Journal of Artificial Intelligence, Virtual Reality, and Human-Centered Computing
ISSN: 3069-2318

PRIME OPEN ACCESS

Volume 1, Issue 1

Research Article

Date of Submission: 17 Nov, 2025
Date of Acceptance: 16 Dec, 2025
Date of Publication: 22 Dec, 2025

Comparative Study on Performance of ML Models for Fall Detection in Older People

Mohammadali Sefidi Esfahani[1]* and Mohammad Fattahian[2]

[1]Amirkabir University of Technology, Tehran, Iran

[2]Kharazmi University, Tehran, Iran

***Corresponding Author:** Mohammadali Sefidi Esfahani, Amirkabir University of Technology, Tehran, Iran.

Citation: Esfahani, M. S., Fattahian, M. (2025). Comparative Study on Performance of ML Models for Fall Detection in Older People. *J AI VR Hum Comput,* 1(1), 01-05.

Abstract

Fall detection systems play a crucial role in addressing the significant health concern of elderly falls, a leading cause of health deterioration and mortality. As the aging population grows and life expectancy increases, the development of accessible tools becomes vital for predicting and preventing falls, offering a practical and widely applicable solution in contrast to costly and expertise-dependent assessment tools. In contrast, due to the formidable challenges encountered, a comprehensive investigation into the comparative performance of standard ML models within this field still needs to be explored. This paper proposes a standard pipeline for pre-processing, training, and evaluating ML models for fall detection on the SisFall dataset. We conducted extensive experiments to evaluate the performance of various ML models for fall detection. The results validate the efficiency of the deep model in identifying the time windows in which a fall occurred. Among the deep models, the architecture, including a combination of convolutional neural networks and fully connected layers, outperforms the others by macro-averaged Precision, macro-averaged Recall, and macro-averaged F1-Score of 87.03%, 86.83%, and 86.93%, respectively.

Keywords: Fall Detection, Intelligent Systems, Applied ML in Healthcare

Introduction

In the face of an aging global population, the prevalence of older adults living independently is on the rise, accompanied by an escalating concern for the significant health risks posed by accidental falls [1]. The World Health Organization's studies underscore the alarming incidence of fatal falls among adults over 65, emphasizing the imperative for immediate attention during emergencies at their residences [2]. This vulnerable demographic faces hindrances in seeking aid due to a lack of technology access in rural areas and physical limitations [3]. Automatic fall detection systems emerge as pivotal solutions, enhancing the quality of life for older individuals and providing crucial living assistance [4]. The statistics reveal that 25% of individuals over 65 experience falls annually, a notable increase to 32%-42% for those over 70 [5]. Long-term care institutions report 30%-50% of residents falling each year, intensifying the urgency for effective interventions. Fall Detection Systems (FDSs) play a vital role by detecting falls in real time and triggering remote notifications for timely aid. The importance of fall prevention systems is emphasized by the high incidence of falls among older adults in community and residential care settings [6]. The economic burden of fall-related injuries, totaling billions of dollars, underscores the societal impact [7]. As the aging population grows, the need for accurate fall detection becomes increasingly apparent, offering a lifeline to those at risk of falls and mitigating the physical, psychological, and economic consequences of this pressing health issue.

Retrospective studies in fall detection have scrutinized the efficacy of ML models in fall detection, underscoring their inherent value and significance. However, due to the formidable challenges encountered, a comprehensive investigation into the comparative performance of standard ML models within this field still needs to be explored.

- **Misleading Performance Metrics:** The intricacies of fall detection are intricately entangled with imbalanced characteristics, presenting a substantial challenge for ML models. Their struggle to adeptly recognize and generalize patterns associated with the minority class poses a significant obstacle, thereby diminishing the overall predictive accuracy. Furthermore, conventional performance metrics, including accuracy, may prove misleading within imbalanced datasets, as a model might attain heightened accuracy by predominantly predicting the majority class, notwithstanding its shortcomings in correctly classifying instances from the minority class.

- **Inaccurate Data Segmentation:** The SisFall dataset, a renowned resource in the realm of fall detection, encompasses recordings from 38 volunteers engaged in 34 distinct activities within a controlled scenario. The segmentation of the SisFall dataset for training and evaluation necessitates uniformity in the distribution of activities, volunteers, and labels across these datasets.

- **Necessity of Windowing:** Given the dynamic nature of fall detection, a static approach proves inadequate. Accurately tracking sensor values over time is crucial, emphasizing the need to capture subtle changes. Despite the wealth of research devoted to event detection, a discernible gap exists in exploring the performance of standard ML models tailored explicitly for fall detection [8,9]. In this paper, we investigated the performance and efficiency of standard ML models, including both traditional and deep ML models.

In summary, our main contributions are as follows:

- We proposed a standard pipeline for pre-processing and training ML models for fall detection based on the SisFall dataset.
- We leveraged an efficient data processing approach, reducing the processing time to 33 minutes.
- We conducted a comparative study on the performance and efficiency of ML models for fall detection in older people in a standard setting.

Problem Statement

In this section, we introduce notation and formally articulate the problem of fall detection. Let W = $\{W_1, W_2, \ldots\}$, where $W_i = \{A, I, M\}$ represents a sequence of time windows in this study. In this formulation, $A = (a_x, a_y, a_z)$, $I = (i_x, i_y, i_z)$, and $M = (m_x, m_y, m_z)$ denote values of *ADXL345, ITG3200,* and *MMA8451Q* sensors within the time window Wi respectively. An activity within this time window is designated as a fall if a volunteer experiences a *fall* while engaging in the corresponding activity during this period.

Objective

Given the stream of time windows W, our goal is to devise and train a machine learning model F according to the following mapping:

$$F(\mathrm{W}) \rightarrow y \in \{0,1\} \qquad (1)$$

Here, y represents the binary outcome, indicating whether a fall (y = 1) is detected or not (y = 0). This formalization sets the foundation for our fall detection system's subsequent development and evaluation.

Dataset

Many datasets of human activities containing fall events have been proposed in the literature, like SisFall, UMAFall, and MobiFall [10-12]. We have selected the SisFall dataset, which comprises recordings from 38 volunteers participating in 34 activities within a controlled scenario. These activities encompass 19 distinct Activities of Daily Living (ADLs) and 15 falls, resulting in 4,510 sequences. The dataset undergoes careful annotation and examination by a healthcare expert. Despite lacking explicit indications of event timing within readings, the data is captured at a frequency of 200 Hz using a custom board equipped with accelerometers and a gyroscope fixed on the waist. Volunteers, categorized into young adults (SA) aged 19 to 30 and elderly individuals (SE) aged 60 to 75, engage in diverse motions, including falls, with noteworthy contributions from SE06, a Judo expert. Detailed subject information, including age, height, weight, and gender, enhances the contextual relevance of the dataset. Including sensor characteristics and conversion equations supports research in biomechanics and motion analysis. Annotations within the SisFall dataset classify each activity as either a fall or an ADL yet lack specificity regarding the temporal occurrence of falls or ADLs within the sequence of readings. For instance, annotations may describe an event such as ”collapsing into a chair while trying to stand up” without indicating the timing of the standing up or the collapse. This dataset is a valuable resource for the comprehensive study of human kinetics across diverse age groups and activities, providing insights into daily movements and contributing to advancements in related research fields.

Methodology

In this study, we proposed a comprehensive pipeline consisting of three principal components. The initial phase involves raw data streams to the Data Processor component, wherein the data undergoes preparation procedures tailored for subsequent model training. Following this, the Training & Evaluation component is engaged, wherein the processed data is utilized to construct and train ML models. Concurrently, an evaluation of the models' performance and efficiency is conducted within this component. Ultimately, the outcomes are systematically stored in data frames through the final component, Storing Results.

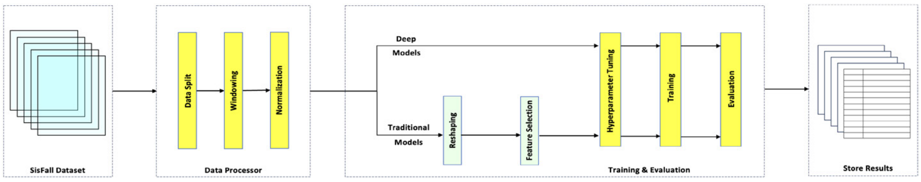

Figure 1: Overview of Our Proposed Pipeline for Fall Detection

Data Processor

The efficacy of ML models in terms of performance, fairness, and reliability is intricately tied to the methodologies employed in data processing. Inaccuracies may arise if developers do not judiciously apply these techniques. As delineated in Section 3.1, the SisFall dataset encompasses 19 distinct ADLs and 15 fall activities involving 38 volunteers with diverse physical characteristics. The literature reveals that in processing such sequential data, the equitable division of the dataset is pivotal, as an inadequate approach may compromise the reliability of outcomes. Specifically, during the dataset segmentation for training and evaluation purposes, a judicious selection of sequence data is imperative, aligning with the distribution of volunteers and activities. Within the Data Processor component, we meticulously undertake the equitable processing of the SisFall dataset, ensuring its preparedness for subsequent stages, particularly the training phase. Subsequent paragraphs expound upon the specific processing steps integral to the Data Processor component. Note that we implemented the Data Processor component equitably and efficiently, and it processed the SisFall dataset in around 33 minutes.

- **Data Split:** The SisFall dataset has a systematic organizational structure where each volunteer is assigned a dedicated directory. The directory captures the sensor readings during various activities and is documented in *.txt* files. As an illustration, the *SE01* directory corresponds to a 71-year-old male volunteer with dimensions of 171cm in height and 102kg in weight. Within this directory, activity-specific records are accessible, such as *D07-SE01-R04.txt,* encompassing data from the fourth trial of *jogging quickly*, an ADL in the SisFall dataset. Employing diverse segmentation techniques, we uniformly divide the dataset based on volunteers' distribution and the nature of the activities, for example, in the case of the *SE01* volunteer undertaking five trials of the *D07* activity. If we want to split the dataset with a ratio of 80%-20%, we randomly select four training trials and one for evaluation. This segmentation approach is consistently applied across all volunteers, ensuring an equitable data division for training and evaluation. We contend that this methodology represents the most equitable data segmentation strategy.

- **Windowing:** In this study, we reorganized the dataset as a time series to dynamically investigate the fall detection problem. To do this, we consider a time window with a constant length of k and use the value of sensors. As an illustration, suppose k = 200, and we exploit the values of all three sensors. In this case, the shape of a single sample of our dataset will be (1,200,9).

- **Normalization:** Normalization of sensor values is imperative due to range variations. This process is applied to the training and test datasets, utilizing the standard deviation and average derived from the training dataset. Normalizing the sensor values gives the model a more nuanced understanding of feature vectors, contributing to enhanced performance. Furthermore, normalization mitigates computation costs, optimizing the model's efficiency across diverse datasets. This essential pre-processing step ensures that the model's training and evaluation processes are standardized, fostering improved generalization and overall performance.

Training & Evaluation

After windowing, we generated 61036 normalized time windows for training and 16375 normalized time windows for evaluation. In this stage, we adopt a supervised learning approach and train a model F that identifies the time windows in which the fall occurred. This model gains the values of sensors, i.e., X, and computes the probability of happening ADLs, i.e., negative class, or referring to a fall, i.e., positive class. Formally, the prediction of model F is performed by solving the following optimization problem:

$$\arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{X}, \mathbf{W}, \mathbf{Y}), \qquad (2)$$

Where L(.) is the loss function and is defined as binary cross entropy. W denotes all parameters that the model learns from the training set, and Y indicates the label of the time windows where one is used for time windows occurring a fall, otherwise zero. Using this setting, we trained various ML models and eventually evaluated their performance based on the Recall, Precision, and f1-score standard classification metrics.

Storing Results

At the final stage, we saved the performance details of models as data frames in the database. These results and information could be used for further analysis and investigation.

Experiments

We conducted a comprehensive series of experiments addressing the following research inquiries. These research questions guide our investigation, and the ensuing experimental results contribute valuable insights into the efficacy and comparative performance of machine learning and deep learning models for fall detection:

- To what extent do machine learning models demonstrate efficiency in the context of fall detection?
- How is the overall performance of deep learning models compared to traditional ML models for fall detection?

Baselines

This work investigated the performance and efficiency of various ML models, including both traditional and deep models. In this section, we delve deeper into the details of these models.

Deep Models

In this study, we investigated the performance of deep Models by implementing three well-known architectures. By deploying these architectures, the study aimed to comprehensively assess the efficacy of standard deep learning models in the context of fall detection:

- **MLP:** We built a fully connected network using a combination of dense layers to classify whether a fall occurred in incoming time windows or not.

- **CNN + MLP:** We implemented a two-segment architecture within this architecture. In the first segment, a combination of convolution and dropout layers extract efficient features from the input time window. In the second segment, we exploited fully connected layers for classification.

- **LSTM + MLP:** Since working on time-series data, we also investigated the efficiency of utilizing LSTM layers for feature extraction. We leveraged the output of LSTM models as an input for a fully connected network, which classifies the incoming time windows.

Traditional Models

While implementing traditional ML models, a crucial preprocessing step involves reshaping the dimension of samples within the training and evaluation datasets, as they are initially incompatible with 3D inputs. We transform the sample dimension from (*n, k, d*) to (*n, k×d*) to achieve this. Following this conversion, the shapes of the training and evaluation datasets become (61036,1800) and (16375,1800), respectively. Additionally, recognizing the inherent challenges posed by high-dimensional data for many traditional machine learning models, we address this concern by reducing the datasets' dimension to 64. This reduction is accomplished by employing a decision tree classifier as a feature selector, ensuring the retention of the most salient features essential for model training and evaluation. After completing these processing steps, we trained the following classifiers and evaluated their performance in fall detection.

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbors

Evaluation Metrics

To evaluate the performance of ML models, we calculated the standard classification metrics known as Precision, Recall, and F1-Score.

Precision: the ratio between the actual falls over all the inputs classified as falls.

$$Pr = \frac{TP}{TP + FP} \tag{3}$$

Recall: the ratio between the number of inputs classified as falls over all the actual falls.

$$Re = \frac{TP}{TP + FN} \tag{4}$$

F1-Score: defined as a combination of Precision and Recall is calculated using the harmonic mean of the two terms.

$$f1 = (\frac{Precision^{-1} + Recall^{-1}}{2})^{-1} \tag{5}$$

Regarding the time windows that refer to the falls, the model must simultaneously achieve high Recall and acceptable Precision. We emphasize the recall metric since we want to identify time windows with falls with great confidence, and we prefer to take advantage of all time windows with label one as much as possible. Although learning discriminative patterns is strenuous in imbalanced data, we hope the fall detection model meets our expectations.

Results

The comprehensive assessment of ML models on the SisFall dataset is presented herein, with a detailed breakdown of results in Table 1, differentiating between fall activities and ADLs. Examination of Table 1 reveals that the MLP and CNN + MLP architectures exhibit superior performance compared to all other baseline models. Key insights derived from the findings are as follows:

- While Random Forest emerges as the top performer for fall detection among traditional ML models, its efficiency lags behind that of deep models. Notably, while optimal within the traditional ML paradigm, ensemble models confront challenges when confronted with high-dimensional data.
- Given the proximate results of deep models, particularly the MLP and CNN + MLP architectures, it can be inferred that a proficient representation of the features matrix can be attained with MLP networks, obviating the necessity for intricate structures.
- In summation, the deep models exhibit commendable performance in fall detection, accurately discerning the temporal windows corresponding to the occurrence of falls.

ML models	Pr_{Fall}	Pr_{ADL}	Re_{Fall}	Re_{ADL}	$F1_{Fall}$	$F1_{ADL}$
Logistic Regression	37.19%	99.16%	82.67%	93.67%	51.31%	96.34%
Decision Tree	55.79%	97.98%	55.63%	98.00%	55.71%	97.99%
Random Forest	91.37%	98.13%	58.16%	99.75%	71.08%	98.93%
KNN	95.62%	97.75%	49.29%	99.89%	65.05%	98.81%
MLP	66.59%	99.44%	87.88%	98.00%	75.77%	98.71%
CNN + MLP	75.21%	98.85%	74.78%	98.88%	75.00%	98.87%
LSTM + MLP	56.45%	99.10%	80.70%	97.17%	66.43%	98.13%

Table 1: Performance of Standard ML Models in Fall Detection

Conclusion

In conclusion, the realm of fall detection in older individuals remains a dynamic area of research, holding significant potential for the well-being of this demographic. This study contributed by conducting a thorough comparative analysis of standard ML models for fall detection. The empirical findings distinctly underscored the efficacy of the CNN + MLP architecture in adeptly discerning and differentiating fall occurrences within time windows. As we look ahead, an intriguing avenue for future research involves exploring point processes and generative models to capture the distribution of sensors' value and learn the conditions where falls occur effectively for early fall detection [13]. Additionally, integrating innovative sampling techniques such as DeepSMOTE stands out as a promising strategy to augment the performance of deep models in the context of fall detection [14]. These proposed directions signify ongoing efforts to advance the Precision and applicability of fall detection systems, ultimately contributing to improved healthcare outcomes for older individuals.

References

1. Lockhart, T. E. (2008). An integrated approach towards identifying age-related mechanisms of slip initiated falls. Journal of Electromyography and Kinesiology, 18(2), 205-217.
2. World Health Organization. (2015). World report on ageing and health. World Health Organization.
3. Fleming, J., & Brayne, C. (2008). Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. Bmj, 337.
4. Huang, C. N., Chiang, C. Y., Chen, G. C., Hsu, S., Chu, W. C., & Chan, C. T. (2010). Fall detection system for healthcare quality improvement in residential care facilities. J. Med. Biol. Eng, 30(4), 247-252.
5. Alshammari, S. A., Alhassan, A. M., Aldawsari, M. A., Bazuhair, F. O., Alotaibi, F. K., Aldakhil, A. A., & Abdulfattah, F. W. (2018). Falls among elderly and its relation with their health problems and surrounding environmental factors in Riyadh. Journal of family and community medicine, 25(1), 29-34.
6. Palestra, G., Rebiai, M., Courtial, E., Giokas, K., & Koutsouris, D. (2017, June). A fall prevention system for the elderly: Preliminary results. In 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 550-551). IEEE.
7. Stevens, J. A., Corso, P. S., Finkelstein, E. A., & Miller, T. R. (2006). The costs of fatal and non-fatal falls among older adults. Injury prevention, 12(5), 290-295.
8. Tongskulroongruang, T., Wiphunawat, P., Jutharee, W., Kaewmahanin, W., Rassameecharoenchai, T., Jennawasin, T., & Kaewkamnerdpong, B. (2022, May). Comparative study on fall detection using machine learning approaches. In 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-4). IEEE.
9. Pandya, B., Pourabdollah, A., & Lotfi, A. (2020). Comparative analysis of real-time fall detection using fuzzy logic web services and machine learning. Technologies, 8(4), 74.
10. Sucerquia, A., López, J. D., & Vargas-Bonilla, J. F. (2017). SisFall: A fall and movement dataset. Sensors, 17(1), 198.
11. Casilari, E., Santoyo-Ramón, J. A., & Cano-García, J. M. (2017). Umafall: A multisensor dataset for the research on automatic fall detection. Procedia Computer Science, 110, 32-39.
12. Vavoulas, G., Pediaditis, M., Chatzaki, C., Spanakis, E. G., & Tsiknakis, M. (2014). The mobifall dataset: Fall detection and classification with a smartphone. International Journal of Monitoring and Surveillance Technologies Research (IJMSTR), 2(1), 44-56.
13. Zeng, F., & Gao, W. (2022, July). Early rumor detection using neural Hawkes process with a new benchmark dataset. In Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies (pp. 4105-4117).
14. Dablain, D., Krawczyk, B., & Chawla, N. V. (2022). DeepSMOTE: Fusing deep learning and SMOTE for imbalanced data. IEEE transactions on neural networks and learning systems, 34(9), 6390-6404.