International Journal of Technology

http://ijtech.eng.ui.ac.id

Research Article

Energy-Efficient TinyML Approach for Wearable Fall Detection on Edge Devices Using Spatial-Temporal Deep Learning

Luong-Cong Duan[1*], Nguyen-Ngoc Minh[1], Truong-Cao Dung[1], Tran-T-Thuc Linh[1]

[1]Faculty of Electronics Engineering I & EDA Lab, Posts and Telecommunications Institute of Technology, Mo Lao, Ha Noi 12110 , Vietnam

*Corresponding author: duanlc@ptit.edu.vn; Tel.: +84-983410746

Abstract: Fall detection systems are critical for elderly care; however, cross-dataset generalization and practical edge deployment are challenges in existing approaches. This paper presents an efficient wearable fall detection system based on CNN-LSTM that achieves robust performance across multiple benchmark datasets, with real-time inference on resource-constrained microcontroller units (MCU) while maintaining low energy consumption. The proposed architecture combines convolutional layers with long short-term memory cells to capture spatial-temporal patterns in tri-axial accelerometer signals and their RMS magnitude. Using LOSO validation on the KFall dataset, the model achieves a subject-averaged accuracy of 99.3% and an F1-score of 98.97%. For cross-dataset validation, zero-shot transfer to SisFall achieved 98.3% accuracy and 97.9% F1-score without retraining, representing approximately 1.0% performance degradation under controlled laboratory conditions. The trained model is successfully deployed on an ESP32-S3 MCU, achieving an inference latency of 113.6 ms per 4.0-s window with an average power consumption of 5.78 mA, enabling up to 7 days of continuous operation or approximately 3 weeks under typical daily usage cycles. The proposed system offers a highly practical, energy-efficient baseline that paves the way for future real-world elderly monitoring applications by successfully integrating efficient MCU execution with robust cross-dataset transfer on simulated falls.

Keywords: CNN-LSTM; Fall detection; Real-time inference; TinyML; Wearable sensors

1. Introduction

Falls pose a significant public health challenge for the aging global population (Noury et al., 2007). The World Health Organization (WHO) reports that approximately 684,000 individuals die annually from fall-related injuries, with adults aged 60 and older experiencing the highest incidence rates (Wang et al., 2020). In the United States alone, more than one-third of older adults experience at least one fall each year, resulting in over $50 billion in healthcare costs annually (Rubenstein, 2006; Kakara et al., 2023). These incidents often result in debilitating fractures and loss of independence. They also cause psychological trauma that significantly reduces the quality of life of older adults (Haslam-Larmer et al., 2021; Wang et al., 2020; Verma et al., 2016). As global life expectancy continues to rise and the number of individuals over 60 is projected to reach 2.1 billion by 2050 (Naja et al., 2017), the need for effective fall detection systems has become increasingly urgent (Jiang et al., 2024; Xu et al., 2018).

Although camera-based and ambient sensor approaches have been explored for fall detection, they often require fixed infrastructure and raise significant privacy concerns due to continuous video recording in private living spaces (Gorce and Jacquier-Bret, 2025; Lau et al., 2022). Wearable fall detection systems (WFDS) have gained considerable attention due to their ability to continuously monitor individuals in real-world settings (Gorce and Jacquier-Bret, 2025; Naeim et al., 2023; Nizam and Jamil, 2020). These systems are known for their portability

https://doi.org/10.14716/ijtech.v17i3.8445
Received March 2026; Revised March 2026; Accepted April 2026

and universality, being economical and easier to implement than other alternatives, while also posing fewer privacy issues (Jiang et al., 2024). Existing fall detection methods face significant challenges in striking a balance between accuracy and computational efficiency. Although traditional threshold-based methods are lightweight and straightforward, they cannot detect complex motion patterns and distinguish falls from normal activities with similar movements (Aziz et al., 2017). In contrast, advanced deep neural networks achieve high accuracy but require high computational resources, making them unsuitable for limited capacity wearable devices (Pereira et al., 2024). This trade-off between accuracy and efficiency remains a significant limitation in the adoption of intelligent fall detection systems in elderly care applications (Liu et al., 2023).

The advent of TinyML, driven by the availability of open-source inference frameworks such as TensorFlow Lite Micro (David et al., 2021) and advances in model compression techniques (Arif and Rashid, 2025; Kallimani et al., 2023), has created new opportunities for on-device inference in wearable healthcare monitoring systems (Maruf et al., 2025; Benoit et al., 2024). By allowing machine learning models to run directly on MCUs, TinyML addresses several critical challenges in healthcare IoT applications: it enhances user privacy through localized data processing, eliminates cloud communication delays for real-time responsiveness, and reduces power consumption by minimizing continuous wireless transmission (Arif and Rashid, 2025; Yahyati et al., 2025). Many TinyML applications rely heavily on quantization and pruning. Researchers can achieve practical deployment through inherent architectural optimization (Mao et al., 2024). Compact models can operate directly with full precision on MCUs that possess sufficient memory. Despite these structural optimizations, adapting deep neural networks to strict hardware constraints remains a significant challenge. The system must preserve high detection accuracy within these extreme computational boundaries (Benoit et al., 2024).

Despite significant advancements in WFDS research, three critical challenges remain unaddressed. First, many machine learning fall detection models show weak generalization across datasets, with sensitivity and specificity often decreasing below 90%, and in some cases even dropping below 70%, when tested on populations or sensor configurations different from the training data (Fula and Moreno, 2024; Silva et al., 2024). This limitation undermines their reliability in various real-world deployment scenarios. Second, state-of-the-art (SOTA) models with millions of parameters are fundamentally incompatible with MCU deployment, creating a substantial gap between laboratory demonstrations and practical applications. Third, while numerous studies report high accuracy metrics, few offer comprehensive validation for actual edge deployments. This includes considerations such as real-time inference latency, energy consumption profiles, and long-term operational viability of battery-powered wearable devices. However, few existing works have demonstrated both cross-dataset transfer and MCU deployment within the same framework, highlighting a crucial gap between algorithmic evaluation and embedded implementation.

This study addresses the current limitations of fall detection using a lightweight CNN-LSTM architecture. The proposed model achieves high detection accuracy and effectively generalizes across other datasets without retraining. The primary methodological novelty lies in the lightweight design's end-to-end integration with robust cross-dataset reliability. This framework is further validated through a comprehensive performance profile of resource-constrained MCUs, bridging the gap between algorithmic theory and practical edge deployment. The main contributions of this work are summarized as follows:

- A lightweight CNN-LSTM architecture with only 21,434 parameters is proposed. The model achieves 99.3% accuracy and 98.97% F1-score on the KFall dataset using LOSO validation, demonstrating the viability of efficient neural networks for fall detection.

- Cross-dataset evaluation shows that the model generalizes reasonably well to the SisFall dataset, achieving 98.3% accuracy and 97.9% F1-score via zero-shot transfer, with a performance drop of approximately 1.0%.

- We present a complete TinyML deployment of the proposed CNN-LSTM model on the ESP32-S3 MCU, achieving an inference latency of 113.6 ms over a 4-s window and an average power consumption of 5.78 mA, enabling continuous operation for 7 days, or an effective lifespan of 3 weeks, under normal usage conditions.

The remainder of this paper is organized as follows: Section 2 reviews related works on WFDS and TinyML applications. The proposed CNN-LSTM architecture for fall detection is presented in Section 3. Section 4 outlines the experimental setup, including the workflow, datasets and preprocessing, training strategy, cross-dataset evaluation, edge deployment, and performance metrics. Section 5 presents and discusses the experimental results, including model performance, cross-dataset generalization capability, ablation studies, energy consumption analysis, limitations of the current study, and future research directions. Finally, Section 6 concludes the paper with a summary of key contributions and findings.

2. Related Works

Early WFDS relied on simple threshold-based approaches to identify falls by monitoring acceleration magnitudes and comparing them against predefined thresholds. Although computationally lightweight, threshold methods suffer from high false-positive rates and poor generalizability across diverse populations and motion scenarios (Yu et al., 2023). Machine learning methods have subsequently emerged to overcome these limitations. Feature engineering approaches have been developed in combination with classical classifiers, such as support vector machines (SVM) and random forests (RF) (Afuan and Isnanto, 2025; Guo and Nakayama, 2025; Kausar et al., 2024; Poh et al., 2019). Hussain et al., 2019 achieved 99.80% accuracy using KNN with hand-crafted features from accelerometer and gyroscope data (Hussain et al., 2019). Xu et al., 2021 proposed a fusion approach that combines threshold-based methods with convolutional neural network (CNN), achieving 98.04% sensitivity and 96.91% specificity by leveraging the strengths of both paradigms for the preliminary detection and confirmation stages (Xu et al., 2021).

Deep learning approaches automatically extract hierarchical feature representations from raw sensor signals, substantially improving detection performance. Convolutional neural networks (CNNs) capture spatial patterns in acceleration data, while long short-term memory (LSTM) networks model temporal dependencies (Amir et al., 2024; Mohan et al., 2024). Hybrid CNN-LSTM architectures combine these complementary strengths. Hu et al., 2025b demonstrated that integrated CNN-LSTM models achieve 96.94% detection accuracy with 98.33% sensitivity on wrist-worn sensors (Hu et al., 2025b). Recent developments incorporate attention mechanisms to enhance feature prioritization. Sahni et al., 2025 proposed an ensemble model that combines a CNN with channel attention and a BiLSTM with temporal attention, achieving 98.99% accuracy (Sahni et al., 2025). Li et al., 2020 demonstrated that multimodal fusion using BiLSTM networks improved cross-participant robustness by reducing accuracy variance by 18.1% and increasing worst-case accuracy by 16.2% (Li et al., 2020).

A critical emerging trend is embedding fall detection directly on wearable devices for real-time operation with extended battery life (Amir et al., 2024; Sehairi and Chouireb, 2023).Yu et al., 2023 proposed TinyCNN, a two-stage efficient feature extraction model, achieving 99% sensitivity and specificity on the KFall dataset with only 37 ms latency on Arduino using quantization techniques (Yu et al., 2023). Hu et al., 2025a introduced MicroFallNet, a model that utilizes the FireModel architecture with Squeeze and Expand layers, achieving a 97.91% geometric mean accuracy while reducing the model size to 8.46 KB and inference time to 30.3 ms on ESP32 smart wristbands. Koo et al., 2023 developed TinyFallNet based on ResNet, achieving 98.00% accuracy with only 0.70 MB memory requirement, demonstrating that image-classification architectures can be effectively adapted for temporal sensor signals (Koo et al., 2023). Abdou et al., 2025 demonstrated that SVM-based approaches achieve 99.7% accuracy on public datasets while consuming only 0.85 KB RAM and 10.4 mAh power, enabling practical

continuous monitoring scenarios (Abdou et al., 2025).

Recent advancements in wearable fall detection have demonstrated notable detection performance. However, these methods still have some practical limitations. Highly accurate models often require computational resources beyond the edge device capacity. Existing TinyML solutions mostly rely on single-dataset evaluations without verifying their generalization across different datasets. Moreover, the lack of detailed energy consumption analysis limits the understanding of their real-world battery life. This study proposes a comprehensive approach that combines high accuracy, cross-dataset generalization, and energy-efficient execution on embedded platforms.

3. Proposed CNN-LSTM Model

The architecture of the proposed model is designed to accurately detect falls while meeting the computational needs of edge devices in wearable settings. As shown in Figure 1, the model effectively processes spatial and temporal information in accelerometer signals by combining convolutional and RNN components.

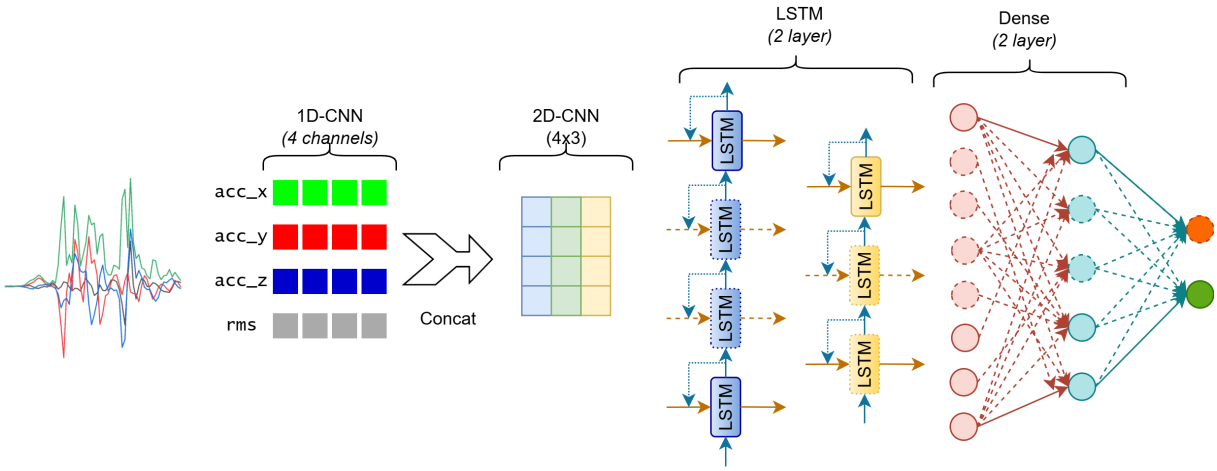

Figure 1 Overview of the CNN-LSTM model architecture

The input comprises raw triaxial accelerometer data and RMS values. The data are first passed through four parallel one-dimensional convolutional neural network (1D-CNN) layers, each of which is responsible for extracting features from a single signal channel. Then, the outputs of these layers are combined to form a single feature tensor. This tensor is then used as input for a two-dimensional convolutional (2D-CNN) layer that learns new spatiotemporal patterns across all channels and time frames.

Table 1 Configuration of the model architecture

Layer	Output Shape	Filter / Units	Kernel size	Param #
Input	(200, 4)	-	-	0
Conv2D (*Channels 1–4*)	(24, 1, 16)	16	(8,1)	544×4
Concatenate	(24, 4, 16)	-	-	0
Conv2D	(21, 2, 32)	32	(4,3)	6,176
LSTM	(42, 32)	32	-	8,320
LSTM	(16)	16	-	3,136
Dense	(64)	64	-	1,088
Dense	(8)	8	-	520
Dense (*Output*)	(2)	2	-	18
Total Parameters:				**21,434**

After feature extraction, the architecture comprises two LSTM layers connected in series. These layers accurately represent the temporal dependencies and dynamic behaviors characteristic of fall events and routine activities. The LSTM blocks learn the temporal features, which are then passed to a stack of dense layers that perform the final classification and assign a probability score to each activity class. Table 1 presents the detailed configuration of each model layer, including output shape and parameter count. The total number of parameters is 21,434, ensuring that the model remains efficient in terms of computational cost and memory usage, which is important for real-time MCU deployment.

4. Experimental Setup

4.1 Experimental Workflow

The experimental workflow was structured to validate the proposed CNN-LSTM model for fall detection using the KFall (Yu et al., 2021) and SisFall (Sucerquia et al., 2017) datasets. As shown in Figure 2, the data from each dataset were first normalized using z-score standardization to ensure consistency and comparability between subjects. A LOSO cross-validation strategy was deployed for the KFall dataset. This approach separated the data into training and testing sets by holding out one subject for evaluation while the remaining data were used for training. This iterative process was repeated across all 32 participants to ensure that each individual was evaluated in the test set exactly once.

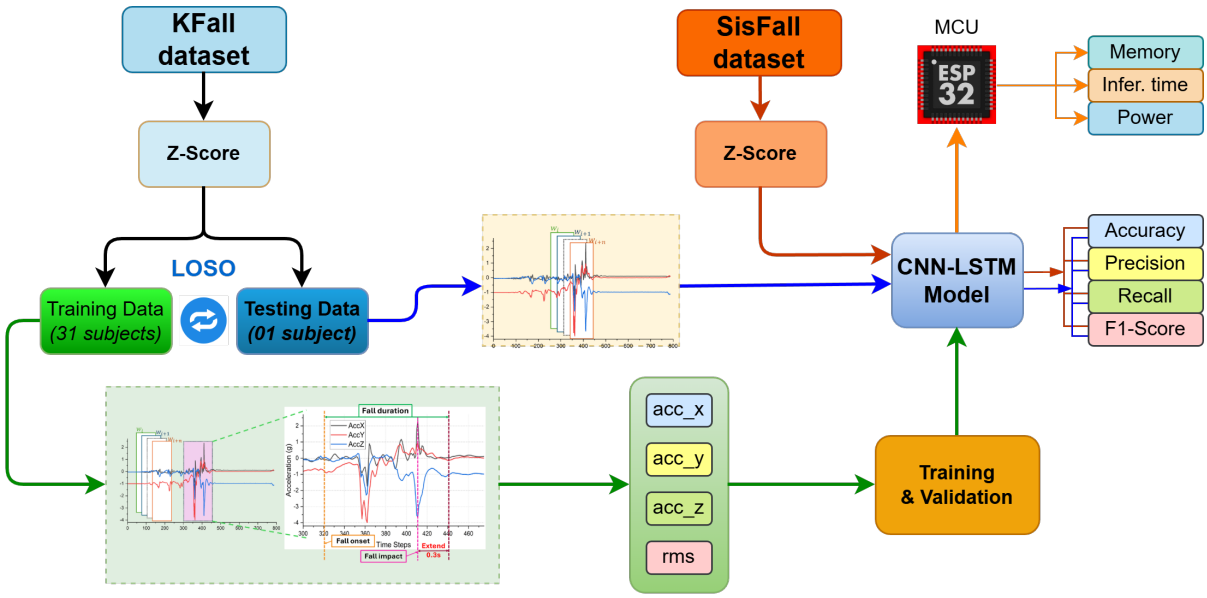

Figure 2 End-to-end fall detection pipeline with the CNN-LSTM model and embedded implementation

Preprocessed accelerometer signals were segmented via a sliding window method, which extracted relevant temporal sequences around falls and everyday activities. The signals from the x, y, z, and RMS values were then organized as the model's input channels.

The prepared data were used for model training and validation. The CNN-LSTM model was evaluated on the held-out subject or transferred to the SisFall dataset for cross-dataset validation after training. The performance of the model was measured using accuracy, precision, recall, and F1-score metrics. The trained model was also deployed on the MCU, where the inference time and memory usage were recorded to evaluate the model's practical feasibility for edge deployment.

This study also measured the system's energy consumption during continuous inference on the MCU. The power consumption tests helped determine the operating time of the device without recharging, which was especially important for wearable applications. These measure-

ments supported the assessment of the device's ability to provide reliable, long-term monitoring in daily settings. This workflow ensured that the proposed approach was thoroughly tested for both generalization across subjects and datasets and its suitability for real-time, energy-efficient WFDS.

4.2 Datasets and Preprocessing

This study used two publicly available fall detection datasets, namely, KFall and SisFall, which offered comprehensive coverage for evaluating human activity recognition models. The KFall dataset included sensor recordings from 32 young participants who performed 21 types of activities of daily living and 15 types of simulated falls, yielding a total of 5,075 trials collected at a frequency of 100 Hz (Yu et al., 2021). In contrast, the SisFall dataset contained data from 23 young and 14 elderly subjects, with 19 normal activities and Fifteen simulated falls with 4,505 trials recorded at 200 Hz (Sucerquia et al., 2017). Table 2 presents a summary of dataset characteristics.

Table 2 Summary of the fall detection datasets used in this study

Dataset	Subjects	ADL / Fall	Trials (A/F)	Sample rate
KFall	32 young	21/15	5,075 *(2,729/2,346)*	100 Hz
SisFall	23 young 14 elder	19/15	4,505 *(2,707/1,798)*	200 Hz

The KFall and SisFall datasets underwent independent preprocessing. For each instance, the tri-axial accelerometer values (a_x, a_y, a_z) were retained, and an additional root mean square (RMS) channel was computed as in Equation (1).

$$RMS = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

Afterward, z-score normalization was applied independently on each dataset over all channels, including the RMS channel. The z-score transform is defined in Equation (2) as follows:

$$x = \frac{x - \mu}{\sigma} \tag{2}$$

where μ and σ denote the mean and standard deviation of each channel, respectively. In addition, both datasets were downsampled to 50 Hz to reduce computational demands. The KFall dataset, with precise labels indicating fall onset and impact times, was used for training, whereas the SisFall dataset was reserved for cross-dataset validation. In the KFall dataset, the duration of fall was defined as the time from onset to impact, extended by 0.5 s. Sliding windows of 4 s (200 samples) with 2 s (100 samples) strides were created to segment the data. Each window was labeled as a fall if at least 50% of the fall duration overlapped it; otherwise, it was assigned as an activity of daily living (ADL).

4.3 Training Strategy

After z-score normalization and segmentation, the preprocessed data from the KFall dataset were used for training using the LOSO cross-validation method. Data from 31 subjects were used for training in each iteration, while the remaining subjects were reserved for testing. This process was repeated 32 times until every subject served as the test subject exactly once. This strategy ensures that the reported metrics reflect person-independent generalization across all 32 participants, rather than performance on any single individual.

The model was trained for 25 epochs for each iteration using the Adam optimizer with a learning rate of 0.001. The weighted categorical cross-entropy loss function was adopted with class weights of 0.8 and 0.2 for the ADL and Fall classes, respectively, to address class

imbalance. Although a higher weight on the Fall class might be expected for a safety-critical system, preliminary experiments showed that this configuration led to a significant increase in false positives, as ADL activities with rapid acceleration profiles were frequently misclassified as falls. Therefore, the selected weights were determined through empirical validation across LOSO folds to achieve the optimal balance between recall and precision.

The TensorFlow framework was used to implement all training and evaluation procedures on a server equipped with two Intel Xeon E7-8880 v4 CPUs, 64 GB of RAM, and two NVIDIA Tesla P100 GPUs with 16 GB of memory each.

4.4 Cross-Dataset Evaluation

After training the CNN-LSTM model on the KFall dataset, the model was evaluated on the SisFall dataset without fine-tuning, demonstrating zero-shot transfer. This evaluation assessed the generalizability and adaptability of the model to new data with varying characteristics. This evaluation approach differs from the window-level assessment used in the KFall LOSO validation because of a fundamental difference in label granularity between the two datasets. The KFall dataset provides precise frame-level timestamps for the onset and impact of falls, enabling direct window-level ground-truth assignment, as detailed in Section 4.2. In contrast, the SisFall dataset provides only trial-level binary labels without any sub-event timing information, rendering window-level evaluation impractical. Consequently, a sample-level aggregation rule was adopted: if at least two consecutive sliding windows are predicted as falls, a trial is classified as a fall; otherwise, it is categorized as an ADL. This criterion is grounded in the sliding window design, which ensures overlap with at least 2 consecutive 4-second windows with a 2-s stride. This rule also matches the detection logic required for WFDS. Such an implementation necessitates sustained predictions during consecutive inference cycles before an alert is triggered.

4.5 Edge Deployment

After the training process, the optimized CNN-LSTM model was deployed on a Seeed Studio XIAO ESP32-S3 MCU to evaluate its real-time feasibility on an embedded platform. The MCU operated at a clock frequency of 240 MHz and included 512 KB of internal SRAM, 8 MB of external PSRAM, and 4 MB of flash memory. The trained model was converted to TensorFlow Lite (TFLite) format, and the TFLite-Micro framework was used to execute the deployment (David et al., 2021). The model was converted to the float32 TFLite format without post-training quantization because the 8 MB PSRAM available on the ESP32-S3 provides sufficient memory for full-precision inference. This choice preserves the numerical fidelity of the original Keras model, eliminating any accuracy degradation associated with integer quantization. In this context, the TinyML contribution of this work lies in the design of a compact, hardware-aware architecture that fits MCU memory constraints from the ground up, the deployment of a complete inference pipeline on a resource-constrained device, and the adoption of a light-sleep power strategy.

The ESP-IDF framework was used to implement the firmware, with additional optimization through the ESP-NN library to enhance the efficiency of neural network operations on the ESP32-S3. The system employed a low-power strategy in which the MCU remained in light sleep mode to retain the model parameters in memory and woke up every 2 s to perform inference on incoming sensor data.

4.6 Evaluation Metrics

To evaluate the performance of the model, we calculated several metrics with fall as the positive class. These metrics were based on the counts of TP, TN, FP, and FN. TP represents the number of correctly detected falls. TN is the number of correctly identified non-falls. FP occurred when non-falls were mistakenly classified as falls, and FN occurred when actual falls were missed. The accuracy (Equation 3) measured the proportion of correctly classified samples

among all samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Recall (Equation 4) quantified the proportion of accurately identified actual falls, also known as sensitivity:

$$Recall = \frac{TP}{TP + TN} \tag{4}$$

Precision (Equation 5) represents the proportion of predicted falls that were true falls:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

The F1-score, defined in Equation (6), is computed as the harmonic mean of precision and recall, which balances detection sensitivity and false alarm rates. Because both factors were essential for WFDS and fall events were rare in practice, this metric provided a more reliable performance evaluation than accuracy.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

Since we employ the LOSO strategy, we report the performance metrics using two aggregation methods:

1. Subject-wise average (Macro-average): The accuracy, precision, recall, and F1-score metrics are calculated independently for each subject, and the mean values across all 32 subjects are reported. This reflects the average performance of the model on an individual user.

2. Global aggregate (micro-average): All predictions from all sliding windows across all 32 subjects are pooled together into a single global confusion matrix to compute the overall metrics. This reflects the performance of the system over the entire dataset.

In addition to the static performance metrics, the system's real-time responsiveness and long-term operational capacity were evaluated. RAM and flash memory usage, as well as the inference time per data window, were measured during deployment to ensure timely responses. Furthermore, the energy consumption was quantified using the Power Profiler Kit 2, and the average current (I_{avg}) during inference was recorded. The expected battery life T_{life} in hours was then estimated using Equation (7), by dividing the standard battery capacity C (assumed to be 1000 mAh for typical wearable devices) (Zhou et al., 2025) by I_{avg} :

$$T_{life} = \frac{C}{I_{avg}} \tag{7}$$

5. Results and Discussion

5.1 Performance on the KFall Dataset

The proposed model achieved outstanding results on the KFall dataset. The model attains a mean accuracy of 99.3% and a mean recall of 99.36% when evaluated on a per-subject basis (subject-wise average), underscoring its ability to minimize missed fall events. Precision reaches 98.60%, and F1 score is 98.97%. These metrics demonstrate the robustness of the model in managing the imbalanced distribution between falls and ADLs.

Figure 3 presents a box plot illustrating the consistency of performance across participants. Notably, 18 of the 32 individuals obtained F1-scores of 99% or higher, indicating a strong and stable detection capability. Only one participant, P26, emerged as a statistical outlier in the

LOSO validation results. This individual achieved an accuracy of 97.4% and an F1-score of 96.35%, which deviated from the group mean due to personal differences in fall kinematics and sensor response patterns. Despite this deviation, the 96.35% F1 score remains clinically acceptable for real-world wearable deployment. All test windows across the 32 subjects were pooled to provide a holistic view of the window-level predictions to construct a global aggregate confusion matrix, as shown in Figure 4(a). From this global perspective, 14,907 out of 15,131 actual fall windows, 14,907 were correctly identified while 224 were missed, yielding a global aggregate recall of 98.51% and a global precision of 99.34%. Furthermore, only 100 ADL windows were misclassified as falls. The slight numerical variation between the subject-averaged and global aggregate metrics is statistically expected due to the varying number of windows generated by different subjects and activity durations. These results confirm the model's reliability, with a low occurrence of false negatives, which is critical for fall detection safety, and a limited number of false positives, indicating its effectiveness in discriminating complex activities.

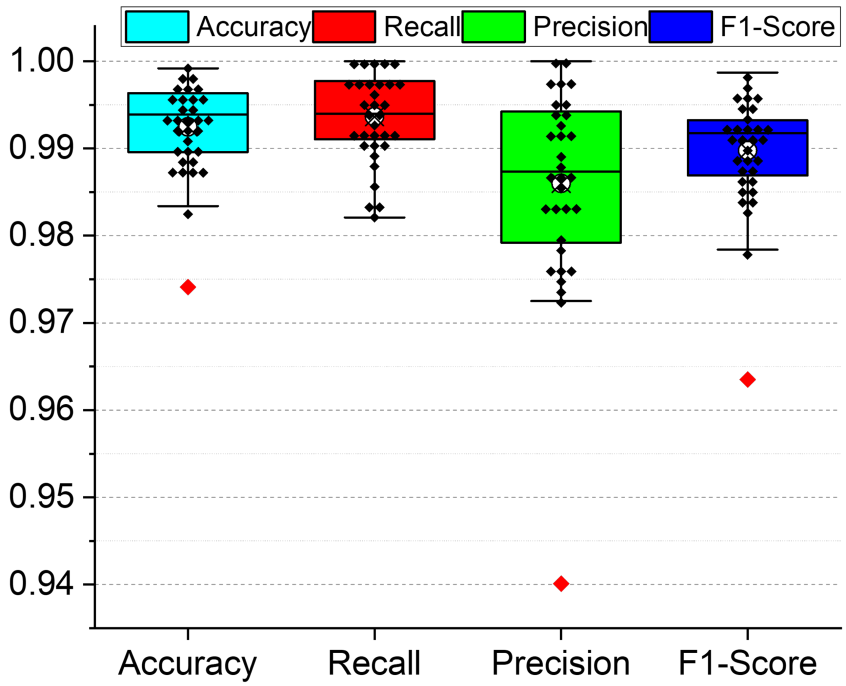

Figure 3 LOSO cross-validation performance distribution on the KFall dataset

5.2 Cross-Dataset Generalization

The trained CNN-LSTM model on the KFall dataset was applied directly to the SisFall dataset without fine-tuning to evaluate the model's generalization capability. The results indicate a minor performance reduction compared with intra-dataset validation, confirming the proposed architecture's robustness. The model achieves an accuracy of 98.3%, recall of 97.6%, precision of 98.2%, and F1-score of 97.9%. This represents a degradation of approximately 1.0% in accuracy and F1-score, demonstrating strong cross-dataset transferability despite differences in sensor configurations and activity patterns between datasets.

The confusion matrix in Figure 4(b) reveals that most predictions remain highly accurate across both classes. However, a small number of ADL samples were incorrectly classified as falls, primarily corresponding to activities with rapid or complex motion patterns, such as D10 (quickly sitting in a low chair and standing up again) and D17 (entering and exiting a car). These activities exhibit brief acceleration periods, similar to real falls, which explains their occasional misclassification. Conversely, several fall activities, notably those involving slower or more controlled movements such as F11 (falling backward when trying to sit down) and F15 (lateral fall while sitting), were misidentified as ADL. These patterns indicate that low-intensity

or gradually occurring falls tend to have smaller acceleration peaks, making them more difficult to distinguish from everyday activities.

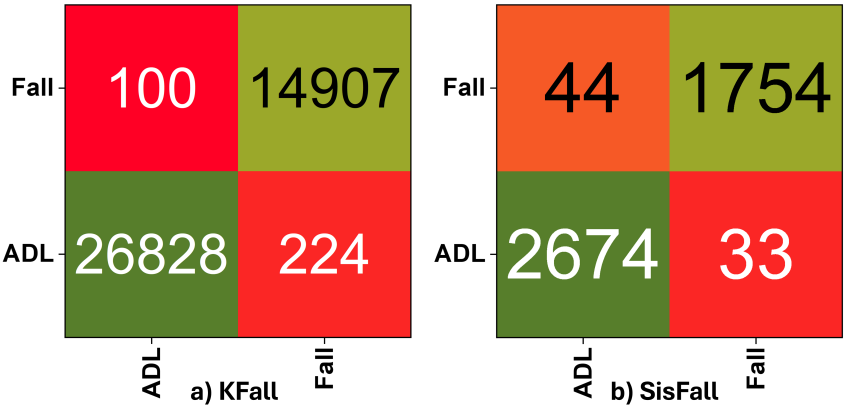

Figure 4 Confusion matrices: (a) Global aggregate matrix across all test windows in KFall LOSO validation and (b) SisFall zero-shot transfer evaluation

5.3 Ablation Studies

An ablation study was conducted using segment lengths ranging from 2.0 to 6.0 seconds to determine the optimal input window size. The analysis focused on evaluating the trade-off between the temporal resolution and the ability of the model to capture complex motion patterns over longer durations. The results summarized in Figure 5 demonstrate that as the window size increases, the model performance on the KFall dataset consistently improves across all four evaluation metrics. Notably, the F1-score rises from 96.0% at 2.0 seconds to 98.97% at 4.0 seconds and continues to approach saturation near 6.0 seconds. This trend shows that extended observation periods improve the model's ability to learn detailed temporal dependencies and recognize fall dynamics more effectively.

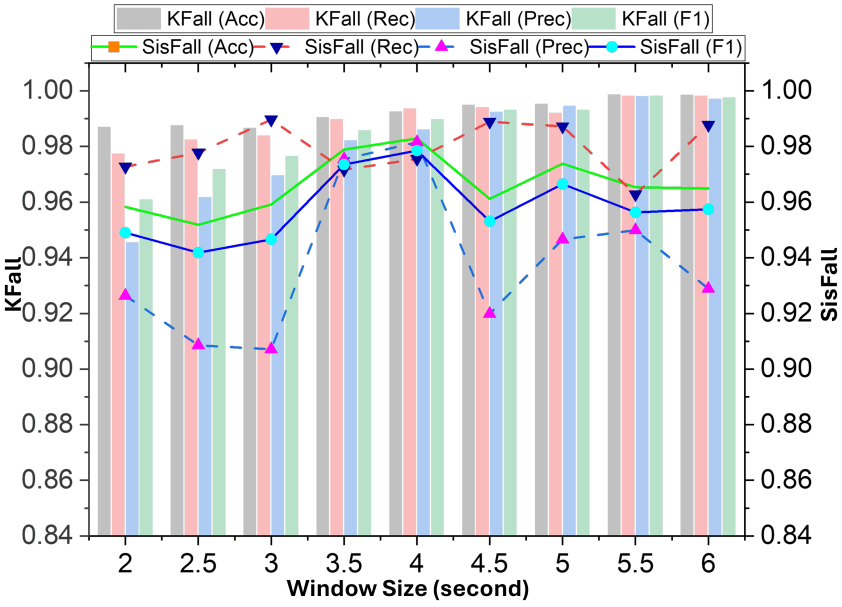

Figure 5 Window size optimization: performance metrics on KFall (bars) and SisFall (lines)

However, when the same model was evaluated under the cross-dataset condition with SisFall, apparent performance degradation was observed for window sizes beyond 4.0 s. Although the overall accuracy and recall remained relatively stable, the precision metric showed a marked decline, dropping from 98.2% at 4.0 s to approximately 92.8% at 6.0 s. This indicates that an

excessively long input window introduces redundant information, increasing the likelihood of misclassifying non-fall activities as falls.

Based on this analysis, a 4.0-s window was selected as the optimal configuration. It provides the best balance between intra-dataset accuracy and cross-dataset generalization, delivering strong temporal representation without unnecessary computational overhead.

5.4 Edge deployment performance

The evaluation results, summarized in Table 3, illustrate the efficiency of the deployed CNN-LSTM model on the ESP32-S3. The CNN-LSTM model occupies 96.9 KB of memory after conversion. The 431.6 KB flash usage confirms that both the executable code and model weights were efficiently integrated without exceeding the available capacity. Similarly, the 199.5 KB RAM consumption ensures safe multitasking operation and efficient handling of temporary buffers during execution. The inference latency of 113.6 ms per data window demonstrates that the proposed system achieves real-time performance suitable for time-sensitive applications.

Table 3 Resource usage and inference performance of the ESP32-S3

Parameter	Value [bytes]	Approx. [KB]
Model size	99,176	96.9
Flash	441,868	431.6
RAM	204,262	199.5
Inference time	113.6 ms	

In addition to memory utilization, energy consumption was examined to assess the edge deployment's overall power efficiency. As illustrated in Figure 6, the detailed measurement results highlight the current variation between the active inference and light sleep modes. During active inference, the ESP32-S3 MCU consumed an average current of 86.5 mA over a duration of 113.6 ms. When idle, the current decreased significantly to 0.927 mA in the light sleep mode. The average current was measured at 5.78 mA over a one-minute observation interval.

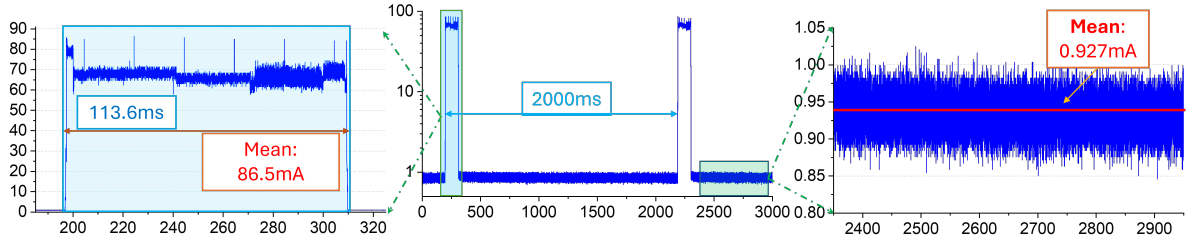

Figure 6 ESP32-S3 power characterization for edge deployment: (Left) single inference, (Center) 3 s intervals, and (Right) idle

Based on these measurements, the device can operate continuously for approximately 170 hours with a nominal 1,000 mAh battery capacity, equivalent to nearly 7 days of uninterrupted function. Under a typical usage pattern of 8 operational hours per day, the system's endurance extends to approximately 21 days. These results verify that the proposed deployment achieves real-time inference capability while maintaining low energy demands. This performance reinforces the feasibility of long-term battery-powered operation for wearable sensor nodes.

5.5 Comparison with SOTA

Table 4 provides an indicative comparison between the proposed CNN-LSTM model and other recent SOTA TinyML fall detection approaches. These studies vary significantly in terms of hardware platforms, model architectures, and evaluation protocols. Consequently, the following benchmarking should be considered as a general performance guide rather than a strictly

equivalent comparison. The proposed method achieves 99.3% accuracy on the KFall dataset, which favorably or marginally surpasses the performance reported in the existing literature. While MicroFallNet and Abdou et al. report much smaller model sizes, they achieve this compactness at the cost of reduced accuracy. For instance, Abdou et al. achieved a model size of 0.85 KB but a lower accuracy of 96.40%. This demonstrates that the modest increase in memory footprint to 96.9 KB in our model provides a significant gain in detection reliability. Such a trade-off is critical for safety-sensitive applications, such as elderly care, where high precision and recall are essential.

Regarding inference latency, the proposed system achieves real-time performance with 113.6 ms on the ESP32-S3. Although TinyCNN reported a lower latency of 37 ms, that result was obtained using int8 quantization on an Arduino platform. In contrast, our results reflect full-precision float32 inference to eliminate any accuracy degradation associated with integer quantization. Furthermore, our latency is significantly lower than that of TinyFallNet at 411 ms, which was evaluated on a PC platform rather than an actual MCU. A major distinction of this work is the explicit validation of cross-dataset generalization, a feature often missing in existing TinyML solutions. Zero-shot transfer to the SisFall dataset achieves 98.3% accuracy with only a 1.0% performance drop. The system demonstrates an average power consumption of 5.78 mA, enabling continuous operation for up to 7 days. The results collectively confirm that the proposed framework offers a robust and energy-efficient balance for practical edge deployment while the benchmarked metrics remain indicative due to experimental variations.

Table 4 Comparison of state-of-the-art fall detection methods

Method	Accuracy	Model size	Latency (ms)	Platform	Cross-Dataset
Abdou et al., 2025	96.40%	0.85 KB	131.8	MCU	×
MicroFallNet (Hu et al., 2025a)	97.91%	8.46 KB	30.3	ESP32	×
TinyFallNet (Koo et al., 2023)	98.00%	0.70 MB	411	PC (Non-MCU)*	×
TinyCNN (Yu et al., 2023)	99.00%	–	37 (*Quantized*) 126 (*Float*)	Arduino	×
CNN-LSTM (*This study*)	99.3% (KFall) 98.3% (SisFall)	96.9 KB	113.6	ESP32	✓

**TinyFallNet is implemented and evaluated on a PC platform; its accuracy is reported here for reference only, without direct comparison of latency or resource usage.*

5.6 Discussion

The experimental results demonstrate significant advancements in WFDS. The zero-shot cross-dataset evaluation achieved an impressive 98.3% accuracy on SisFall, having been trained solely on KFall, and exhibited only a 1.0% degradation without the need for retraining. This outcome is particularly noteworthy given the variations in sensor specifications, sampling rates, demographic factors, activity protocols, and fall simulations in the two datasets. The minimal performance loss confirms that the CNN-LSTM architecture effectively captures the intrinsic fall characteristics rather than relying on the dataset-specific features.

The architectural design incorporates convolutional layers with LSTM cells to effectively capture the multidimensional characteristics of fall events. The convolutional layers analyze triaxial accelerometer data to identify distinct patterns, such as sudden acceleration and impact features, thereby capturing the immediate physical dynamics associated with falls. Subsequently, the LSTM cells process these spatial features to model temporal dependencies, encoding the trajectory of falls from instability to immobility. This capability enables the system to distinguish falls from similar activities, such as quickly sitting down or jumping, which may present comparable acceleration patterns but differ in timing. The CNN-LSTM model effectively captures the complete fall signature by integrating both spatial and temporal dimensions, resulting in robust classification performance.

The evaluation of the MCU demonstrates the system's practical viability for wearable de-

vice development. An inference latency of 113.6 ms for a 4.0-s window meets the strict timing requirements. The average current consumption of 5.78 mA (19 mW) enables continuous operation for 7 days on a standard battery. This energy efficiency is crucial for elderly monitoring applications, where frequent battery replacements reduce user compliance and system reliability. The successful deployment on the resource-constrained ESP32-S3 MCU demonstrates that sophisticated deep learning models can operate effectively without requiring expensive hardware or cloud connectivity, thereby reducing system cost and eliminating privacy concerns associated with transmitting sensitive health data.

The balanced performance between precision and recall, reflected in F1-scores exceeding 98%, is essential for maintaining user trust. High precision minimizes false alarms that cause alert fatigue among caregivers and may lead to genuine alerts being ignored. Simultaneously, high recall ensures that actual fall events trigger timely assistance. The LOSO validation methodology strengthens confidence in these results by ensuring person-independent generalization, which is crucial for deployment scenarios where individual-specific calibration is impractical. The consistent performance across different subjects indicates that the model has learned robust fall patterns applicable to diverse users without requiring personalized training data.

5.7 Limitations and Future Work

Despite the promising results, some limitations must be acknowledged. The primary limitation concerns the datasets used for training and evaluation. Both KFall and SisFall consist predominantly of simulated falls performed by healthy young volunteers in controlled laboratory settings rather than genuine falls involving elderly individuals. Research demonstrates that real-world falls exhibit distinct characteristics, particularly in terms of descent velocity, impact force, and post-fall movements, due to reduced muscle strength in elderly populations. Furthermore, the evaluation does not account for real-world deployment challenges, including sensor placement variations due to daily clothing changes, interference from loose garments, and diverse environmental conditions, such as different floor surfaces. These factors may affect accelerometer readings and influence detection accuracy. Additionally, the study focuses exclusively on single-location accelerometer data, which may not capture the full dynamics of falls compared with multi-sensor approaches that incorporate data from gyroscope and magnetometer sensors.

Future research should address these limitations in several ways. First, long-term field trials involving elderly participants in residential environments would provide valuable insights into real-world performance while also considering the ethical and safety implications. Second, multimodal sensor fusion, which combines inertial measurement data with complementary modalities, such as ambient sensors, could reduce false alarms while maintaining sensitivity. Third, adaptive learning mechanisms that personalize on the basis of individual patterns would enhance accuracy and user acceptance by leveraging online learning techniques or efficient transfer learning with minimal computational overhead. Finally, extending the framework to support fall risk assessment and prevention through gait analysis and activity monitoring would provide value beyond reactive detection. Such predictive capabilities enable early intervention strategies to reduce the incidence of falls, thereby complementing the immediate alert functionality.

6. Conclusions

This study presents an energy-efficient wearable fall detection system based on the CNN-LSTM architecture, which successfully bridges the gap between laboratory accuracy and practical edge deployment for elderly care. The proposed approach combines CNN layers with LSTM cells to capture spatial-temporal patterns in triaxial accelerometer data, enabling a robust distinction between fall events and ADL. The experimental evaluation demonstrated exceptional performance, achieving 99.3% accuracy and 98.97% F1-score using LOSO validation on the KFall dataset. Moreover, the transfer across datasets to SisFall achieved 98.3% accuracy and 97.9%

F1 score without retraining. This result represents a performance degradation of 1.0%. This capability confirms that the architecture learns fundamental fall patterns rather than artifacts specific to the dataset. Furthermore, the successful deployment on an ESP32 S3 microcontroller achieved an inference latency of 113.6 ms per window of 4.0 s. The system has an average power consumption of 5.78 mA, enabling continuous monitoring for up to 7 days. The proposed architecture combines high accuracy, robust generalization, and energy efficiency to offer a practical hardware solution for wearable monitors. Note that both KFall and SisFall contain simulated falls recorded under controlled settings. Therefore, future clinical validation using natural data from older adults is necessary before this technology can actively prevent falls-related injuries.

Acknowledgements

Luong-Cong Duan was funded by the PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), VinUniversity, code VINIF.2025.TS71.

The Posts and Telecommunications Institute of Technology (PTIT) supported this work. We would also like to thank the EDA Laboratory for providing the necessary experimental resources for this study.

Author Contributions

L.C. Duan: Methodology, software, implementation, and original draft. N.N. Minh: Conceptualization, funding, resources, and code review T.C. Dung: Original draft and code review. T. T. T. Linh: Revision and finalization.

Conflict of Interest

The authors declare no conflicts of interest.

Supplementary Materials

The datasets analyzed in this study are publicly available. The KFall dataset can be accessed at https://sites.google.com/view/kfalldataset, and the SisFall dataset is available at http://sistemic.udea.edu.co/investigacion/proyectos/english-falls/

References

Abdou, A., Mascret, Q., Gurve, D., Gosselin, B., & Krishnan, S. (2025). Embedded tinyml approach for fall detection in geriatric care. https://doi.org/10.36227/techrxiv.174918043.39615584/v1

Afuan, L., & Isnanto, R. (2025). A comparative study of machine learning algorithms for fall detection in technology-based healthcare system: Analyzing svm, knn, decision tree, random forest, lstm, and cnn. *E3S Web of Conferences*, *605*, 03051. https://doi.org/10.1051/e3sconf/202560503051

Amir, N., Dziyauddin, R., Mohamed, N., Ismail, N., Kaidi, H., Ahmad, N., & Izhar, M. (2024). Fall detection system using wearable sensor devices and machine learning: A review. https://doi.org/10.36227/techrxiv.171084921.16728034/v1

Arif, M., & Rashid, M. (2025). A literature review on model conversion, inference, and learning strategies in edgeml with tinyml deployment. *Computers, Materials & Continua*, *83*(1), 13–64. https://doi.org/10.32604/cmc.2025.062819

Aziz, O., Musngi, M., Park, E., Mori, G., & Robinovitch, S. (2017). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted triaxial accelerometer signals from a comprehensive set of falls and non-fall trials. *Medical & Biological Engineering & Computing*, *55*(1), 45–55. https://doi.org/10.1007/s11517-016-1504-y

Benoit, A., Escriba, C., Gauchard, D., Esteve, A., & Rossi, C. (2024). Analyzing and comparing deep learning models on an arm 32 bits microcontroller for pre-impact fall detection. *IEEE Sensors Journal*, *24*(7), 11829–11842. https://doi.org/10.1109/JSEN.2024.3364249

David, R., Duke, J., Jain, A., Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., & Warden, P. (2021). Tensorflow lite micro: Embedded machine learning on tinyml systems. http://arxiv.org/abs/2010.08678

Fula, V., & Moreno, P. (2024). Wrist-based fall detection: Towards generalization across datasets. *Sensors*, *24*(5), 1679. https://doi.org/10.3390/s24051679

Gorce, P., & Jacquier-Bret, J. (2025). Fall detection in elderly people: A systematic review of ambient assisted living and smart home-related technology performance. *Sensors*, *25*(21), 6540. https://doi.org/10.3390/s25216540

Guo, P., & Nakayama, M. (2025). A feature engineering method for smartphone-based fall detection. *Sensors*, *25*(20), 6500. https://doi.org/10.3390/s25206500

Haslam-Larmer, L., Donnelly, C., Auais, M., Woo, K., & DePaul, V. (2021). Early mobility after fragility hip fracture: A mixed methods embedded case study. *BMC Geriatrics*, *21*(1), 181. https://doi.org/10.1186/s12877-021-02083-3

Hu, J., Cheng, F., Liu, M., Xu, X., & Li, X. (2025a). Microfallnet: A lightweight model for real-time fall detection on smart wristbands. *Pervasive and Mobile Computing*, 102046. https://doi.org/10.1016/j.pmcj.2025.102046

Hu, X., Yu, S., Zheng, J., Fang, Z., Zhao, Z., & Qu, X. (2025b). A hybrid cnn-lstm model for involuntary fall detection using wrist-worn sensors. *Advanced Engineering Informatics*, *65*, 103178. https://doi.org/10.1016/j.aei.2025.103178

Hussain, F., Hussain, F., Ehatisham-ul-Haq, M., & Azam, M. (2019). Activity-aware fall detection and recognition based on wearable sensors. *IEEE Sensors Journal*, *19*(12), 4528–4536. https://doi.org/10.1109/JSEN.2019.2898891

Jiang, Z., Al-Qaness, M., Al-Alimi, D., Ewees, A., Abd Elaziz, M., Dahou, A., & Helmi, A. (2024). Fall detection systems for internet of medical things based on wearable sensors: A review. *IEEE Internet of Things Journal*, *11*(21), 34797–34810. https://doi.org/10.1109/JIOT.2024.3421336

Kakara, R., Bergen, G., Burns, E., & Stevens, M. (2023). Nonfatal and fatal falls among adults aged 65 years — united states, 2020–2021. *MMWR. Morbidity and Mortality Weekly Report*, *72*(35), 938–943. https://doi.org/10.15585/mmwr.mm7235a1

Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., & López, O. (2023). Tinyml: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications*, *83*(10), 29015–29045. https://doi.org/10.1007/s11042-023-16740-9

Kausar, F., Mesbah, M., Iqbal, W., Ahmad, A., & Sayyed, I. (2024). Fall detection in the elderly using different machine learning algorithms with optimal window size. *Mobile Networks and Applications*, *29*(2), 413–423. https://doi.org/10.1007/s11036-023-02215-6

Koo, B., Yu, X., Lee, S., Yang, S., Kim, D., Xiong, S., & Kim, Y. (2023). Tinyfallnet: A lightweight pre-impact fall detection model. *Sensors*, *23*(20), 8459. https://doi.org/10.3390/s23208459

Lau, X., Connie, T., Goh, M., & Lau, S. (2022). Fall detection and motion analysis using visual approaches. *International Journal of Technology*, *13*(6), 1173. https://doi.org/10.14716/ijtech.v13i6.5840

Li, H., Shrestha, A., Heidari, H., Le Kernec, J., & Fioranelli, F. (2020). Bi-lstm network for multimodal continuous human activity recognition and fall detection. *IEEE Sensors Journal*, *20*(3), 1191–1201. https://doi.org/10.1109/JSEN.2019.2946095

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

Mao, W.-L., Wang, C.-C., Chou, P.-H., Liu, K.-C., & Tsao, Y. (2024). Meckd: Deep learning-based fall detection in multilayer mobile edge computing with knowledge distillation. *IEEE Sensors Journal*, *24*(24), 42195–42209. https://doi.org/10.1109/JSEN.2024.3456577

Maruf, M., Haque, M., Hasan, M., Farhan, M., & Islam, A. (2025). State-of-the-art review on fall prediction among older adults: Exploring edge devices as a promising approach for the future. *Measurement: Sensors*, *39*, 101878. https://doi.org/10.1016/j.measen.2025.101878

Mohan, D., Al-Hamid, D., Chong, P., Sudheera, K., Gutierrez, J., Chan, H., & Li, H. (2024). Artificial intelligence and iot in elderly fall prevention: A review. *IEEE Sensors Journal*, *24*(4), 4181–4198. https://doi.org/10.1109/JSEN.2023.3344605

Naeim, M., Chung, G., Lee, I., Tiang, J., & Tan, S. (2023). A mobile iot-based elderly monitoring system for senior safety. *International Journal of Technology*, *14*(6), 1185. https://doi.org/10.14716/ijtech.v14i6.6634

Naja, S., Din Makhlouf, M., & Chehab, M. (2017). An ageing world of the 21st century: A literature review. *International Journal Of Community Medicine And Public Health*, *4*(12), 4363. https://doi.org/10.18203/2394-6040.ijcmph20175306

Nizam, Y., & Jamil, M. (2020). Classification of daily life activities for human fall detection: A systematic review of the techniques and approaches. In *Challenges and trends in multimodal fall detection for healthcare* (pp. 137–179). Springer International Publishing. https://doi.org/10.1007/978-3-030-38748-8_7

Noury, N., Fleury, A., Rumeau, P., Bourke, A., Laighin, G., Rialle, V., & Lundy, J. (2007). Fall detection - principles and methods. *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1663–1666. https://doi.org/10.1109/IEMBS.2007.4352627

Pereira, C., De Oliveira, E., & De Souza, A. (2024). Machine learning applied to edge computing and wearable devices for healthcare: Systematic mapping of the literature. *Sensors*, *24*(19), 6322. https://doi.org/10.3390/s24196322

Poh, S.-C., Tan, Y.-F., Cheong, S.-N., Ooi, C.-P., & Tan, W.-H. (2019). Anomaly detection for home activity based on sequence pattern. *International Journal of Technology*, *10*(7), 1276. https://doi.org/10.14716/ijtech.v10i7.3230

Rubenstein, L. (2006). Falls in older people: Epidemiology, risk factors and strategies for prevention. *Age and Ageing*, *35*(suppl_2), ii37–ii41. https://doi.org/10.1093/ageing/afl084

Sahni, S., Jain, S., & Saritha, S. (2025). A novel ensemble model for fall detection: Leveraging cnn and bilstm with channel and temporal attention. *Automatika*, *66*(2), 103–116. https://doi.org/10.1080/00051144.2025.2450553

Sehairi, K., & Chouireb, F. (2023). Implementation of motion detection methods on embedded systems: A performance comparison. *International Journal of Technology*, *14*(3), 510. https://doi.org/10.14716/ijtech.v14i3.5950

Silva, C., Casilari, E., & García-Bermúdez, R. (2024). Cross-dataset evaluation of wearable fall detection systems using data from real falls and long-term monitoring of daily life. *Measurement*, *235*, 114992. https://doi.org/10.1016/j.measurement.2024.114992

Sucerquia, A., López, J., & Vargas-Bonilla, J. (2017). Sisfall: A fall and movement dataset. *Sensors*, *17*(1), 198. https://doi.org/10.3390/s17010198

Verma, S., Willetts, J., Corns, H., Marucci-Wellman, H., Lombardi, D., & Courtney, T. (2016). Falls and fall-related injuries among community-dwelling adults in the united states. *PLOS ONE*, *11*(3), e0150939. https://doi.org/10.1371/journal.pone.0150939

Wang, X., Ellul, J., & Azzopardi, G. (2020). Elderly fall detection systems: A literature survey. *Frontiers in Robotics and AI*, *7*, 71. https://doi.org/10.3389/frobt.2020.00071

Xu, T., Se, H., & Liu, J. (2021). A fusion fall detection algorithm combining threshold-based method and convolutional neural network. *Microprocessors and Microsystems*, *82*, 103828. https://doi.org/10.1016/j.micpro.2021.103828

Xu, T., Zhou, Y., & Zhu, J. (2018). New advances and challenges of fall detection systems: A survey. *Applied Sciences*, *8*(3), 418. https://doi.org/10.3390/app8030418

Yahyati, C., Lamaakal, I., Maleh, Y., Makkaoui, K., Ouahbi, I., Almousa, M., & Abd El-Latif, A. (2025). A systematic review of state-of-the-art tinyml applications in healthcare, education, and transportation. *IEEE Access*, 1–1. https://doi.org/10.1109/ACCESS.2025.3633575

Yu, X., Jang, J., & Xiong, S. (2021). A large-scale open motion dataset (kfall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors. *Frontiers in Aging Neuroscience*, *13*, 692865. https://doi.org/10.3389/fnagi.2021.692865

Yu, X., Park, S., Kim, D., Kim, E., Kim, J., Kim, W., An, Y., & Xiong, S. (2023). A practical wearable fall detection system based on tiny convolutional neural networks. *Biomedical Signal Processing and Control*, *86*, 105325. https://doi.org/10.1016/j.bspc.2023.105325

Zhou, H., Zhang, X., Feng, Y., Zhang, T., & Xiong, L. (2025). Efficient human activity recognition on edge devices using deepconv lstm architectures. *Scientific Reports*, *15*(1), 13830. https://doi.org/10.1038/s41598-025-98571-2