitting**

- At the start of the fall, the person is sitting on a chair or another piece of furniture approximately at the same height.
- The head descends in a free fall fashion until its height is reduced to the floor.
- The body is lying close to the chair in this position.
- The falling process is divided into 1–3 s sub-actions.

c. **Fall from Walking or Standing**

- When the fall begins, the person is either standing or walking.
- The head lowers itself to the floor while lying on it from a level that is equal to the person's height. It might move slightly while lying.
- Typically, the fall is unidirectional.

1.3 Fall Detection

Devices that are worn, devices that rely on cameras, and devices that measure the environment are the three primary categories that fall detection methods may be broken down into, devices that are worn, devices that rely on cameras, and devices that measure the environment which are the three primary categories that fall detection methods may be broken down into. Classification of fall detection is shown in Fig. 3. The persons who are at danger of falling are required to wear some kind of wearable gadget or apparel in the strategy that utilizes wearable technologies. The data that these sensors collect on the movement or posture of the body is then sent into a processing algorithm which determines whether or not a tumble has occurred. However, some users feel that wearable technology is excessively obtrusive and a hassle to carry about with them. They do not bother to continuously have a gadget on their person. In addition, there is an issue with the apparatus' placement. Some actions, such as dozing or moving, may displace the device from its original position, resulting in less precise results. Sleeping and moving around are examples of such activities. It would seem that the camera-based technique is successful in resolving some of these issues. The cameras are set up in strategic locations so that they may carry out unobtrusive, round-the-clock surveillance on the elderly. In contrast to sensors, cameras have the capability of evaluating and analyzing a wide variety of characteristics simultaneously. When camera prices were higher, originally fewer people wanted these kinds of devices because they were more costly. These gadgets also have the capability of saving the data they collect so that it may be analyzed and consulted at a later time. The strategy known as the ambiance device requires the installation of certain sensors in close proximity to the individuals being monitored, including on a wall, floor, or bed. These sensors are responsible for collecting data, which are then used as input by an algorithm in order to identify whether or not a fall has happened. As a direct result of this, the incidence is reported to the carers.

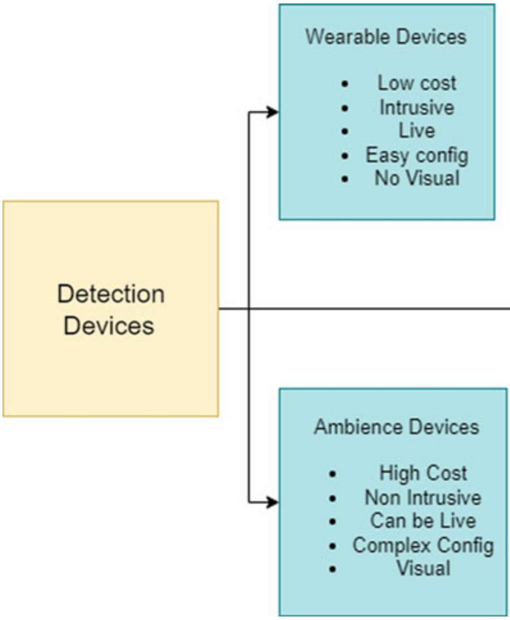


Fig. 3 Fall detection devices

Because the associated individual does not have to wear a sensor, they do not have any concerns about any form of oversight [2].

In fall detection and prediction systems, camera-based sensors are widely employed. Separate cameras are utilized in such systems to monitor the routing activities of each individual. Camera-based methods are costly and necessitate a massive quantity of data storage and processing. This method of operation is extremely complex and requires a more potent GPU and CPU. In addition to their advantages, camera-based systems have disadvantages such as privacy concerns and the incapacity to track beyond the camera's field of view [2]. Because low-cost physical sensors are becoming more readily available, there has been a recent explosion in interest in wearable sensor-based computing systems. Real-time monitoring can be obtained via the employment of wearable-based devices rather than environment-based monitoring equipment. As a result, collect data that belongs to the user. In these types of systems, the devices that are used are often microcontrollers that are outfitted with inertial measurement units. This helps to reduce the overall size of the device while also extending the battery's lifespan. Wearable technology often results in reduced overall economic expenses as compared to context-aware technologies [1]. In addition to accelerometers, gyroscopes, and force sensors, the components of wearable technologies also include gyroscopes. However, it is challenging for an individual to wear multiple devices. In contrast, smartphone-based systems are inexpensive and can be utilized outside of controlled environments as the user goes about his or her daily life. Moreover, smartphones incorporate sensors such as accelerometers, gyroscopes, and magnetometers. Thus, smartphones are frequently considered the most appropriate technology for applications in health care, security, athletics, fitness, gait analysis, and accident prediction [4]. Due to their proximity to the human body's center of gravity, the sternum and the waist have been demonstrated to be the optimal locations for a wearable accelerometer designed to detect falls accurately. Certain studies have revealed that carrying a smartphone in a pocket can hinder the effectiveness of detection systems, particularly when the device is allowed to move freely within the pocket, and the accelerometer fails to determine the user's movement accurately. Some suggested solutions propose optimal results when the smartphone is securely attached using an adjustable band around the chest, waist, or a similar fastening element. However, this rigid attachment compromises user comfort and limits their ability to access the smartphone's standard features [5]. Smart watches are wristwatches with a miniature display, integrated sensors, and Internet connectivity. Smartwatch manufacturers seek to develop a new form of wearable device capable of displaying brief communications such as SMS, RSS feeds, and Facebook notifications. Smartwatches enhance the system's ergonomics and (typically) the resolution and range of the integrated accelerometers in comparison to smartphones. In contrast, the wrist movement (where the chronometer is affixed) does not always indicate the stability of the body. Therefore, abrupt limb movements that are not inherently associated with falls can readily produce false positives. (i.e., activities that are incorrectly identified as falls) [5]. A series of ambiance device approaches are installed in the immediate proximity of the associated individuals in the ambience device method, including on a wall, floor, and bed. Using the information collected from these sensors, an algorithm determines whether a fall has occurred. The incident is consequently conveyed to the attendants. As the individual is not required to wear a sensor, he or she is unconcerned about any type of surveillance [2].

1.4 Fall Prevention

The prevention of falls is an essential aspect of providing for senior individuals, despite the impossibility of ensuring their complete prevention. There are, however, there are measures that can be taken to reduce the danger of accidents and guarantee the safety of the targeted population. This can be accomplished by routinely assessing and continuously monitoring recognized fall risk factors [6]. If these parameters' values lie within an acceptable range, it can be presumed that the individuals are in a relatively secure zone. The following exercises and practices can help prevent falls:

1. Observe their mobility pay close attention to whether they have trouble rising from a chair or walking unassisted. If they appear unsteady or cling to walls or objects frequently for balance, this may indicate an increased risk of collapsing. Encourage the use of canes and walkers, if necessary.

2. Certain medications can cause vertigo, lethargy, and other adverse effects that increase the risk of falling. Discuss their medications with their healthcare provider to ensure that the prescribed medications are suitable and do not pose a fall risk. Any concerns regarding adverse effects should be addressed immediately.
3. Consider their general health condition, including any chronic maladies, balance issues, or sensory impairments, when assessing their overall health. Regular medical examinations and communication with healthcare providers can assist in identifying and treating health conditions that may contribute to falls.
4. Examinations of the eyes and eyewear on a regular basis: Vision problems can substantially increase the risk of falling. Encourage regular eye exams to detect any changes in vision, and ensure that they have the necessary eyewear (glasses or contact lenses). Vision correction can enhance spatial awareness and reduce the likelihood of stumbling or misjudging distances.
5. Create a safe living atmosphere: Remove potential hazards from their living environment. Remove debris, secure any loose rugs or carpets, and clear all pathways. To provide additional support, install handrails along staircases and in restrooms. Ensure that there is adequate illumination in all areas, particularly at night.
6. Encourage regular physical activity, regular exercise can improve strength, balance, and flexibility, all of which are crucial for preventing falls. Encourage them to participate in senior-specific activities such as walking, tai chi, and chair exercises. Before beginning any exercise program, it is essential to consult a healthcare professional to ensure that it is appropriate for the individual's abilities and medical condition.
7. Encourage a healthful lifestyle in order to preserve an individual's overall health. This includes a healthy diet, sufficient hydration, and adequate rest. A nutrient-dense diet can support bone health and muscle stamina. Staying hydrated helps maintain proper physiological functions, and adequate rest ensures that the individual is vigilant and less likely to be involved in an accident.

These measures can substantially reduce the risk of falls, but they cannot eliminate the possibility entirely. A supportive environment, regular monitoring, and ongoing communication with healthcare professionals are essential for promoting the welfare and well-being of senior citizens.

2 Background

Falls are a common occurrence among people of all ages, but they are particularly prevalent among the elderly due to the gradual decline in their physical abilities. Falls can result in severe injuries such as fractures, concussions, and even fatalities. In recent years, significant research has focused on developing automated methods for detecting and analyzing falls. The use of advanced machine learning techniques for fall detection is gaining importance, as these techniques enable systems to learn from data collected through various sensors that capture information related to different aspects of falls. By applying machine learning algorithms to this data, the system can classify and identify fall events based on specific criteria.

There are several machine learning and deep learning algorithms widely used for fall prevention and detection, including SVM, Artificial Neural Networks (ANNs), RF, KNN, NB, CNN, RNN, etc. These algorithms have shown promise in accurately and efficiently detecting falls. Researchers aim to leverage the capabilities of these advanced algorithms to develop highly accurate and efficient fall detection systems that can help prevent falls in a timely manner and minimize their negative consequences [1, 2, 8].

3 Literature Review

Important in healthcare and senior care, fall detection seeks to automatically detect accidents and alert caregivers or emergency services. By analyzing sensor data from peripheral devices or cameras, traditional ML algorithms can be used for detection of fall. A smartphone and smartwatch-based fall detection system is utilizing smartwatch and smartphones accelerometers, gyroscopes, and magnetometers. The majority of smartphones contain a GPS module that can be added to other portable devices, like smart watches and bands. It is possible to connect a smart band, watch, or other portable device lacking a GPS module to a smartphone's GPS module. In this instance, the mobile phone functions as both a monitoring device and an Internet gateway that can transmit real-time location information [9]. A variety of sensors, such as those on the wall, the floor, the bed, are installed throughout a person's residence in order to track their movements while using an ambience device (Fig. 4). These sensors capture data, which are then analyzed by an algorithm to ascertain whether a fall occurred. If these sensors detect an accident, the monitoring service will notify the caretaker [8, 10]. Researchers discovered that smartphones and ambient devices can cooperate reasonably well, so they devised a novel method for fall detection using smartphones as the master monitoring device and ambient device sensors as subordinate sensors [7].

3.1 Traditional Machine Learning Approach for Fall Detection

Traditional machine learning methods have gained significant popularity in the domain of fall detection due to their ability to analyze intricate data patterns and make well-informed decisions based on historical data examples. In contrast to deep learning models, which often demand large volumes of data and computational

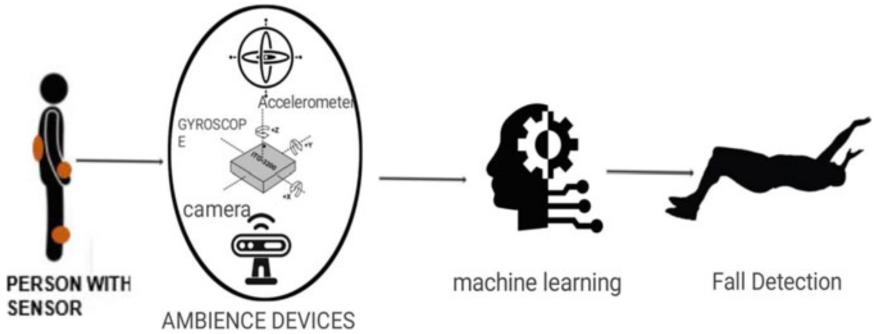


Fig. 4 Fall detection system

resources, traditional machine learning techniques can prove to be more practical and efficient, particularly in certain fall detection scenarios. The crux of the traditional machine learning approach lies in the model training phase. A variety of classifiers, including SVM, random forests, KNN, decision trees, etc. are employed. During the training process, the classifiers are fed with a labeled dataset, where each data sample is designated as either a fall or a non-fall instance. The model endeavors to discern meaningful patterns and associations between the extracted features and their corresponding class labels. By utilizing historical data and extracting informative features, these traditional machine learning models can effectively detect falls and differentiate them from regular movements. This capability plays a pivotal role in enhancing the safety and well-being of individuals, especially the elderly or those at risk of falling. Prompt detection of falls can lead to rapid responses, such as alerting caregivers, medical professionals, or emergency services, potentially minimizing the severity of injuries and improving the overall care for vulnerable populations.

In the realm of fall detection methods, several researchers have proposed innovative approaches. Ramachandran et al. [7] introduced a method that considers both sensor measurements and the individual's biological profile. They used the UMA_ADL_FALL_Dataset and employed Ordinal Logistic Regression, with KNN yielding the highest accuracy of 84.1%. Hussain et al. [8], on the other hand, utilized the Sisfall dataset and incorporated a low-pass IIR Butterworth filter and six extracted features. Remarkably, their algorithm achieved an impressive accuracy of 99.98% with SVM outperforming other methods. Toda and Shinomiya [10] took a unique approach using passive RFID (Fig. 5) sensors attached to footwear, applying the random forest algorithm. Their method achieved high accuracy with F-measure scores of 98% for person-dependent cases and 94% for person-independent cases. In a different study, Vallabh et al. [11] investigated fall detection using the "MobiFall" dataset, focusing on distinguishing between Activities of Daily Living (ADL) and fall activities. They employed various classification techniques, with KNN performing the best and achieving an accuracy of 87.5%. Chelli and Pätzold [12] evaluated KNN and ANN in identifying human activities, including falls. Both algorithms achieved high accuracies, with KNN at 81.2% and ANN at 87.8%. The researchers further improved their results by extracting additional features from acceleration and angular velocity data. In another perspective, Miawarni et al. [13] utilized SVM and deep learning techniques on the eHomeSeniors dataset, which included thermal sensors, reaching an accuracy of 84.62% by adjusting the gamma value without normalization or standardization. Conversely, Rashid et al. [14] simulated the Sisfall dataset and tested various algorithms, such as DT, NB, SVM, KNN, and Ensemble Classifiers. Fine KNN has emerged as the top-performing algorithm, achieving accuracies of 83.76% and 84.64% in different experiments. The comparative analysis highlights that SVM, KNN, and ANN are commonly used and achieved high accuracies in fall detection tasks. Each method possesses its unique strengths and limitations, and the choice of the most suitable approach depends on factors such as dataset characteristics, computational efficiency, and specific application requirements. Overall, the advancements in fall detection research showcase diverse algorithms and techniques, ranging from logistic regression to random forest and deep learning, that contribute to improving the accuracy and reliability of fall detection systems.

Using sensor data to identify fall-related patterns and characteristics, traditional machine learning algorithms can detect falls effectively. For fall detection, decision trees and Naive Bayes are two additional machine learning algorithms that may require manual feature engineering. Both can be trained to detect falls by analyzing the features that are indicative of falls using sensor data. As with SVMs and random forests, however, traditional ML methods are used for fall detection, but they have limitations when compared to deep learning methods [3, 4, 10–12].

For fall detection using smartphones, peripheral devices, and ambient devices, deep learning models offer several advantages over conventional machine learning. These benefits include accurate detection, resistance to environmental changes,

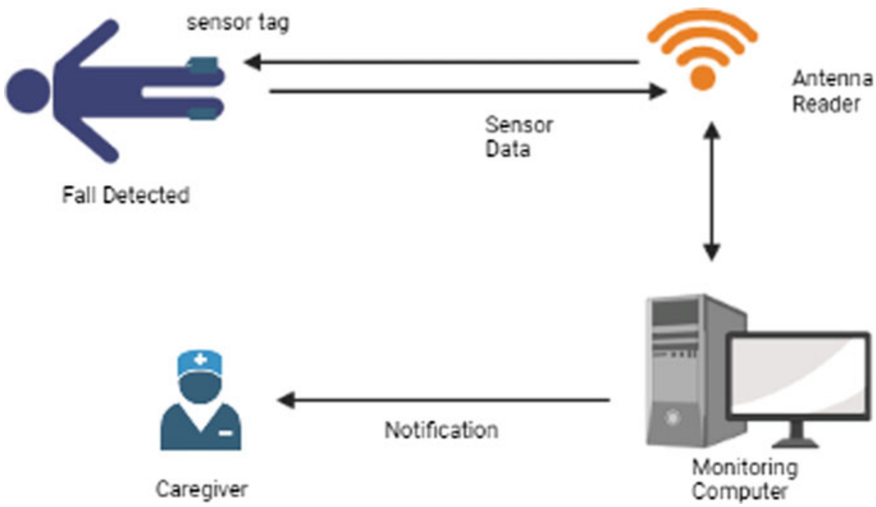


Fig. 5 Alert-based fall detection system

feature extraction from raw data, real-time processing, scalability, transfer learning, individual user adaptability, non-intrusive monitoring, context-aware detection, continuous monitoring, integration with emergency services, and evolving models. These developments contribute to the creation of more dependable and effective fall detection systems, thereby enhancing the safety and well-being of individuals, especially the elderly and vulnerable populations. The advantages are describing below.

- Accurate Detection:

  Models employing deep learning can detect accidents with a high degree of precision. Traditional machine learning models require a substantial quantity of labeled data, and the model's accuracy is highly dependent on the labeling quality. Deep learning, on the other hand, models can learn from unprocessed data and, with the assistance of complex neural networks, can recognize patterns and make accurate predictions.
- Robustness:

  Traditional machine learning models are more susceptible to environmental changes than deep learning models. Traditional machine learning models require consistent data with regard to quality, format, and sampling rate. However, deep learning models can adapt to changes in the environment and perform well despite chaotic or insufficient data.
- Feature Extraction:

  The ability of DL models to extract features from unprocessed data eliminates the need for domain-specific knowledge and feature engineering. Traditional machine learning models, in contrast, require time-intensive and domain-specific feature engineering.
- Real-Time Processing:

  Real-time data processing by deep learning models is essential for fall detection. Traditional machine learning models may require bulk processing, which may introduce latency into the system and pose a problem for applications requiring real-time processing.
- Scalability:

  Deep learning models are highly scalable and able to manage massive data volumes. Traditional machine learning models may struggle to scale as the model's complexity and data volume increase.
- Individual User Adaptability:

  Deep learning models can adapt to the behavior and movement patterns of individual users. By perpetually learning from data collected from a particular user, the model can customize fall detection based on the user's unique characteristics and behaviors. This adaptability increases the accuracy of fall detection systems and decreases false alarms, making them more trustworthy for individual users.

- Monitoring Without Invasion:

  Using smartphones, wearable devices, and ubiquitous devices for fall detection provides nonintrusive monitoring, enabling individuals to maintain their privacy and independence. These devices can be incorporated into users' daily activities without causing discomfort or inconvenience. By analyzing sensor data from these devices, deep learning models enable unobtrusive fall detection without requiring individuals to wear or carry additional specialized equipment.
- Aware of Context Detection:

  Along with movement patterns, deep learning models can capture contextual information to improve the accuracy of fall detection. By analyzing additional contextual data such as time of day, location, and environmental conditions, deep learning models can differentiate between normal activities and falls more effectively. This context-aware approach reduces false positives and improves fall detection system reliability.
- Multimodal Data Fusion:

  Deep learning models can effectively combine data from multiple sensors to enhance the effectiveness of fall detection. Smartphones, wearable devices, and ambient devices frequently contain GPS, accelerometers, gyroscopes, barometers, and other sensors. Models employing deep learning can incorporate data from these various sensors, thereby obtaining a more complete picture of users' movements and activities. By integrating data from multiple modalities, the models can distinguish between normal activities and accidents more effectively.
- Continuous Observation:

  Models based on deep learning enable continuous monitoring of individuals, providing fall detection capabilities around the clock. Smartphones, wearable devices, and ubiquitous devices can collect data throughout the day, allowing users' activities to be monitored in real time. This continuous stream of data can be processed by deep learning models, allowing falls to be detected promptly and appropriate actions to be taken.
- Compatibility with Emergency Services

  Fall detection systems based on deep learning can integrate seamlessly with emergency services and caregiver notifications. When a fall is detected, the system can autonomously send alerts to designated caregivers or emergency services, ensuring that the individual in need receives immediate assistance. This integration expedites response times and improves the safety and well-being of all users.
- Evolving Models:

  As more data becomes available, deep learning models can evolve and develop continuously. By retraining the model with new labeled data, fall detection accuracy can be improved. This adaptability enables fall detection systems to remain current and enhance their performance by learning from new examples.

- Computing Capabilities at the Edge:

  The optimization of deep learning models for edge computing enables fall detection to be performed directly on smartphones, wearable devices, and ambient devices. Computing at the network's edge reduces the need for cloud-based processing, which can enhance response times and privacy. By executing deep learning models locally on the devices, fall detection can be conducted in real-time without the need for a constant Internet connection. This capability is especially advantageous in instances where network connectivity is limited or unreliable.

3.2 Deep Learning Approach for Fall Detection

Fall detection is a critical area of research aimed at ensuring the safety and well-being of vulnerable populations, particularly the elderly. Deep learning (DL) techniques have gained significant attention in recent years due to their ability to automatically learn intricate patterns and representations from raw data, often outperforming traditional machine learning approaches in various domains. In the context of fall detection, DL methods offer promising avenues for more accurate and robust detection systems.

In the domain of fall detection and activity recognition, numerous studies have explored the effectiveness of various machine learning techniques and deep learning methods. Syed et al. [15] introduced an innovative system that combines fall detection with the recognition of daily activities using data from the IMU accelerometer and gyroscope. Their CNN achieved an unweighted average recall rate of 88%, demonstrating its superior performance compared to other methods. In a separate study, Luna-Perejon et al. [16] investigated the use of Gated Recurrent Neural Networks (RNNs) based on LSTM and GRU for real-time fall detection using wearable devices with accelerometers. The selected architecture achieved impressive F1-scores of above 0.98 for falls and 0.85 for background activities, showcasing the effectiveness of RNN-based models. Likewise et al. [17] examined three datasets containing falls and activities of daily living. They applied Singular Value Decomposition (SVD) and 1D convolutional neural networks (CNNs) for feature extraction and recognition. The combination of dimension reduction features like SMV + SVD improved the accuracy to 75.65%, demonstrating the effectiveness of the proposed approach. Moreover, Garg, Sankalp, Bijaya Ketan Panigrahi, and Deepak Joshi [18] proposed a Deep Neural Network (DNN) for fall detection, showcasing its robustness to noise and achieving high accuracy, sensitivity, specificity, precision, and F-Score. The DNN performed well even in noisy environments, making it a valuable tool for real-time fall detection applications. Additionally, Kumar, H. Senthil, et al. [19] presented a comprehensive fall detection and activity identification system that utilized a CNN for feature extraction and XGB for categorization. The gradient-boosted CNN achieved an unweighted average recall of 89%, surpassing previous approaches. Overall, these studies demonstrate the effectiveness of deep learning methods, such as CNNs and RNNs, in fall detection and activity recognition tasks. The combination of deep learning models with other techniques, like XGB, enhances the accuracy and robustness of the systems. The proposed methods offer promising results for real-world fall detection applications, holding potential benefits in healthcare and elderly care settings. However, the choice of the most suitable method should consider factors such as dataset characteristics, computational resources, and specific application requirements. Wisesa, I. Wayan Wiprayoga, and Genggam Mahardika [20] utilized RNNs to analyze sensor data for fall detection and activity recognition. They used the UMA FALL ADL dataset, employing a one-layer LSTM architecture with 100 hidden neurons. The best performance was achieved using X-axis accelerometer data, with good overall classification on falls. Combining all sensor data yielded lower performance.

3.3 Observation and Findings

- Camera-based methods are expensive and require a powerful GPU and CPU, which makes them difficult to use and necessitates storing and processing an enormous quantity of data.
- The disadvantages of camera-based systems include privacy concerns and the incapacity to observe beyond the camera's field of view.
- Smartphones are not compatible with wearable fall detection devices. A fall detection system must measure four to six g (one g $= 9.8$ m/s^2), but smartphone accelerometer sensors may measure up to 2 g. Software adjustment can modify that.
- Using smartphone sensors like the accelerometer and gyroscope depletes the battery, which is a disadvantage for mobile devices. Optimization of software can extend the battery life of mobile devices.
- It may be difficult for medical professionals to comprehend technical terms such as energy consumption, battery backup, response time, and sensor installation.
- The use of wearable and ambient devices can provide users with greater privacy than camera-based fall detection systems, which pose significant privacy risks.
- In addition to detecting falls, wearable devices can monitor pulse rate, blood pressure, and sleep patterns.
- According to this study, KNN and SVM have the highest accuracy for mobile-based approaches, while CNN and RNN have the highest accuracy for ambience-based approaches.
- Deep learning models offer superior performance, reduced need for feature engineering, increased scalability and adaptability.
- As a solution, a hybrid approach combining smartphones and ambient devices with a model of deep learning is employed. A hybrid approach that incorporates inexpensive wearable and ambient devices can assist in problem resolution.
- In addition to detecting injuries, wearable devices can provide alerts and notifications for medication reminders, appointment reminders, and other vital information to elderly.

- The Sisfall dataset and the UMA Fall dataset are widely utilized in the field of fall detection research and are regarded as significant assets for the development and evaluation of fall detection algorithms and systems.

4 Proposed Model

One-dimensional CNN networks have emerged as prominent deep learning models in fall detection systems. They are utilized to extract meaningful features from input signals, which are then employed for classification. In the context of fall detection, 1D CNNs can analyze sensor data from peripheral devices or cameras to identify patterns related to falls. They are particularly effective at detecting temporal patterns in sequential data, which is often the case in fall detection applications [4, 15, 19].

IoT and cloud technologies have become integral components of fall detection system development and implementation. By placing Internet of Things devices, such as sensors, on the body or in the environment, falls or changes in motion indicative of falls can be detected. The data collected by these sensors can then be transmitted to the cloud for processing and analysis using machine (Fig. 6) learning algorithms like 1D CNN. Leveraging cloud technology enables remote monitoring and real-time alerts in the event of a fall. Caregivers or medical professionals can receive alerts on their mobile devices or computers and respond promptly to provide assistance. The cloud also facilitates the storage and analysis of large volumes of data, which can be utilized to improve the accuracy and effectiveness of fall detection systems over time. Moreover, the integration of IoT and cloud technologies enables the development of more sophisticated fall detection systems with additional capabilities such as predictive analytics and personalized feedback.

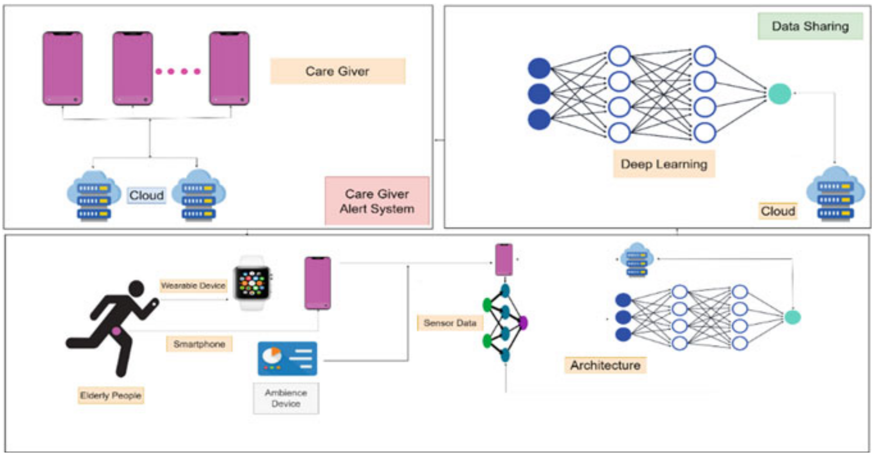


Fig. 6 Proposed model

To address fall detection among the elderly, we are currently designing a revolutionary system that combines smartphone and ambient device technology. Our proposed system utilizes deep learning to create a highly accurate fall detection model capable of distinguishing between falls and non-falls. This model forms the basis of an IoT-based alert system that incorporates both a smartphone and an ambient device, enabling the detection of falls both indoors and outdoors. If a fall occurs indoors, the model sends an alert to a family member inside the house, whereas it notifies a nearby caretaker about the user's location in the case of an outdoor fall. For falls occurring outside, the system automatically alerts a nearby caregiver. Wearable devices such as smartwatches and smart bands are connected (Fig. 7) to the system via Bluetooth and WiFi. However, even in instances where a person is not wearing any wearable devices while at home or does not own a smartphone, the system can still detect falls using ambient sensors. When a person falls outside, the system utilizes their smartphone and peripheral devices to detect the fall. The system remains connected to a cloud server, allowing the alert system to reach all nearby caregivers within the same network. Furthermore, the system prioritizes the fatality rate and issues alerts accordingly.

Our fall detection system represents an innovative solution aimed at improving the quality of life for the elderly. By harnessing advanced technology, we can detect falls with greater accuracy, ensuring prompt medical attention and potentially saving lives. The deep learning-based model can distinguish between falls and other movements, providing precise alerts only when necessary. The IoT-based alert system is a crucial feature that ensures that caregivers are promptly notified, irrespective of whether the user is indoors or outdoors. This feature is particularly vital in emergency situations

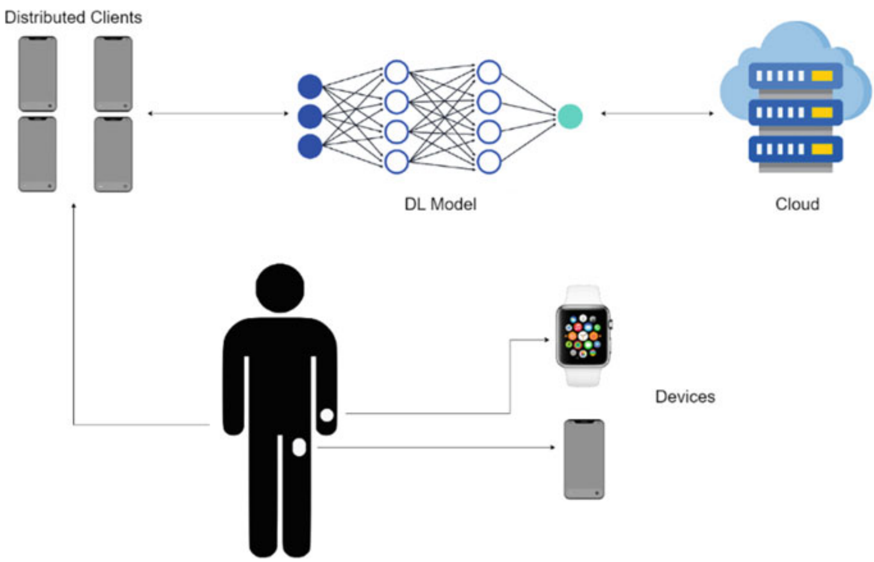


Fig. 7 Proposed model scheme

where every second counts. Additionally, the system's ability to prioritize high-risk falls ensures immediate attention from caregivers. The flexibility of our system is also noteworthy, as it can detect falls even without wearable devices or smartphones. This capability is especially valuable for individuals who may forget to wear their devices or do not own a smartphone. Through our groundbreaking technology, we believe that our fall detection system has the potential to revolutionize the elderly care industry. Accurate and timely fall detection can significantly enhance the quality of life for the elderly and their caregivers [3, 4, 9, 19].

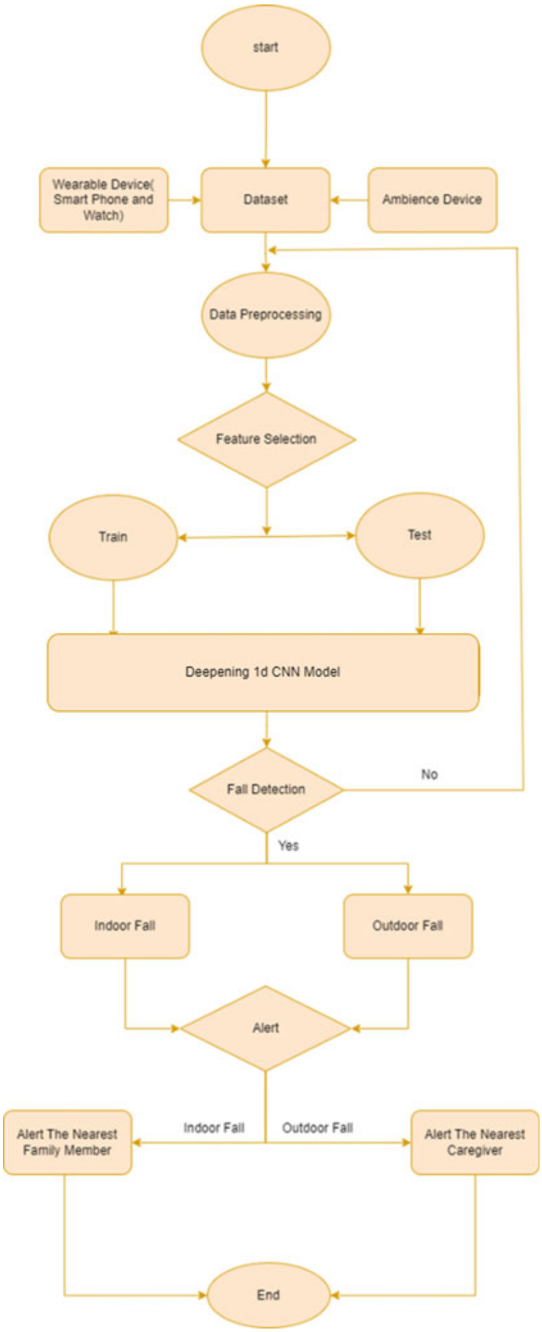
4.1 Proposed Model Architecture

The proposed model architecture starts by collecting data from two sources: a wearable device and an ambient device. The wearable device records data from sensors like accelerometers and gyroscopes, while the ambient device captures audio or video recordings. These two sets of data are combined to create a comprehensive dataset. The data then goes through preprocessing and feature extraction steps. This involves cleaning the data to remove any noise or outliers and performing sensor fusion to integrate information from the different sensors (Fig. 8). Relevant features, such as statistical measures or frequency-domain features, are extracted from the preprocessed data. After preprocessing, the dataset is divided into two groups: the training data and the test data. The training data is used to train two deep learning models: a 1D CNN. The 1D CNN model learns spatial patterns from the data using multiple convolutional layers and pooling layers for down-sampling. The output of the CNN is then flattened and connected to fully connected layers for classification. This allows the model to learn temporal dependencies in the data. Finally, the output from the fully connected layers is used for classification tasks [4, 15, 18, 19].

Once the models are trained, the preprocessed data is inputted into both models to detect fall events. The models produce probabilities indicating the likelihood of a fall event occurring. These probabilities are compared against a predetermined threshold to determine whether a fall has happened or not. In the case of an indoor fall, the alert system is triggered to notify the nearest family member. For outdoor falls, the system alerts the nearest caregiver.

To facilitate the alert system, the models are integrated into a cloud system. This cloud system enables real-time processing and analysis of the data, ensuring prompt detection of fall events. Once a fall is detected, the cloud system sends notifications to the designated recipients, such as the nearest family member or caregiver. These notifications can be delivered through various means, such as mobile applications, email, or SMS (Fig. 9). The integration with the cloud system allows for scalability, remote access, and efficient management of the alert system. The process loops back to the data processing step after triggering the alert system, allowing continuous monitoring and analysis. The process continues until no fall events are detected [3, 4, 18, 19].

Fig. 8 Proposed model architecture



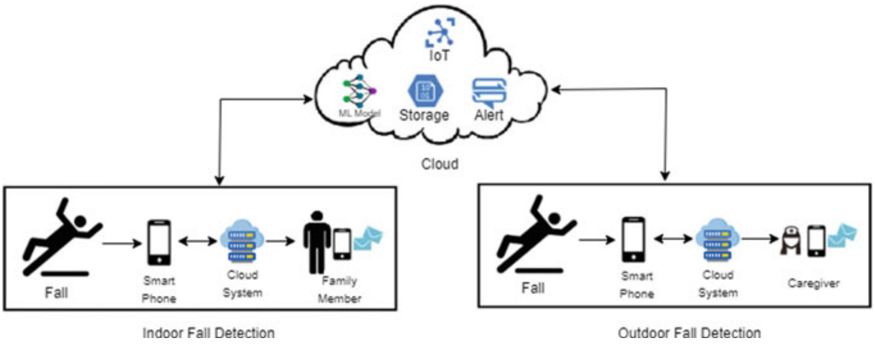


Fig. 9 Alert system scheme

4.2 Methodology

Data Collection

In this study, two datasets were used: the Sisfall dataset [21] and the UMA Fall dataset [22]. The Sisfall dataset was collected with the participation of 38 volunteers, who were divided into two categories: elderly and young adults. The geriatric group consisted of 15 participants (8 males and 7 females), while the young adults group consisted of 23 participants (11 males and 12 females). All participants were retirees from the Universidad de Antioquia and parents of active employees. It is important to note that all participants were in good health, without any gait problems (Fig. 10). The young adults performed activities of daily living (ADLs) and simulated falls, while the elderly individuals were advised not to perform falls and certain activities due to physical limitations or medical advice. Notably, a 60-year-old Judo expert, who was one of the participants, simulated both accidents and ADLs. Prior to their involvement in the study, all participants provided informed consent. The study protocol was approved by the Bio-Ethics Committee of the Medicine Faculty at the Universidad de Antioquia UDEA (Medellin, Colombia) in accordance with the principles outlined in the Declaration of Helsinki. The dataset was collected using a custom-built embedded device that included a Kinets MKL25Z128VLK4 microcontroller (NPX, Austin, Texas, USA), an Analog Devices ADXL345 accelerometer (16 g, 13 bits ADC), a Freescale MMA8451Q accelerometer (8 g, 14 bits ADC), and an ITG3200 gyro. The device was attached to the participants' waists, allowing accurate differentiation between activities using a single accelerometer system. For this study, only the acceleration data captured by the ADXL345 sensor was utilized due to its energy efficiency and wider range. However, the data collected by the second accelerometer and the gyroscope is also available for future research purposes. The sensor orientation was established with the positive z-axis facing forward, the positive y-axis aligned with gravity, and the positive x-axis positioned on the participant's right side. All experiments were conducted with a sampling frequency of 200 Hz from the beginning of the recording.

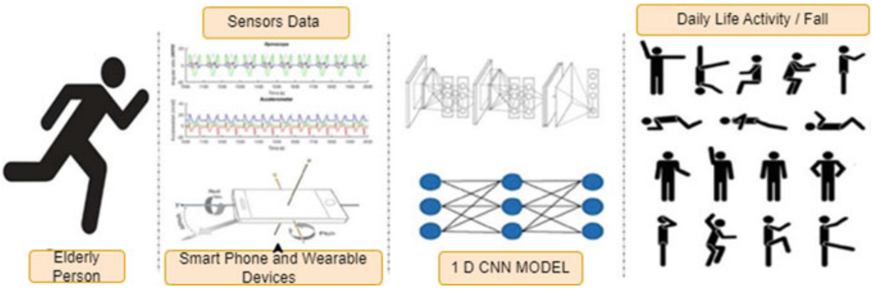


Fig. 10 Data collection technique

The UMA Fall dataset [22] was created by Santoyo-Ramón, José Antonio, Eduardo Casilari, and José Manuel Cano-García. The main objective of this dataset was to track the movements of participants during falls. In the initial experimental setup, 17 participants were equipped with smartphones connected wirelessly to four sensing nodes, or "motes," which were placed on their chest, waist, wrist, and ankle. Texas Instruments CC2650 SimpleLink™ Bluetooth low energy/multi-standard Sensor Tag modules were used as the sensing nodes. Each Sensor Tag module comprised an ARM CC2650 microcontroller, MEMS sensors, and an InveSense MPU-9250 Inertial Measurement Unit (IMU) with triaxial sensors for accelerometer, gyroscope, and magnetometer readings. The Sensor Tags were powered by a CR2032-type battery, allowing for wireless communication and full mobility. These sensing motes used a 2.4 GHz wireless MCU with ultra-low power consumption, supporting communication via BLE, 6LowPAN, or ZigBee. In the experimental setup, a smartphone has served as the central device of a Bluetooth Low Energy (BLE) piconet, acting as the master, while the four Sensor Tags has functioned as slaves. The smartphone received packets containing readings from the Sensor Tags. To assess fall detection algorithms, the researchers compared their performance using various sampling frequencies ranging from 5 to 256 Hz. To avoid Bluetooth network saturation, the Sensor Tags were set to transmit data at 20 Hz. The firmware of the Sensor Tags was modified to transmit the readings from the three IMU triaxial sensors via BLE at a rate of 50 ms. Furthermore, a smartphone, equipped with its own IMU, acted as a fifth sensor and was consistently placed in the subject's trouser pocket to capture thigh movement. The smartphone measurements were recorded at a sampling frequency of 200 Hz. This comprehensive dataset provides valuable information for evaluating fall detection algorithms and understanding human movements during falls in real-world scenarios.

After the original signal has undergone preprocessing, the next stage is featuring extraction for classification purposes. Typically, two types of feature extraction methods are used [8]; one employs nine features (Figs. 11 and 12) comprised data from all sensors, and the other employs 25 features (Figs. 13 and 14). These extracted

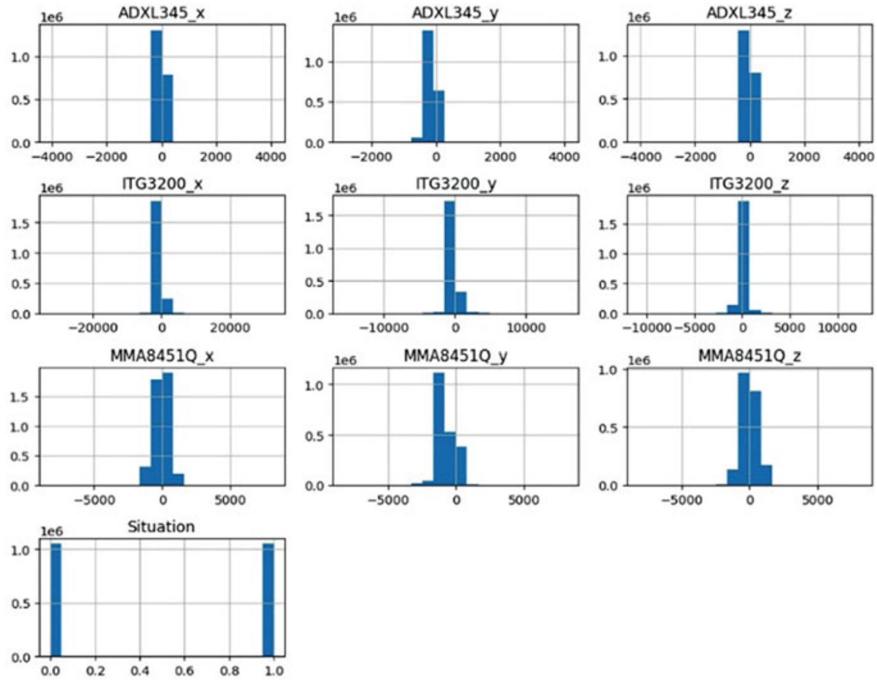


Fig. 11 Sisfall dataset with nine features

features include the signal's maximum amplitude, minimum amplitude, mean amplitude, variance, kurtosis, skewness, angular velocity, acceleration. These characteristics provide valuable information that can be used to distinguish and classify distinct patterns or signal characteristics. By taking into account these distinct characteristics, machine learning algorithms can effectively analyze and classify signal data for subsequent analysis or decision-making processes.

For an accelerometer signal: Let us assume that the accelerometer signal is denoted by $a(i)$, where i ranges from 1 to N (total number of samples).

- Maximum amplitude: Max_Acceleration = max($a(i)$).
  - This formula calculates the maximum value of the acceleration signal. It finds the highest recorded acceleration value in the signal.
- Minimum amplitude: Min_Acceleration = min($a(i)$).
  - This formula calculates the minimum value of the acceleration signal. It finds the lowest recorded acceleration value in the signal.
- Mean amplitude: Mean_Acceleration = $(1/N)$ * sum($a(i)$).
  - This formula calculates the mean (average) value of the acceleration signal. It sums up all the acceleration values in the signal and divides the sum by the total number of samples.

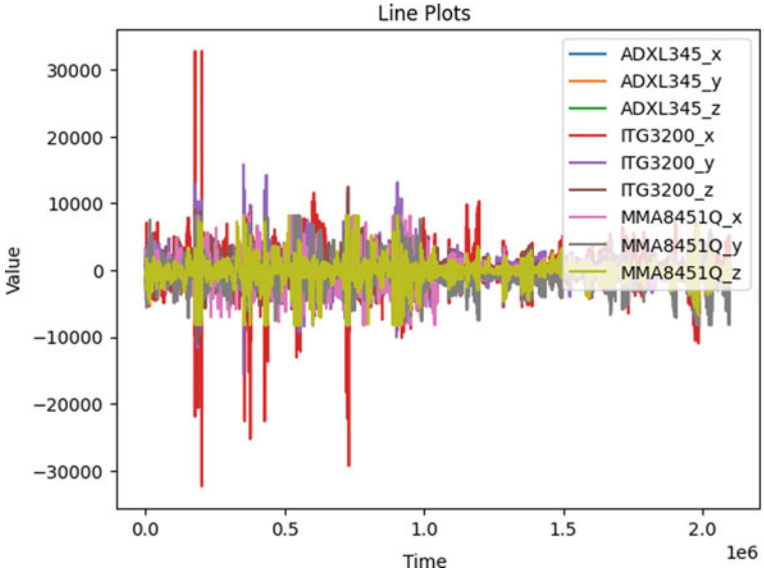


Fig. 12 Sisfall dataset with nine features (Line Plots)

- Variance: Variance_Acceleration = (1/N) * sum((a(i) - Mean_Acceleration) ^2).
  - This formula calculates the variance of the acceleration signal. It measures the spread or dispersion of the acceleration values around the mean. It sums up the squared differences between each acceleration value and the mean and then divides that sum by the total number of samples.
- Kurtosis: Kurtosis_Acceleration = (1/N) * sum(((a(i) Mean_Acceleration)/sqrt (Variance_Acceleration)) ^4).
  - This formula calculates the kurtosis of the acceleration signal. Kurtosis is a measure of the "tailedness" or the presence of outliers in the distribution of the signal. It normalizes the fourth moment of the acceleration signal by dividing it by the variance.
- Skewness: Skewness_Acceleration = (1/N) * sum(((a(i) - Mean_Acceleration)/ sqrt (Variance_Acceleration)) ^3).
  - This formula calculates the skewness of the acceleration signal. Skewness measures the asymmetry of the signal's distribution. It normalizes the third moment of the acceleration signal by dividing it by the variance.

For a gyroscope signal: Let us assume that the gyroscope signal is denoted by $g(i)$, where i ranges from 1 to N (total number of samples).

- Maximum amplitude: Max_AngularVelocity = max($g(i)$).
- Minimum amplitude: Min_AngularVelocity = min($g(i)$).

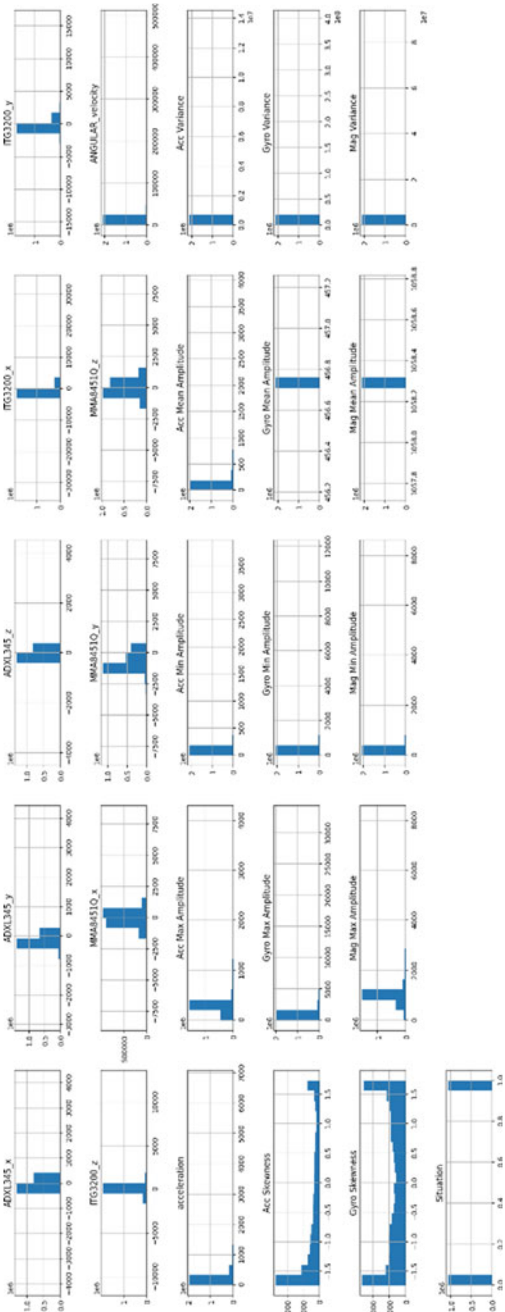


Fig. 13 Sisfall dataset with 25 features

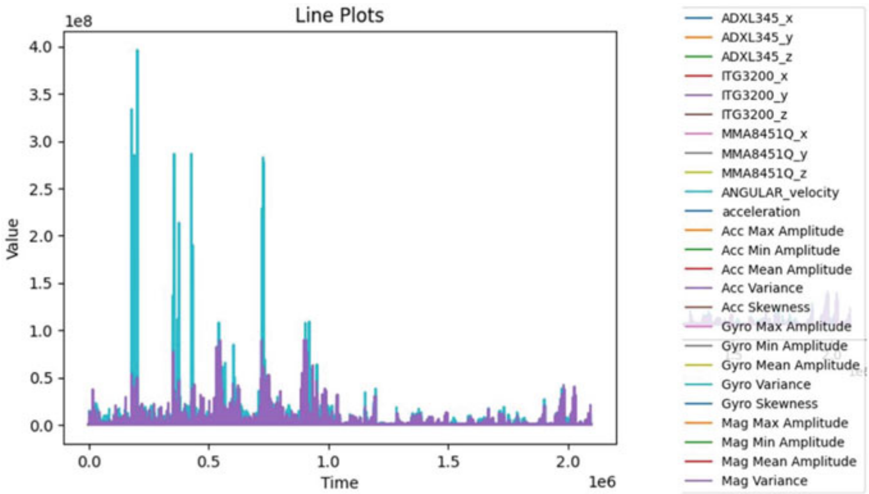


Fig. 14 Sisfall dataset with 25 features (Line Plots)

- Mean amplitude: Mean_AngularVelocity = (1/N) * sum($g(i)$).
- Variance: Variance_AngularVelocity = (1/N) * sum(($g(i)$ - Mean_AngularVelocity) ^2).
- Kurtosis: Kurtosis_AngularVelocity = (1/N) * sum((($g(i)$ - Mean_AngularVelocity)/sqrt (Variance_AngularVelocity)) ^4).
- Skewness: Skewness_AngularVelocity = (1/N) * sum((($g(i)$ - Mean_AngularVelocity)/sqrt (Variance_AngularVelocity)) ^3).

For a magnetometer signal: Let us assume that the magnetometer signal is denoted by $m(i)$, where i ranges from 1 to N (total number of samples).

- Maximum amplitude: Max_MagneticField = max($m(i)$).
- Minimum amplitude: Min_MagneticField = min($m(i)$).
- Mean amplitude: Mean_MagneticField = (1/N) * sum(m(i)).
- Variance: Variance_MagneticField = (1/N) * sum(($m(i)$ - Mean_MagneticField.

- Angular Velocity: Angular Velocity (ω) = $\Delta\theta/\Delta t$.
  - In a three-dimensional scenario, where an object can rotate around multiple axes, the formula for angular velocity (ω) is represented as a vector:

$$\omega = (\omega x, \omega y, \omega z),$$

where ωx represents the angular velocity around the x-axis, ωy represents the angular velocity around the y-axis, and ωz represents the angular velocity around the z-axis. The values of ωx, ωy, and ωz can be calculated using differentiation (taking the rate of change) of the respective angular displacement with respect to time.

A low-pass filter is applied to the angular velocity signals to remove high-frequency noise or vibrations. The cutoff frequency determines the point at which the filter starts attenuating the high-frequency components. A Butterworth filter is used, which provides a maximally flat response in the passband. The 'filtfilt' function is used to apply the filter to the angular velocity signals and ensure zero-phase filtering. The 'b' and 'a' coefficients of the filter are obtained from the [8] 'butter' function. The filtered angular velocity signals for each axis are concatenated into a single array 'w' using the np concatenate function. The Euclidean norm (magnitude) of the vector is computed and assigned to a new feature called 'ANGULAR_velocity'.

Using the aforementioned formulas, we were able to identify the top 25 hand-engineered features (Fig. 12). These include the accelerometer (ADXL345) axes: 'ADXL345_x', 'ADXL345_y', 'ADXL345_z', the gyroscope (ITG3200) axes: 'ITG3200_x', 'ITG3200_y', 'ITG3200_z', and the magnetometer (MMA8451Q) axes: 'MMA8451Q_x'. Additionally, the following features (Fig. 13) were also included: Accelerometer Maximum Amplitude: 'Acc Max Amplitude', Accelerometer Minimum Amplitude: 'Acc Min Amplitude', Accelerometer Mean Amplitude: 'Acc Mean Amplitude', Accelerometer Variance: 'Acc Variance', Accelerometer Skewness: 'Acc Skewness', Gyroscope Maximum Amplitude: 'Gyro Max Amplitude', Gyroscope Minimum Amplitude: 'Gyro Min Amplitude', Gyroscope Mean Amplitude: 'Gyro Mean Amplitude', Gyroscope Variance: 'Gyro Variance', Gyroscope Skewness: 'Gyro Skewness', Magnetometer Maximum Amplitude: 'Mag Max Amplitude', Magnetometer Minimum Amplitude: 'Mag Min Amplitude', Magnetometer Mean Amplitude: 'Mag Mean Amplitude', and Magnetometer Variance: 'Mag Variance'. The same feature extraction technique was utilized for the UMA Fall dataset.

Proposed Deep Learning Model

In our proposed method for detecting falls using the Sisfall dataset and UMA Fall dataset, we utilize a 1D convolutional neural network (1DCNN) model. 1DCNN (convolutional neural network): This model utilizes convolutional layers to extract relevant features from the input data. By applying filters and aggregation operations, the CNN learns spatial patterns and captures crucial data for fall detection [4, 15, 17, 19].

One-Dimensional Convolutional Neural Network (1DCNN)

1D CNNs operate on sequential data with a single dimension, such as time series or text. They use convolutional layers to extract features from the input data, similar to other CNNs. In 1D CNNs, the convolutional operation is performed along the temporal or spatial axis of the data, as opposed to across two-dimensional spatial dimensions, as in image data. In a 1D CNN, an input sequence is convolved with a filter of a fixed size by gliding over the sequence and computing a dot product between the filter weights and the values in the current window. This procedure generates a feature map that emphasizes the presence of particular patterns or features in the input sequence. A 1D CNN can capture various levels of granularity in the input data by employing multiple filters of varying sizes. Typically, the resultant feature maps are transmitted through activation functions and aggregating layers to further process the features and reduce the data's dimension. On the extracted features, one or more fully connected layers may be used to accomplish classification or regression [4, 17–19].

One-dimensional convolutional neural networks (1DCNNs) can be used for fall detection. The input signal is first passed through a convolutional layer, which performs feature extraction. The output of the convolutional layer is then passed through a max-pooling layer, which down samples the feature map. Finally, the output of the pooling layer is passed through a fully connected layer for classification.

The output of the convolutional layer can be computed using the following equation:

$$y[i] = b + \sum(j = 0 \text{ to } m - 1) w[j] x[i + j], \tag{1}$$

where $y[i]$ is the output at position i, b is the bias term, $w[j]$ is the weights of the filter, $x[i + j]$ is the input values, and m is the size of the filter.

The output of the max-pooling layer can be computed using the following equation:

$$y[i] = \max(x[\text{is} : \text{is} + k]), \tag{2}$$

where $y[i]$ is the output at position i, $x[\text{is: is} + k]$ is the input segment of length k starting at position $i*s$, and s is the stride.

The output of the fully connected layer can be computed using the following equation:

$$y = f\left(b + \sum(i = 0\, to\, n - 1)\; w[i]\, x[i]\right), \tag{3}$$

where y is the output, b is the bias term, $w[i]$ is the weights, $x[i]$ is the inputs, n is the number of inputs, and f is the activation function. In this study, we employ a 1D convolutional neural network (CNN) model that is well-suited for extracting unique features from datasets with present window lengths. The size of the testing set is 20% of the total dataset. StandardAero is utilized to normalize the input features so that the data have a mean of zero and a standard deviation of one. We transform the input data into a 3D tensor so that it can be processed by the 1D CNN. The tensor has three dimensions, which include sample count, time increments, and characteristics.

Algorithm

This algorithm describes the steps taken in the provided code to train a CNN model for fall detection and evaluate its efficacy.

- Import required libraries: pandas, numpy, sklearn, keras, matplotlib.
- Load the dataset and split it into input (X) and output (y) variables.
- Split the data into training and testing sets using train_test_split() from sklearn.
- Scale the input features using StandardScaler from sklearn.
- Reshape the input data to a 3D tensor for use with 1D CNN.
- Build a 1D CNN model using Sequential () from keras.
- Add Conv1D and MaxPooling1D layers to the model.
- Flatten the output from the Conv1D layer and add Dense layers to the model.
- Compile the model using binary_crossentropy loss function, adam optimizer, and accuracy metric.
- Train the model using fit() from keras.
- Plot the training and validation accuracy and loss using matplotlib.
- Make predictions on the testing data using predict () from keras.
- Convert the probabilities to class labels.
- Print the classification report using classification_report() from sklearn.
- Plot the confusion matrix using confusion_matrix() from sklearn and matplotlib.

Explanation of the algorithm

Data Preparation:

Imports required libraries such as Pandas, NumPy, Sklearn, and Keras. This is the initial step. The next step in the data preparation process is the import of the data. Following that, the data are separated into the two variables that were input.

Data Preprocessing:

The next stage is to preprocess the data, where the data is divided into training and testing sets with the help of the train test split function from sklearn. Eighty % of the data will be used for training, while the remaining 20% will be used for testing. StandardScaler, which is included in sklearn, is used to do the scaling on the input features. In this stage, the features are standardized by first calculating the mean and then scaling the mean down to the unit variance. The input data is reformatted into a three-dimensional tensor so that it can be processed by the one-dimensional CNN model.

Building the 1D CNN Model:

In order to construct the 1D CNN model (Fig. 15), the model architecture is specified by utilizing the Sequential API that is provided by Keras. Two convolutional layers have been added, each with 128 and 256 filters correspondingly. After each convolutional layer comes a max-pooling layer, which helps minimize the spatial dimensions of the data. The output is then flattened by the model, and it is run through a dense layer that has 64 units and a dropout layer in order to prevent overfitting. In the end, a dense layer that only contains a single unit and uses sigmoid activation is added in order to do binary classification. The Adam optimizer is used in the compilation process, along with binary cross-entropy loss.

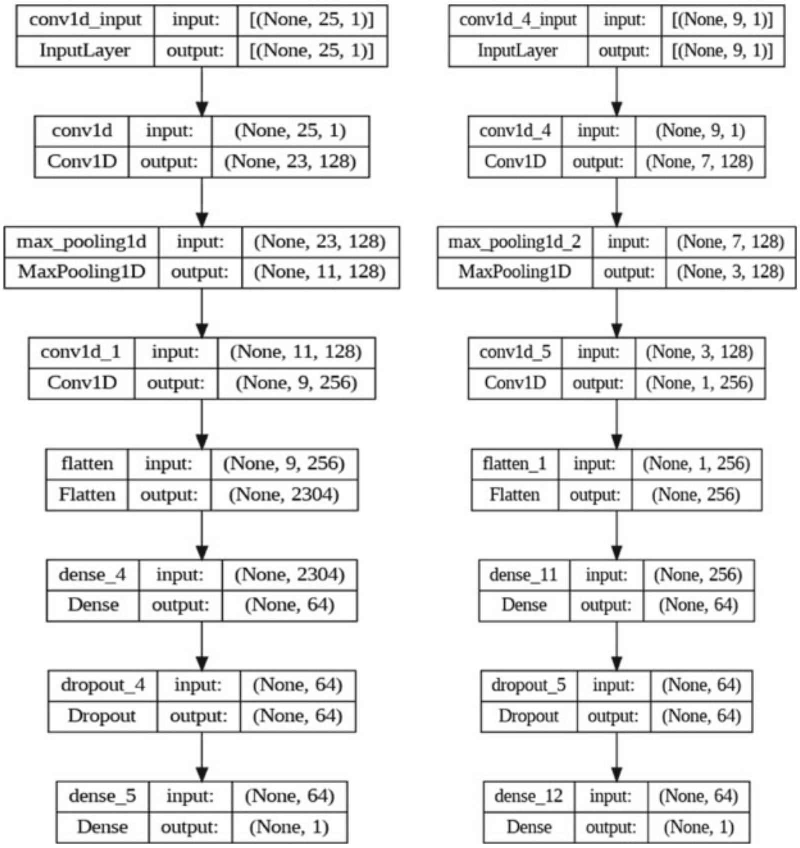


Fig. 15 CNN model building

Training and Testing the Model:

The training of the model is carried out using the fit technique with a batch size of 32 and a total of 10 epochs. Matplotlib is used to create plots of the training and validation accuracies as well as the losses.

Evaluation and Performance Analysis:

Model Analysis Metrix

A classification model's efficacy is evaluated using the metrics True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). These metrics provide granular insight into the model's ability to correctly classify instances into their respective classifications.

- True Positive (TP): This metric represents the number of instances correctly classified by the model as positive (class 1) instances. It measures the number of instances in which the model correctly predicted the positive class.
- False Positive (FP): This metric indicates the number of instances incorrectly classified as positive (class 1) by the model. It measures the number of instances in which the model predicted the positive class, but the actual class was class 0 (negative).
- True Negative (TN): This metric represents the number of instances correctly classified by the model as negative (class 0). It measures the number of times the model correctly predicted the negative class.
- False Negative (FN): This metric represents the number of instances improperly classified by the model as negative (class 0). It quantifies the number of situations in which the model predicted a negative class, but the actual class was positive (class 1).
- These metrics are used to calculate additional performance metrics, including accuracy, precision, recall, and F_1-score.
- Precision: Precision is the ratio of true positives (TP) to the sum of true positives (TP) and false positives (FP). It measures the proportion of correctly identified positive instances among all predicted positive instances.
- Recall: Recall is the ratio of true positives (TP) to the sum of true positives (TP) and false negatives (FN). It measures the proportion of correctly identified positive instances among all actual positive instances.
- F_1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance by considering both precision and recall. F_1-score is calculated as (2 * Precision * Recall)/(Precision + Recall).
- Support: Support refers to the number of instances for each class in the dataset. It can be represented by the sum of true positives (TP) and false negatives (FN) for a specific class.
- Accuracy: Accuracy is the ratio of correct predictions (sum of true positives and true negatives) to the total number of predictions. It measures the overall correctness of the model's predictions.
- Macro avg: Macro average calculates the average precision, recall, and F_1-score across all classes. It treats each class equally, regardless of its support. To calculate macro average precision, recall, and F_1-score, you would take the average of the respective metric values for each class.
- Weighted avg: Weighted average calculates the average precision, recall, and F_1-score, taking into account the support of each class. It gives more weight to the metrics of the class with higher support.

1DCNN Model for Sisfall Dataset

After applying the trained model to the testing data of the Sisfall dataset and generating predictions, a threshold of 0.5 is used to transform the projected probabilities into class labels.

The 1DCNN model demonstrated an accuracy score of 89% for the normal Sisfall dataset and 91% for the Sisfall dataset with 25 features. This suggests that the model accurately predicted 89% of outcomes in the Sisfall dataset with nine features and 91% of outcomes in the Sisfall dataset with nine features. In (Fig. 16) is a plot of training loss and validation loss over epoch, training accuracy and validation accuracy over epoch, and a confusion matrix of 0, 1 for Not fall and fall situations for Sisfall Dataset with 9 features, also (Fig. 17) show for Sisfall dataset with 25 features.

Comparing 1DCNN Models with 9 Features and 25 Features Using the Sisfall Dataset

Model 2, the 1DCNN with 25 features, demonstrates slightly superior performance compared to Model 1, the 1DCNN with 9 features, in terms of precision, recall, and F1-score for both classes. It accomplishes greater precision and recall, resulting in a higher F1-score. Both models exhibit high accuracy, with Model 2 obtaining a slightly higher accuracy of 0.91 than Model 1, which achieves an accuracy of 0.89. The macro and weighted average metrics for Model 2 are also greater, indicating a superior performance across all classifications. On the basis of these results, it can be

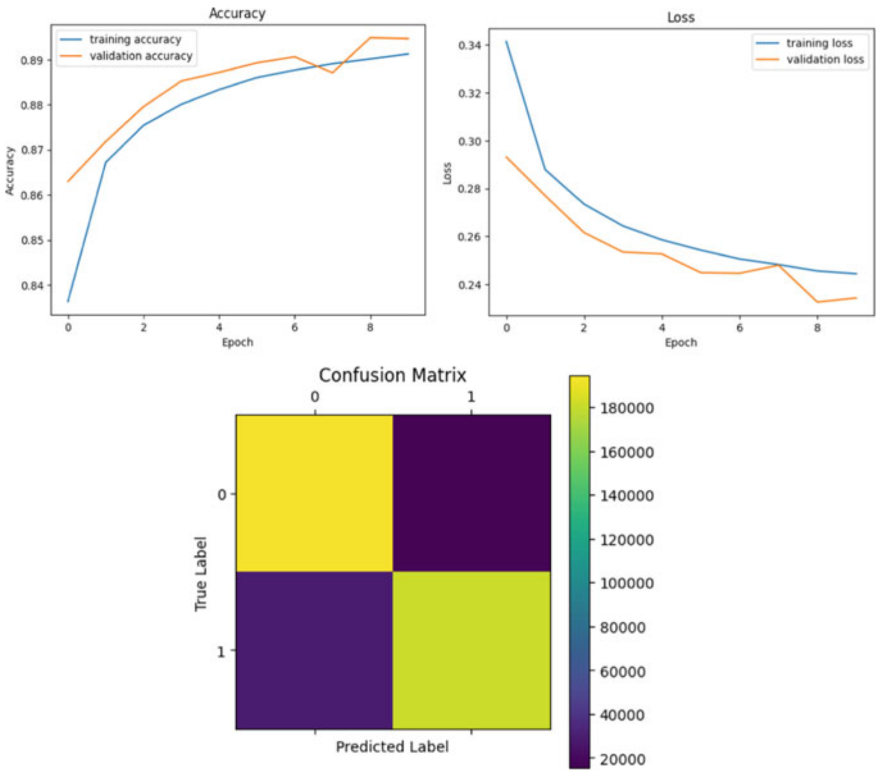


Fig. 16 1DCNN Model Matrix Sisfall with nine features

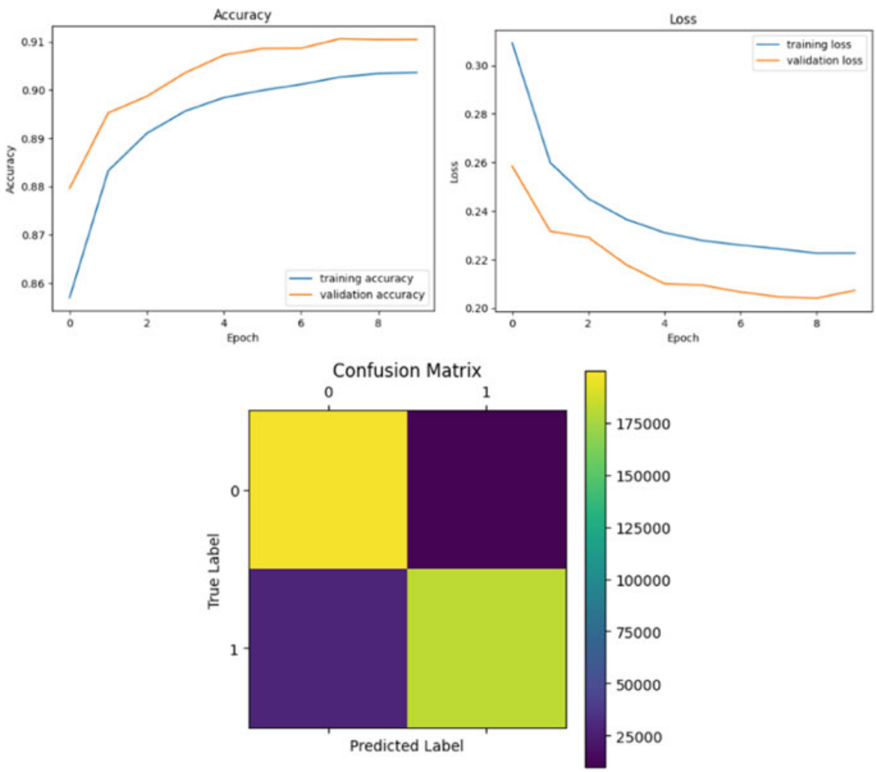


Fig. 17 CNN model matrix Sisfall dataset with 25 features

concluded that the CNN model with 25 features outperforms the CNN model with nine features on the Sisfall dataset.

Model 1: 1DCNN with Sisfall Dataset 9 Features

- Class 0 has a precision of 0.87 and class 1 has a precision of 0.92.
- Class 0 has a recall F_1 rate of 0.93, while class 1 has a recall F_1 rate of 0.86.
- Class 0 has an F_1-score of 0.90, whereas class 1 has an F_1-score of 0.89.
- Accuracy: 0.89 is the model's accuracy.
- The average precision, recall F_1, and F_1-score at the macro level are 0.90, 0.89, and 0.89, respectively.
- The weighted average precision is 0.90, recall F_1 is 0.89, and the F_1-score is also 0.89 (Table 1).

Model 2: 1DCNN with Sisfall Dataset 25 Features

- Class 0 has a precision of 0.88, whereas class 1 has a precision of 0.95.
- Class 0 has a recall F_1 rate of 0.95, while class 1 has a recall F_1 rate of 0.87.
- Class 0 has an F_1-score of 0.91, while class 1 has an F_1-score of 0.91.

Table 1 1DCNN with nine features Sisfall dataset

	Precision	Recall F_1	F_1-score	Support
0	0.87	0.93	0.90	209,768
1	0.92	0.86	0.89	209,662
Accuracy			0.89	419,430
Macro avg.	0.90	0.89	0.89	419,430
Weighted avg.	0.90	0.89	0.89	419,430

- The model is accurate to 0.91 degrees.
- The average precision, recall F_1, and F_1-score at the macro level are 0.91, 0.91, and 0.91, respectively.
- The weighted average precision is 0.91, the recall F_1 is 0.91, and the F_1-score is 0.91 (Table 2).

1DCNN Model for UMA Fall Dataset

For the UMA Fall dataset with nine features and for the UMA Fall dataset with 25 features, the 1DCNN model exhibited an accuracy score of 90% and 92%, respectively.

This indicates that 90% of outcomes in the UMA Fall dataset with 9 features and 91% of outcomes in the UMA Fall dataset with 25 characteristics were correctly predicted by the model. For the UMA Fall dataset with nine features, a plot of training loss and validation loss over epoch, training accuracy and validation accuracy over epoch, and a confusion matrix of 0, 1 for Not fall and fall circumstances are shown in (Fig. 18). UMA fall with 25 features is shown in (Fig. 19).

Comparison of 1D CNN Models with 9 Features on the UMA Fall Dataset

Model 2, the 1D CNN with 25 features, outperforms Model 1, the 1D CNN with 9 features, in terms of precision, recall, and F1-score for both classes. It achieves higher precision, recall, and F1-score values for both classes. Both models show high accuracy, with Model 2 achieving a slightly higher accuracy of 0.92 compared to Model 1 with an accuracy of 0.90. The macro and weighted average metrics for Model 2 are also higher, indicating better overall performance across all classes.

Table 2 1DCNN with 25 features' Sisfall dataset

	Precision	Recall F_1	F_1-score	Support
0	0.88	0.95	0.91	209,768
1	0.95	0.87	0.91	209,662
Accuracy			0.91	419,430
Macro avg.	0.91	0.91	0.91	419,430
Weighted avg.	0.91	0.91	0.91	419,430

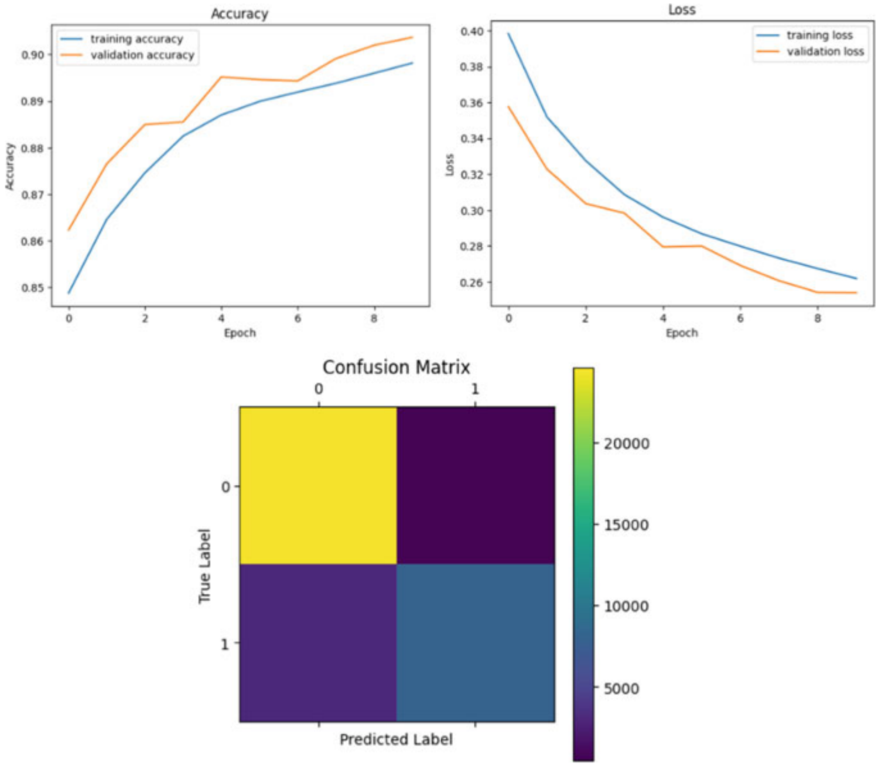


Fig. 18 1DCNN model matrix UMA fall with nine features

Based on these results, it can be concluded that the 1D CNN model with 25 features performs better than the 1D CNN model with nine features on the UMA Fall dataset.

Model 1: 1D CNN with nine Features

- Precision: For class 0, the precision is 0.89, and for class 1, it is 0.94.
- Recall: For class 0, the recall F_1 is 0.90, and for class 1, it is 0.73.
- F1-score: For class 0, the F_1-score is 0.93, and for class 1, it is 0.82.
- Accuracy: The accuracy of the model is 0.90.
- Macro average: The macro average precision is 0.92, recall F_1 is 0.85, and F_1-score is 0.88.
- Weighted average: The weighted average precision is 0.91, recall F_1 is 0.90, and F_1-score is 0.90 (Table 3).

Model 2: 1D CNN with 25 Features

- Precision: For class 0, the precision is 0.90, and for class 1, it is 0.95.
- Recall: For class 0, the recall F_1 is 0.98, and for class 1, it is 0.76.
- F1-score: For class 0, the F_1-score is 0.94, and for class 1, it is 0.85.

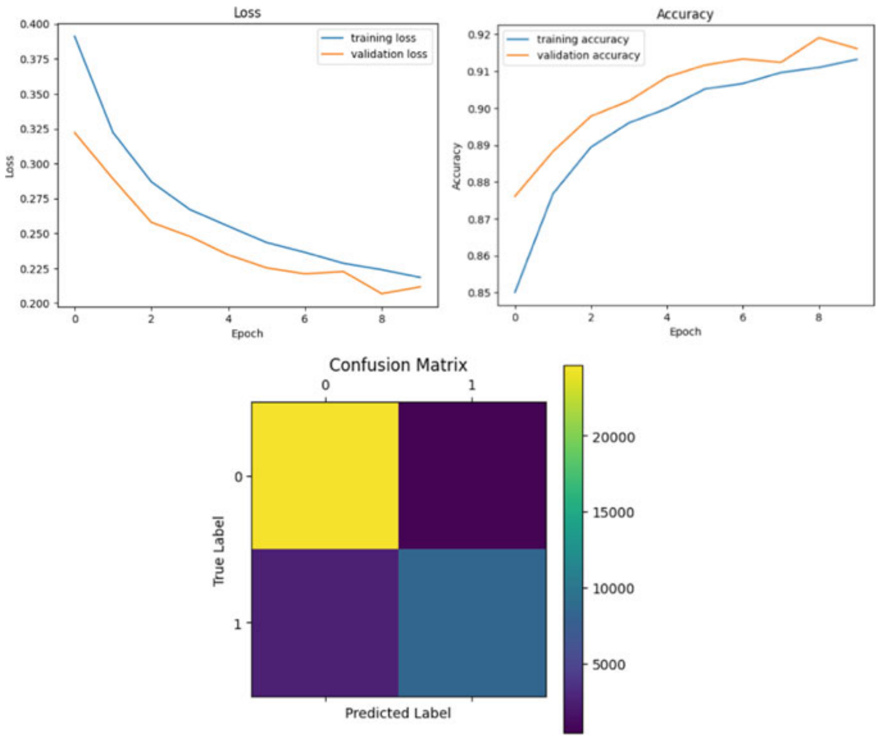


Fig. 19 1DCNN model matrix UMA Fall with 25 features

Table 3 1DCNN with 9 features UMA Fall dataset

	Precision	Recall F_1	F_1-score	Support
0	0.89	0.90	0.93	25,066
1	0.94	0.73	0.82	10,990
Accuracy			0.90	36,056
Macro avg	0.92	0.85	0.88	36,056
Weighted avg	0.91	0.90	0.90	36,056

- Accuracy: The accuracy of the model is 0.92.
- Macro average: The macro average precision is 0.93, recall F_1 is 0.87, and F_1-score is 0.89.
- Weighted average: The weighted average precision is 0.92, recall F_1 is 0.92, and F_1-score is 0.91 (Table 4).

Table 4 1DCNN with 25 features UMA Fall dataset

	Precision	Recall F_1	F_1-score	Support
0	0.90	0.98	0.94	25,066
1	0.95	0.76	0.85	10,990
Accuracy			0.92	36,056
Macro avg.	0.93	0.87	0.89	36,056
Weighted avg.	0.92	0.92	0.91	36,056

4.3 Experiment Setup

In this study, machine learning experiments were conducted using Google Colab and an HP laptop. The HP laptop featured a ninth-generation i7 processor, 16 gigabytes of RAM, and a one-terabyte solid-state drive. Our development environment for executing machine learning tasks was Jupyter Notebook. Cloud-based platform Google Colab gave us access to potent computational resources. It allowed us to utilize Google's high-performance GPUs and TPUs to expedite our machine learning experiments. We utilized Colab's collaborative features to readily share and collaborate with other researchers on our code and findings. We utilized the HP laptop's local computational capability and storage capacity for specific experiments. The i7 ninth-generation CPU ensured the efficient processing of our machine learning algorithms, while the 16 gigabytes of RAM enabled us to manage large datasets and intricate models. The one terabyte SSD was sufficient for storing our datasets, models, and intermediate results. Our primary development environment was Jupyter Notebook, which allowed us to write and execute code in an interactive and reproducible manner. To implement and evaluate our models, we utilized numerous machine learning libraries and frameworks, such as TensorFlow and Keras. The adaptability and extensive data visualization support of Jupyter Notebook assisted us in analyzing and interpreting our experimental results. By combining Google Colab and our HP laptop, we obtained a comprehensive experimental configuration that enabled us to conduct effective machine learning research. This configuration provided us with the flexibility to utilize both cloud-based resources and local computational capacity, allowing us to address a variety of research challenges and gain insightful knowledge.

4.4 Result Analysis

In this detailed comparison of various models (Fig. 20), their respective accuracies are examined in a classification task. Ramachandran et al. [7] employed Ordinal Logistic Regression, achieving an accuracy of 84.1%. Vallabh et al. [11] used KNN, reaching an accuracy of 87.5%. Chelli and Pätzold [12] utilized ANN, obtaining a higher accuracy of 87.8%. Miawarni, Herti, et al. [13] applied SVM, resulting in an accuracy of 84.62%. Rashid et al. [14] introduced Cubic SVM with an accuracy

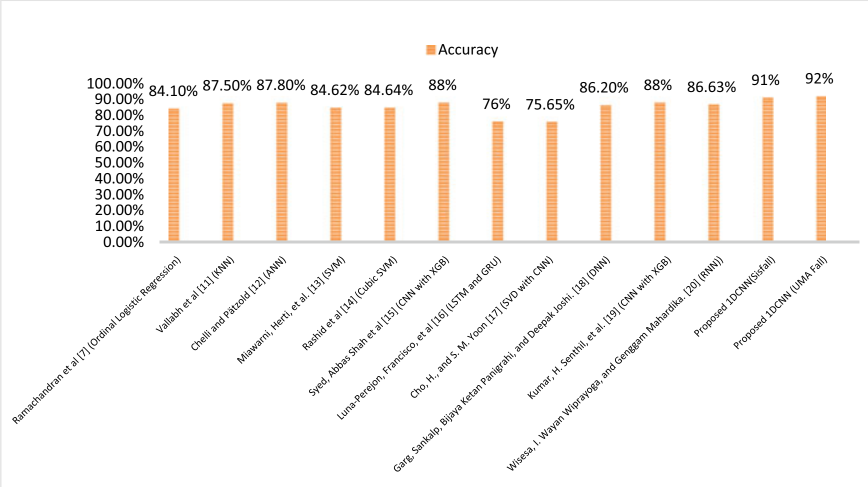


Fig. 20 Comparison with others

of 84.64%. Syed, Abbas Shah et al. [15] combined CNN with XGBoost, achieving an accuracy of 88%. Luna-Perejon et al. [16] used LSTM and GRU, obtaining an accuracy of 76%. Cho and Yoon [17] combined SVD with CNN, reaching an accuracy of 75.65%. Garg et al. [18] employed DNN, achieving an accuracy of 86.2%. Kumar et al. [19] also combined CNN with XGBoost, resulting in an accuracy of 88%. Wisesa et al. [20] used RNN, achieving an accuracy of 86.63%. Additionally, the proposed 1DCNN (Sisfall) model demonstrated an accuracy of 91%, showcasing its ability to accurately detect falls. The 1DCNN (UMA Fall) model achieved an even higher accuracy of 92%, indicating its superior performance compared to the other models (Table 5).

4.5 Limitations

Limited Resources: The development and maintenance of an efficient Fall Detection System (FDS) requires considerable resources. Obtaining extensive and diverse datasets, undertaking field trials, and ensuring data privacy and security all require financial investments. Additionally, expertise and personnel are required for data acquisition, model development, and system deployment. These aspects may not be fully realized due to limited resources, which may have an effect on the system's overall performance and scalability.

Limited Computational Power: Implementing sophisticated deep learning models, such as 1DCNNs, often demands significant computational power. Training and executing these models efficiently can be computationally costly and may necessitate high-performance hardware, such as GPUs or specialized processors. The inability

Table 5 Comparative result table

Name of Model	Accuracy (%)
Ramachandran et al. [7] (Ordinal Logistic Regression)	84.1
Vallabh et al. [11] (KNN)	87.5
Chelli and Pätzold [12] (ANN)	87.8
Miawarni et al. [13] (SVM)	84.62
Rashid et al. [14] (Cubic SVM)	84.64
Syed et al. [15] (CNN with XGB)	88
Luna-Perejon et al. [16] (LSTM and GRU)	76
Cho and Yoon [17] (SVD with CNN)	75.65
Garg et al. [18] (DNN)	86.2
Kumar et al. [19] (CNN with XGB)	88
Wisesa et al. [20] (RNN))	86.63
Proposed 1DCNN (Sisfall)	91
Proposed 1DCNN (UMA Fall)	92

to investigate and utilize more complex models can be hampered by a system's limited computational capacity, thereby compromising its potential accuracy and performance.

To address these constraints, resource management and strategic planning are required. Effectively allocating resources, such as prioritizing data collection efforts based on available funding, can mitigate the effect of limited resources. Exploring optimization techniques, model compression methods, or utilizing cloud computing resources can assist in circumventing computational power limitations. Although these constraints present challenges, it is essential to acknowledge them and pursue solutions that maximize the system's potential within the constraints available. Even with limited resources and computational capacity, it is possible to develop an alert system with effective performance and usability by maximizing available resources and investigating alternative approaches.

5 Future Scope

Future emphasis should be placed on the following areas to improve the effectiveness and usability of the Fall Detection System (FDS):

Integrating the alert system with a cloud infrastructure is a necessary and logical step in developing a scalable and effective solution. The advantages of cloud connectivity include scalability, dependability, and accessibility. The system can achieve real-time monitoring, seamless integration with other systems, and remote access from a variety of devices and locations by utilizing cloud hosting. This permits the generation of timely alerts and improves the system's overall functionality. To facilitate cloud integration, a comprehensive system for applications that can connect with smartphone wearable devices like smart watches and smart bands must be developed. Moreover, integration with ambient devices is necessary. This integrated system enables the transmission of location data to caregivers via IoT technology in the event of an outdoor fall. In such situations, the alert system can quickly identify the nearest caregiver and notify them of the exact location, allowing for immediate intervention. Additionally, if a fall occurs inside the home, the system should be able to alert family members or other individuals who are present. This ensures that the individual in need can receive immediate assistance. By leveraging cloud infrastructure and integrating multiple devices and systems, the proposed solution improves the alert system's effectiveness and efficiency. The ability to seamlessly connect wearable devices, utilize IoT technology for outdoor fall detection, and notify caregivers and family members in real time greatly enhances the system's overall safety and response capabilities.

Expand and diversify the dataset to enhance the model's ability to generalize, it is essential to acquire a more extensive and diverse dataset. By collecting information from a variety of sources, environments, and demographics, the model will be able to manage a wider variety of real-world scenarios. This can include information from various sensors, locations, and user profiles, taking into account age, gender, and physical abilities. Moreover, data augmentation techniques can be utilized to artificially increase the dataset's size and diversity, emulating various scenarios and enhancing the models generalizability.

Utilize advanced deep learning models, the efficacy of the system can be improved by incorporating more complex and sophisticated deep learning models. Recurrent neural networks (RNNs), attention-based models, and transformer models have demonstrated superior performance in time series analysis and sequential data processing domains. Exploring these models enables the identification of intricate patterns and long-term dependencies within the data, resulting in enhanced accuracy and predictive abilities.

Real-World testing and field trials for assure the alert system's practical applicability, it is essential to conduct exhaustive real-world testing and field trials. Evaluating the model's performance under real-world conditions provides invaluable insight into its usefulness and efficacy. Field evaluations can help identify any limitations or enhancement areas that must be addressed, ensuring that the system performs accurately and reliably in real-world situations.

Security and Privacy Considerations, it is of the utmost necessity to ensure the security and privacy of the collected data. Implementing comprehensive data anonymization techniques and adhering to applicable privacy regulations will increase stakeholder and user confidence. Prioritizing data security and privacy protects the integrity and secrecy of personal information, thereby enhancing user confidence in the system.

6 Conclusion

In conclusion, the study focuses on enhancing elderly fall detection systems through the integration of deep learning and IoT technologies. The results demonstrate the impressive effectiveness of the proposed 1DCNN models, accurately detecting falls in different datasets with accuracies of 91% and 92% on the Sisfall and UMA Fall datasets, respectively.

A significant aspect highlighted in the study is the incorporation of fatality rates into the alert system. This consideration enables caregivers and family members to be promptly notified in critical situations, allowing for timely assistance and potentially saving lives. This proactive approach adds an extra layer of safety and support to the fall detection system, making it more effective in real-world scenarios. The thesis provides valuable insights into the field of fall detection by introducing novel models that outperform existing approaches in terms of accuracy. Future research and development should focus on further refining deep learning algorithms, incorporating diverse datasets, integrating advanced sensor technologies, and considering fatality rates to further enhance the system's accuracy, applicability, and reliability.

Continued efforts in research and development are essential to optimize the proposed fall detection models and address any limitations. Successfully integrating these technologies into healthcare and assisted living environments will significantly improve the safety and well-being of individuals at risk of falls. By incorporating fatality rates into the alert system, the fall detection technology can promptly notify caregivers or family members in critical situations, ensuring timely assistance and potentially saving lives. This crucial feature reinforces the system's overall effectiveness and its potential positive impact on vulnerable individuals' lives. Ultimately, advancements in fall detection technology, along with the integration of fatality rates, have the potential to enhance the overall quality of life for at-risk individuals by providing timely assistance and minimizing the risks associated with falls. The findings of this research contribute to the ongoing development of reliable and effective fall detection systems, further improving the safety and well-being of vulnerable individuals in real-world settings.

References

1. Salah OZ, Selvaperumal SK, Abdulla R (2022) Accelerometer-based elderly fall detection system using edge artificial intelligence architecture. Int J Electr Comput Eng 12(4):4430
2. Tanwar R et al (2022) Pathway of trends and technologies in fall detection: a systematic review. In: Healthcare, vol 10, no 1. Multidisciplinary Digital Publishing Institute
3. Xia K, Huang J, Wang H (2020) LSTM-CNN architecture for human activity recognition. IEEE Access 8:56855–56866
4. Thakur D, Biswas S (2022) Attention-Based deep learning framework for hemiplegic gait prediction with smartphone sensors. IEEE Sens J 22(12):11979–11988
5. Casilari E, Oviedo-Jiménez MA (2015) Automatic fall detection system based on the combined use of a smartphone and a smartwatch. PloS one 10(11):e0140929

6. Chandak A, Chaturvedi N (2022) Machine-learning-based human fall detection using contact- and noncontact-based sensors. Comput Intell Neurosci 2022
7. Ramachandran A et al (2018) Machine learning-based techniques for fall detection in geriatric healthcare systems. In: 2018 9th International conference on information technology in medicine and education (ITME). IEEE
8. Hussain F et al (2019) An efficient machine learning-based elderly fall detection algorithm. arXiv preprint arXiv:1911.11976
9. Shipkovenski G et al (2022) Accelerometer based fall detection and location tracking system of elderly. In: 2022 International symposium on multidisciplinary studies and innovative technologies
10. Toda K, Shinomiya N (2019) Machine learning-based fall detection system for the elderly using passive RFID sensor tags. In: 2019 13th International conference on sensing technology (ICST). IEEE
11. Vallabh P et al (2016) Fall detection using machine learning algorithms. In: 2016 24th international conference on software, telecommunications and computer networks (SoftCOM). IEEE
12. Chelli A, Pätzold M (2019) A machine learning approach for fall detection and daily living activity recognition. IEEE Access 7:38670–38687
13. Miawarni H et al (2022) Enhancing classification of elderly fall detection system using tuned RBF-SVM. In: 2022 IEEE international conference on imaging systems and techniques (IST). IEEE
14. Rashid FA, Sandrasegaran K, Kong X (2021) Simulation of SisFall dataset for fall detection using matlab classifier algorithms. In: 2021 12th International symposium on parallel architectures, algorithms and programming (PAAP). IEEE
15. Syed AS et al (2022) A deep convolutional neural network-xgb for direction and severity aware fall detection and activity recognition. Sensors 22(7):2547
16. Luna-Perejon F et al (2019) An automated fall detection system using recurrent neural networks. In: Proceedings 17 artificial intelligence in medicine: 17th conference on artificial intelligence in medicine, AIME 2019, Poznan, Poland, June 26–29, 2019. Springer International Publishing
17. Cho H, Yoon SM (2019) Applying singular value decomposition on accelerometer data for 1D convolutional neural network based fall detection. Electron Lett 55(6):320–322
18. Garg S, Panigrahi BK, Joshi D (2019) An accelerometer based fall detection system using deep neural network. In: 2019 IEEE 5th international conference for convergence in technology (I2CT). IEEE
19. Kumar HS et al (2022) Fall detection and activity recognition using hybrid convolution neural network and extreme gradient boosting classifier. In: 2022 International conference on innovative computing, intelligent communication and smart electrical systems (ICSES). IEEE
20. Wisesa IWW, Mahardika G (2019) Fall detection algorithm based on accelerometer and gyroscope sensor data using recurrent neural networks. In: IOP conference series: earth and environmental science, vol 258, no 1. IOP Publishing
21. Sucerquia A, López JD, Vargas-Bonilla JF (2017) SisFall: a fall and movement dataset. Sensors 17(1):198
22. Santoyo-Ramón JA, Casilari E, Cano-García JM (2018) Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection with supervised learning. Sensors 18(4):1155