

Article

# The Impact of the Accelerometer Sampling Rate on the Performance of Machine and Deep Learning Models in Wearable Fall-Detection Systems

Manny Villa <sup>1,2,\*</sup>  and Eduardo Casilari <sup>2</sup> 

<sup>1</sup> Programa de Ingeniería Electrónica, Universidad de Investigación y Desarrollo (UDI), Bucaramanga 680001, Colombia

<sup>2</sup> Departamento de Tecnología Electrónica, Instituto TELMA, Universidad de Málaga, 29071 Málaga, Spain; ecasilari@uma.es

\* Correspondence: mvilla2@udi.edu.co

## Abstract

Population aging has intensified the prevalence of falls among older adults, making automatic Fall Detection Systems (FDS) a key component of telemonitoring and remote care. Among wearable-based approaches, inertial sensors, particularly accelerometers, offer an effective and low-cost alternative for continuous monitoring. However, the impact of the selection of the sampling frequency on model performance remains insufficiently explored. This work seeks to determine the sampling rate that best balances accuracy, stability, and computational efficiency in wearable FDS. Five representative algorithms (CNN-LSTM, CNN, LSTM-BN, k-NN, and SVM) were trained and evaluated using the SisFall dataset at 10, 20, 50, and 100 Hz, followed by a multi-stage validation including the real-fall repositories FARSEEING and Free From Falls, as well as a seven-day continuous monitoring test under real-life conditions. The results show that deep learning architectures consistently outperform traditional classifiers, with the CNN-LSTM model at 20 Hz achieving the best balance of accuracy (98.9%), sensitivity (96.7%), and specificity (99.6%), while maintaining stable performance across all validations. The observed consistency indicates that intermediate frequencies, around 20 Hz and down to 10 Hz, provide sufficient temporal resolution to capture fall dynamics while reducing data volume, which translates into more efficient energy usage compared to higher sampling rates. Overall, these findings establish a solid empirical foundation for designing next-generation wearable fall-detection systems that are more autonomous, robust, and sustainable in long-term IoT-based monitoring environments.

**Keywords:** fall detection systems; wearable devices; sampling frequency; deep learning; CNN-LSTM; inertial sensors; IoT; energy efficiency; telemonitoring



Academic Editor: Zixiang Gao

Received: 1 December 2025

Revised: 22 December 2025

Accepted: 25 December 2025

Published: 26 December 2025

**Copyright:** © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

The aging of the global population has become one of the major public health challenges of the twenty-first century. According to projections by the World Health Organization (WHO), by 2050 the number of people aged 60 years and older is expected to surpass 2.1 billion, accounting for about 22% of the world's population [1]. This demographic shift has significantly increased the occurrence of falls among the elderly, placing them among the main causes of disease burden and death within this age cohort. An estimated

684,000 individuals annually die from fall-related injuries, the majority of whom are over 60 years old [2,3].

Falls represent not only a traumatic physical incident but also a major cause of secondary injuries such as hip fractures, traumatic brain damage, and other musculoskeletal damage. Beyond the physical impact, falls commonly lead to psychological and social repercussions -such as loss of confidence, fear of recurrence, reduced mobility, and even depression. These factors collectively contribute to a gradual decline in quality of life and personal autonomy [4,5]. Previous studies have reported that between 28% and 35% of adults aged 65 years or older experience at least one fall per year, a figure that increases to 42% among individuals over 70 [4]. Nearly half of those affected are unable to stand up without assistance. Consequently, after a fall, a significant proportion remain on the ground for prolonged periods, thereby raising the likelihood of severe medical complications [6].

In this regard, automatic Fall Detection Systems (FDS) have emerged as appealing tools for remote health monitoring and assistance of older individuals or those with limited mobility. These technologies provide instant notifications to caregivers or medical staff, allowing rapid responses that help prevent severe outcomes [7]. Among the different technological strategies, three primary approaches are typically distinguished: vision-based methods, ambient sensor networks, and wearable systems [8]. The first two, often referred to as context-aware systems, face major challenges such as privacy issues, reliance on indoor settings, and high deployment costs. Conversely, wearable devices stand out for their affordability, portability, continuous monitoring capability, and energy efficiency [9]. As the main disadvantage, portable FDSs are chiefly constrained by the hardware limitations (memory, computation capability, battery) of wearable devices.

Within the wearable category, inertial sensors, particularly accelerometers, gyroscopes, and in some cases magnetometers, represent the most widely used technology due to their compactness and low cost [10,11]. These sensors enable the characterization of kinematic patterns associated with falls, allowing them to be distinguished from conventional Activities of Daily Living (ADL) [12]. In this context, the performance of FDS is influenced not only by the sensor configuration or the body location of the device, but also by the algorithms used for signal analysis. Previous research has investigated different methodologies, including rule-based thresholds [8,13–16], machine learning (ML) techniques, and deep learning (DL) frameworks. Threshold-oriented methods are valued for their simplicity and low computational cost, whereas ML and DL models generally provide superior accuracy, though at the price of higher system complexity and greater energy consumption [12,16–19]. Recently, hybrid approaches have been proposed that combine a preliminary detection stage using threshold-based methods, which take advantage of their simplicity and low energy consumption. These approaches apply ML/DL techniques in a subsequent stage to refine classification and reduce false positives, achieving a balance between efficiency and performance [20–22].

A particularly critical aspect in this type of system is the sampling frequency of inertial sensors. The literature reveals a wide variability in the frequencies employed, ranging from 10 Hz to values as high as 800 Hz [23]. In this context, several studies have explored its impact on activity recognition.

As early as 2006, U. Maurer et al. [24] introduced the eWatch device, a multisensor wrist-worn platform equipped with a biaxial accelerometer designed to recognize various body postures. The authors evaluated several classification algorithms, including Decision Trees, k-Nearest Neighbors (k-NN), Naïve Bayes, and Bayes Net, and concluded that Decision Trees offered the most suitable trade-off between accuracy and computational requirements. The recognition accuracy improved as the sampling frequency increased,

reaching stability between 15 and 20 Hz, which ultimately guided the adoption of a 20 Hz sampling rate in the final implementation.

Santoyo-Ramón et al. [25] carried out a systematic investigation on how sampling frequency influences wearable fall-detection systems based on accelerometer data. In their work, they implemented a Convolutional Neural Network (CNN) model, which was trained and evaluated using 15 publicly available datasets. Through progressive decimation of the time series, they observed that sensitivity and specificity remained above 95% for frequencies in the 15–20 Hz range. Noticeable degradation occurred only below 10–15 Hz. The power spectrum analysis and low-pass filtering experiments at 10–20 Hz corroborated the dominance of low-frequency components of human movements, leading the authors to recommend approximately 20 Hz as a compromise between performance and energy consumption. A similar conclusion has been reached in other studies [26–29] that have analyzed the relationship between the frequency content and the biomechanics a conventional human movement during daily life activities.

In contrast, Ajerla et al. [30] evaluated sampling frequency in a real-time wearable fall-detection system deployed at the edge. Using accelerometer data collected between 12.5 Hz and 200 Hz, and an LSTM-based classifier, they reported that a sampling rate of 50 Hz achieved the best detection performance, highlighting the influence of system-level factors beyond signal frequency content.

Additionally, Liu et al. [31] examined how varying sampling frequencies influence wearable fall-detection systems that utilize machine learning techniques. Their analysis combined data from the public SisFall repository and a proprietary dataset. By gradually decreasing the sampling frequency from 200/128 Hz down to 3 Hz, they assessed the performance of several classifiers, including Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Naïve Bayes (NB), and Decision Trees (DT). The findings indicated that a sampling frequency close to 22 Hz was sufficient to maintain an accuracy of at least 97% in most models, while the SVM with a radial basis kernel achieved comparable outcomes even at frequencies as low as 5.8 Hz. Concerning this, the choice of sampling frequency becomes particularly relevant in Internet of Things (IoT) scenarios, where wearable devices are frequently linked through energy-efficient wide-area communication systems, for instance LoRaWAN and Sigfox. In such systems, the available bandwidth and transmission time are strongly constrained [32]. Transmitting data at high frequencies increases energy consumption and reduces the device's autonomy. By contrast, reducing the sampling rate decreases data volume, accelerates transmission, and extends battery life without compromising system accuracy [31]. Table 1 summarizes the studies presented above, highlighting the sampling frequencies, algorithms, and reported performance.

Despite the advances achieved in wearable fall-detection systems, several limitations persist in prior work. Many studies adopt fixed accelerometer sampling frequencies without systematic justification, often based on empirical choices, which prevents a clear assessment of their impact on detection performance, model stability, and computational efficiency [24,25,31]. In addition, most approaches focus on a single class of algorithms, either traditional machine learning or deep learning, limiting fair comparisons across modeling strategies and the evaluation of hybrid solutions under a uniform experimental protocol [33–35]. Furthermore, evaluations are frequently conducted using simulated falls under laboratory conditions, with limited validation on real-fall datasets or continuous real-life monitoring scenarios [36,37]. Finally, the practical implications of sampling frequency on energy consumption, data volume, and device autonomy, which are critical factors in IoT-based wearable systems, are often overlooked.

**Table 1.** Summary of studies on sampling frequency in wearable fall-detection systems.

Model	Algorithm/Model	Sampling Frequency (Hz)	Main Findings
Maurer et al. [24]	DT, k-NN, NB, Bayes Net	1–30 (downsampled from 50 Hz)	Accuracy stabilized at 15–20 Hz; Decision Trees showed the best accuracy–complexity trade-off; final system implemented at 20 Hz
Santoyo-Ramón et al. [25]	CNN	1–140 Hz (effective, via decimation)	Sensitivity and specificity > 95% at 15–20 Hz
Liu et al. [31]	SVM (RBF), k-NN, NB, DT	3–200 (200/128 Hz original)	Accuracy $\geq$ 97% at ~22 Hz; SVM remains effective at lower sampling rates.
Ajerla et al. [30]	LSTM (edge framework)	12.5, 25, 50, 100 and 200 (real-time data collection)	50 Hz performed best; below 50 Hz missed falls; >50 Hz similar. Best overall system: 95.8% accuracy (waist + wrist).
This work	CNN-LSTM, CNN, LSTM, SVM, k-NN	10, 20, 50 and 100 (downsampled from 200 Hz)	20 Hz provides best accuracy–efficiency trade-off

Therefore, this study aims to determine and validate the optimal sampling frequency that ensures the best balance between accuracy and sensitivity in wearable fall-detection models. Four sampling rates (10, 20, 50, and 100 Hz) were analyzed using the SisFall dataset [38], evaluating both deep learning architectures (CNN-LSTM, CNN, and LSTM with batch normalization) and classical machine learning algorithms (k-NN with 5 and 15 neighbors, and SVM). Beyond previous works, this study extends the analysis through a multi-stage validation, including assessments on external real-fall datasets (FARSEEING and Free From Falls) and on a custom wearable prototype operating at 20 Hz. The objective is to identify the sampling frequency that provides the most stable and accurate performance, confirming its suitability for fall detection under real-use conditions.

## 2. Methodology

### 2.1. Data and Preprocessing

The SisFall dataset [38], one of the most popular repositories considered by the fall-detection literature [29], was used in this study. This dataset comprises a total of 4505 motion recordings (2707 ADL and 1798 simulated falls) collected from 38 participants (19 older adults and 19 young subjects) [39]. Each record contains triaxial acceleration and angular velocity signals sampled at 200 Hz using sensors placed on the waist. The experimental protocol includes 15 types of falls (frontal, lateral, backward, from standing, while sitting, syncope, trips, and slips, among others) and 19 types of ADL (walking, climbing stairs, sitting, lying down, etc.), some of which exhibit motion patterns similar to falls. This similarity increases the classification challenge.

For the experimental setup, only the ADXL345 accelerometer (Analog Devices Inc., Wilmington, MA, USA) was used to record motion along the X, Y, and Z axes. Data fields corresponding to rotational parameters and measurements from a secondary accelerometer were omitted, as the proposed fall-detection method focuses exclusively on the acceleration magnitude derived from a single sensor.

The raw signals collected from the ADXL345 were converted into gravity units (g) to ensure accurate physical interpretation. The conversion followed the specifications of

the sensor, taking into account its 13-bit resolution and a measurement range of  $\pm 16$  g, as expressed in the following equation:

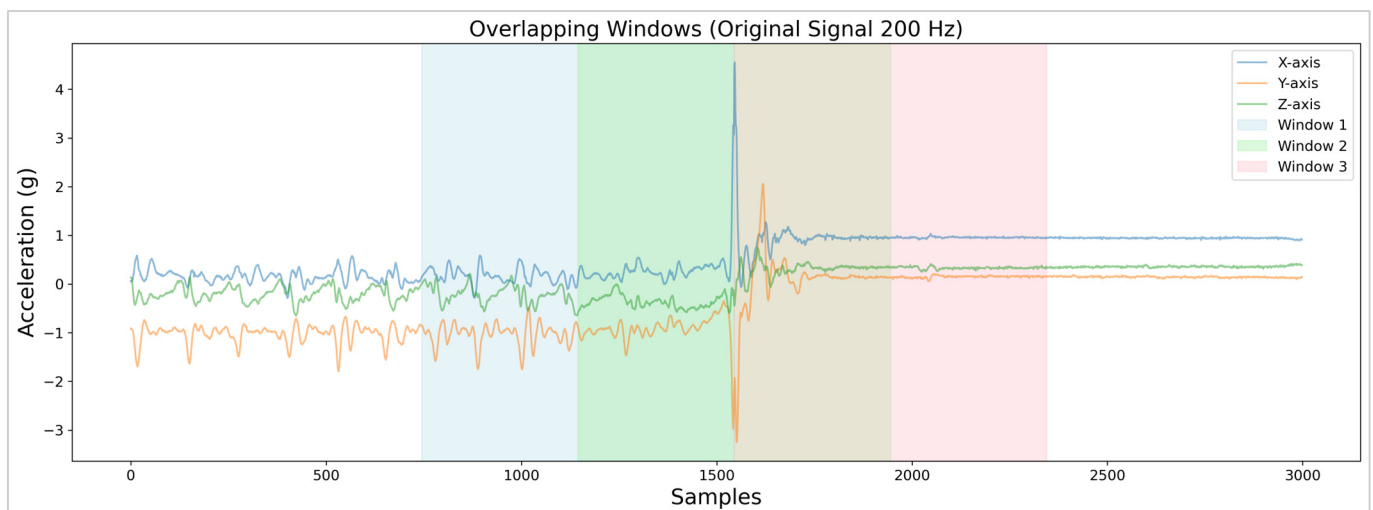
$$a(g) = \frac{2 \times Range}{2^{Resolution}} \times D_{bits} \quad (1)$$

where  $a(g)$  represents the acceleration in gravity units,  $Range$  is the accelerometer's measurement range,  $Resolution$  denotes the bit depth of the sensor, and  $D_{bits}$  corresponds to the raw digital output. In this way, each digital reading was transformed into acceleration values expressed in g, providing a consistent basis for subsequent signal analysis.

To analyze the effect of sampling frequency, the original 200 Hz signals were processed and re-sampled in Python 3.12.12 within a Google Colab environment. In particular, a linear-phase FIR filter was applied to each trace using the `firwin` function [40], which was used to adjust the cutoff frequency to the corresponding Nyquist frequency for each new sampling rate. Subsequently, a polyphase resampling was performed with `resample_poly` function [41] to obtain equivalent versions of the time series in the dataset at 100, 50, 20, and 10 Hz. This approach enabled explicit control of antialiasing and precisely preserved the relevant spectral content at each frequency.

Each record in the repository (representing a single action: fall or ADL) was segmented into 4 s windows centered on the maximum of the acceleration magnitude ( $|\vec{a}|$ ), which was computed from each instance of the acceleration components ( $a_x$ ,  $a_y$  and  $a_z$ ) using Equation (2). In the case of the series generated during falls, this maximum corresponds to the instant of impact of the body against the floor. The selected 4 s interval is wide enough to represent the typical dynamics of both the pre-fall and post-fall phases [42]. In some cases, a 50% overlap was applied between consecutive windows to increase the number of available samples while controlling the redundancy between adjacent temporal segments, as illustrated in Figure 1, following a processing framework commonly adopted in wearable fall-detection studies [12].

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (2)$$



**Figure 1.** Four-second consecutive windows (with 50% overlap) extracted around the impact generated by a fall. A visible color change appears where two consecutive windows overlap.

The resulting segments were exported as CSV (Comma Separated Values) files, merged into a single dataset, and cleaned to retain only the accelerometer magnitude, which was normalized using the Z-score technique, along with the corresponding binary classification label describing the movement (fall or ADL). The obtained patterns (a set of 4 s sequence



of acceleration magnitude plus the labels) were then used to generate the training, validation and testing data. For model evaluation, a subject-independent protocol with ten participants was adopted. Six individuals were assigned to training, two to validation, and two to testing, ensuring that no participant was included in more than one subset. This experimental design provided a realistic estimation of model performance, consistent with the methodological approach described by Salah et al. [12].

## 2.2. Evaluated Models

Five representative approaches from the fall detection literature were evaluated, considering their diversity in computational complexity and modeling capability. All models were trained and validated using resampled signals at 10, 20, 50, and 100 Hz, allowing for a consistent comparative analysis. The configuration of each model was defined based on a combination of previous evidence from the literature [33–35,43–46] and exploratory experiments conducted in our environment [37–43].

For the advanced neural architectures, including CNN, LSTM combined with batch normalization, and CNN–LSTM models, a grid search was initially applied to explore hyperparameters, evaluating learning rates in the range of  $10^{-5}$  to  $10^{-1}$  and convolutional filter sizes between 16 and 256, followed by empirical fine-tuning using validation accuracy as the main criterion. The number of epochs and batch size were kept constant to ensure comparability. For k-NN, values of  $K = 1–20$  were tested on the validation set, selecting  $K = 5$  and  $K = 15$  as representative configurations. For SVM, a linear kernel with  $C = 1.0$  was chosen, consistent with related studies and representing a balance between bias and variance. Each training process was repeated five times to verify result stability, and the best-performing model from each configuration was selected for further analysis. Table 2 summarizes the final configuration of the models evaluated at the different sampling frequencies.

**Table 2.** Model configurations across sampling frequencies after hyperparameter grid search.

Model	Common Parameters	Sampling Rate (Hz)	Model Configuration
CNN-LSTM	Window = 4 s; 70 epochs; 5 runs; Batch = 32; LSTM = 64–64; Dense = 128 + Dropout = 0.4; Optimizer = Adam; Loss = BCE; Output = Binary classification (Fall/No-Fall)	10	Conv(32,64), Dropout = 0.2/0.2, no PrePool
		20	Conv(32,64), Dropout = 0.2/0.2, no PrePool
		50	Conv(64,128), Dropout = 0.25/0.25, with PrePool
		100	Conv(128,256), Dropout = 0.3/0.3, with PrePool
CNN	Window = 4 s; 70 epochs; 5 runs; Batch = 32; 3 Conv + BN + Dropout + GAP; Dense = 128 + Dropout = 0.4; Optimizer = Adam; Loss = BCE; Output = Binary classification (Fall/No-Fall)	10	Conv(32,64), Dropout = 0.2, no PrePool
		20	Conv(32,64), Dropout = 0.2, no PrePool
		50	Conv(64,128), Dropout = 0.25, with PrePool
		100	Conv(128,256), Dropout = 0.3, with PrePool
LSTM + BN	Window = 4 s; 70 epochs; 5 runs; Batch = 32; LSTM(32,32) + BN + Dropout = 0.2; Dense = 64 + Dropout = 0.5; Optimizer = Adam; Loss = BCE; Output = Binary classification (Fall/No-Fall)	10	Window = 40 samples
		20	Window = 80 samples
		50	Window = 200 samples
		100	Window = 400 samples
k-NN	Window = 4 s; StandardScaler; Validation sweep $K = 1–20$ ; Test with fixed $K$ (5 or 15); Metric = Accuracy; Output = Binary classification (Fall/No-Fall)	10	Window = 40 samples, $K = 5$ or 15
		20	Window = 80 samples, $K = 5$ or 15
		50	Window = 200 samples, $K = 5$ or 15
		100	Window = 400 samples, $K = 5$ or 15
SVM	Window = 4 s; Linear kernel; $C = 1.0$ ; StandardScaler; Optional kernels = RBF/Poly; Metric = ROC-AUC; Output = Binary classification (Fall/No-Fall)	10	Window = 40 samples
		20	Window = 80 samples
		50	Window = 200 samples
		100	Window = 400 samples

As shown in Table 2, the configurations illustrate a balance between more complex architectures, capable of capturing temporal and spatial dependencies (CNN, LSTM + BN, and CNN-LSTM), and classical models with lower computational cost (k-NN and SVM). This diversity allowed the evaluation not only of performance in terms of accuracy, but also of the practical implications related to model complexity and training time, aspects that are analyzed in the following sections.

### 2.3. Experimental Settings

All experiments were conducted using Python 3.12.12 within the Google Colab cloud-computing environment [47]. Signal filtering and resampling were performed using SciPy v1.16.3, classical machine learning models were implemented with Scikit-learn v1.6.1, and deep learning architectures were developed using TensorFlow/Keras v2.19.0. Model training and evaluation were accelerated using an NVIDIA Tesla T4 GPU, as provided by the Colab runtime, ensuring efficient execution and reproducibility of the experiments.

### 2.4. Evaluation Metrics

To evaluate the effectiveness of the models in detecting falls, we compute five standard indicators commonly considered to assess binary classifiers: accuracy, sensitivity (or recall), specificity, precision (positive predictive value), and F1-score. These indicators were obtained from the confusion matrix and describe both the proportion of falls correctly recognized and the model's capability to differentiate them from activities of daily living (ADL) [34]. They are formally defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (5)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \quad (7)$$

In these equations, *TP* (True Positives) refers to the number of fall events correctly recognized by the model, whereas *TN* (True Negatives) corresponds to activities of daily living (ADLs) that were accurately identified as non-fall events. Similarly, *FP* (False Positives) indicates instances of ADL that were mistakenly classified as falls, and *FN* (False Negatives) represents real falls that went undetected. Sensitivity is directly linked to the proportion of actual falls that the models are able to identify, while Specificity describes the ability of the classifiers to avoid false alarms. Precision in turns quantifies how many of the detected events were actual falls, while Accuracy represents the global hit ratio of the models and the F1-score reflects the harmonic mean between precision and sensitivity, serving as a more balanced metric of overall detection performance. For all neural models, a fixed probabilistic decision threshold of 0.5 was applied to the sigmoid output to convert predicted probabilities into binary class labels (fall/no-fall) when computing these metrics.

### 2.5. Experimental Validation of the Selected Model

Based on the comparative results, the model that demonstrates the best overall performance will undergo an additional validation process. This stage will be conducted at the sampling frequency identified as optimal in our experiments and will comprise two complementary steps: (i) the evaluation of the system using external datasets of real-world falls, specifically FARSEEING [36] and the Free From Falls (FFF) study [48], in order to verify the model's generalization capability; and (ii) implementing and testing the selected classifier on a wearable device based on the Arduino Nano 33 BLE Sense Rev2 (Arduino S.r.l., Monza, Italy) [49], which incorporates a triaxial accelerometer positioned at the waist, as shown in Figure 2, to assess its performance under conditions closer to real-world use (see our work in [11] for further details of the employed prototype of FDS).

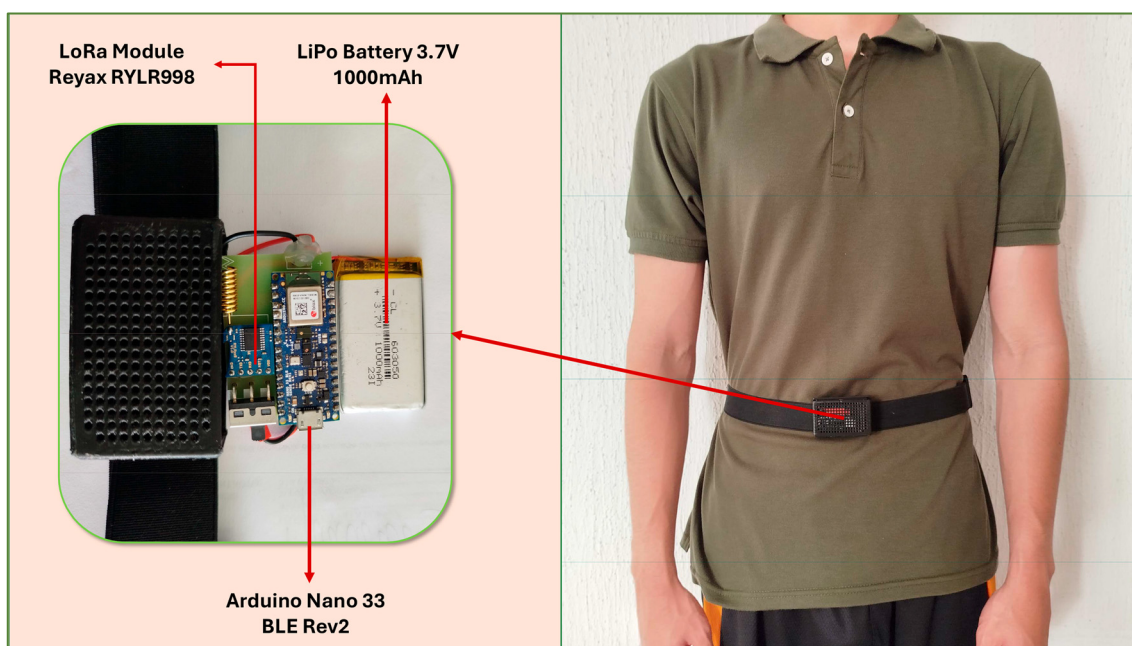


Figure 2. Hardware setup and on-body placement of the wearable device [11].

## 3. Results

### 3.1. Comparative Performance Across Sampling Rates

To analyze the impact of temporal resolution on fall detection, a comparative evaluation was conducted using five representative model types—CNN, LSTM (Long Short-Term Memory) with batch normalization, a combination of CNN and LSTM, k-NN (evaluated with  $k = 5$  and  $k = 15$ ), and SVM—at four sampling frequencies (10, 20, 50, and 100 Hz). Table 3 presents the results in terms of accuracy, sensitivity, and specificity. For learning-based models, the reported Best Accuracy (%) corresponds to the best-performing run among five independent executions, while Accuracy (mean (%)  $\pm \sigma$ ) reflects the average performance and variability across these runs; sensitivity and specificity are reported for the best-performing run. As can be observed from the table, deep learning frameworks consistently achieved superior performance compared with traditional machine-learning models, emphasizing the strong impact of the sampling frequency on the effectiveness of several models in differentiating falls from activities of daily living (ADL).



**Table 3.** Performance of All Models on SisFall at Different Sampling Rates.

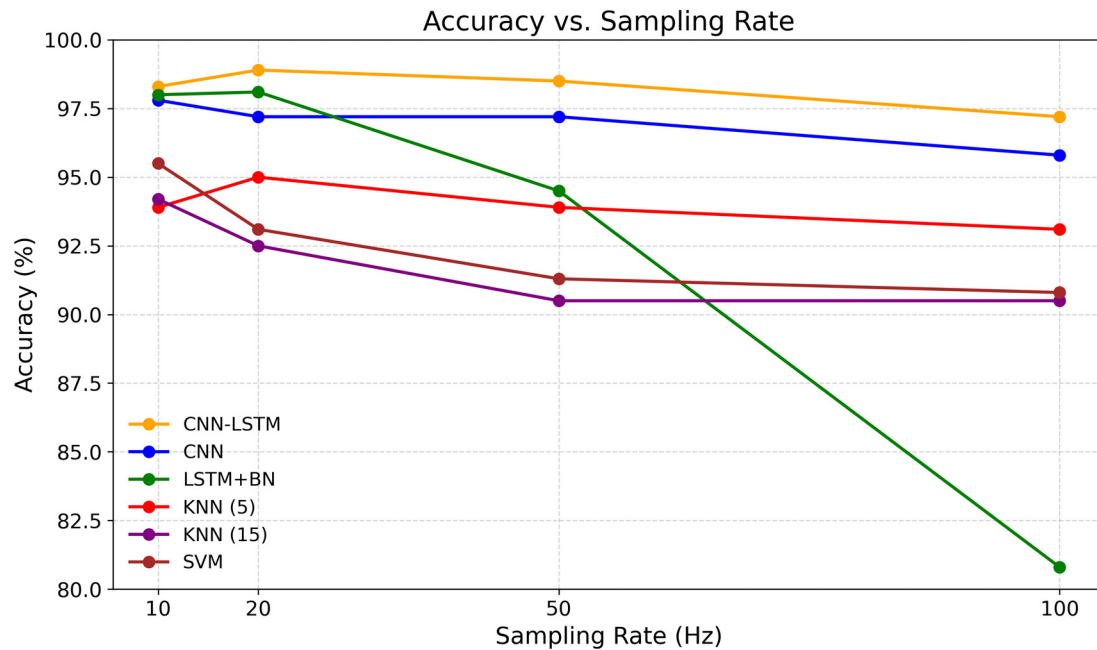
Model	Sampling Rate (Hz)	Best Accuracy (%)	Sensitivity (%)	Specificity (%)	Accuracy (Mean (%) $\pm \sigma$ )
CNN-LSTM	10	98.33	96.67	98.89	97.56 $\pm$ 0.77
	<b>20</b>	<b>98.89</b>	<b>96.67</b>	<b>99.63</b>	<b>98.11 <math>\pm</math> 0.54</b>
	50	98.61	97.78	98.89	97.33 $\pm$ 0.80
	100	97.22	97.78	97.04	96.06 $\pm$ 0.73
CNN	<b>10</b>	<b>97.78</b>	<b>96.67</b>	<b>98.15</b>	<b>97.61 <math>\pm</math> 0.22</b>
	20	97.22	94.44	98.15	96.50 $\pm$ 0.52
	50	97.22	92.22	98.89	95.61 $\pm$ 1.13
	100	95.83	88.89	98.15	95.50 $\pm$ 0.27
LSTM (with Batch Norm)	10	98.06	93.33	99.63	96.72 $\pm$ 0.75
	<b>20</b>	<b>98.06</b>	<b>95.56</b>	<b>98.89</b>	<b>97.06 <math>\pm</math> 0.89</b>
	50	94.44	96.67	93.70	89.72 $\pm$ 5.67
	100	80.83	37.78	95.19	79.94 $\pm$ 0.59
K-NN (5 neighbors)	10	93.89	83.33	97.41	N/A
	<b>20</b>	<b>95.00</b>	<b>83.33</b>	<b>98.89</b>	
	50	93.89	80.00	98.52	
	100	93.06	78.89	97.78	
K-NN (15 neighbors)	<b>10</b>	<b>94.17</b>	<b>83.33</b>	<b>97.78</b>	N/A
	20	92.50	70.00	100.00	
	50	90.56	64.44	99.26	
	100	90.56	62.22	100.00	
SVM	<b>10</b>	<b>95.56</b>	<b>93.33</b>	<b>96.30</b>	N/A
	20	93.06	90.00	94.07	
	50	91.39	90.00	91.85	
	100	90.83	90.00	91.11	

Bold values indicate the best performance for each model across the evaluated sampling rates. N/A denotes metrics not available for certain models.

As illustrated in Figure 3, the CNN-LSTM model achieved the highest and most stable performance, peaking at 20 Hz (accuracy = 98.89%, specificity = 99.63%). This result confirms that intermediate sampling rates effectively capture the distinctive motion dynamics of falls while minimizing redundancy and low-frequency noise often associated with ADL. In contrast, the standalone LSTM model showed a pronounced drop in accuracy above 50 Hz, suggesting that excessive temporal resolution may interfere with the preservation of sequential dependencies while unnecessarily increasing model complexity and the number of trainable parameters. Meanwhile, classical approaches such as KNN and SVM exhibited limited adaptability, with noticeable degradation at 100 Hz.

#### Controlled Evaluation Using a Unified CNN-LSTM Architecture

To isolate the effect of sampling frequency from architectural variability, an additional controlled experiment was conducted using a unified CNN-LSTM architecture across all sampling rates. For 10 Hz and 20 Hz, the CNN-LSTM architecture was already identical, whereas for 50 Hz and 100 Hz the same configuration was enforced to eliminate architectural tuning effects. The performance obtained with this unified architecture is summarized in Table 4. As can be observed, the performance trends reported in Table 3 are preserved, confirming that the observed differences are mainly driven by sampling frequency rather than architectural adjustments.



**Figure 3.** Accuracy vs. Sampling Rate.

**Table 4.** Performance of the CNN-LSTM model using a unified architecture across sampling rates.

Model	Common Parameters and Model Configuration	Sampling Rate (Hz)	Best Accuracy (%)	Sensitivity (%)	Specificity (%)
CNN-LSTM	Window = 4 s; 70 epochs; 5 runs; Batch = 32; LSTM = 64–64; Dense = 128 + Dropout = 0.4; Optimizer = Adam; Loss = BCE; Output = Binary classification (Fall/No-Fall); Conv(32,64), Dropout = 0.2/0.2, no PrePool	10	98.33	96.67	98.89
		20	98.89	96.67	99.63
		50	97.50	97.78	97.41
		100	97.22	98.89	96.67

### 3.2. Optimal Model Configuration and In-Depth Evaluation

After comparing all models across sampling rates, the optimal configuration for each algorithm was determined to facilitate performance comparison. Table 5 summarizes these best results per model, highlighting the sampling rate at which each achieved its highest accuracy, sensitivity, and specificity.

**Table 5.** Best Results per Model on SisFall.

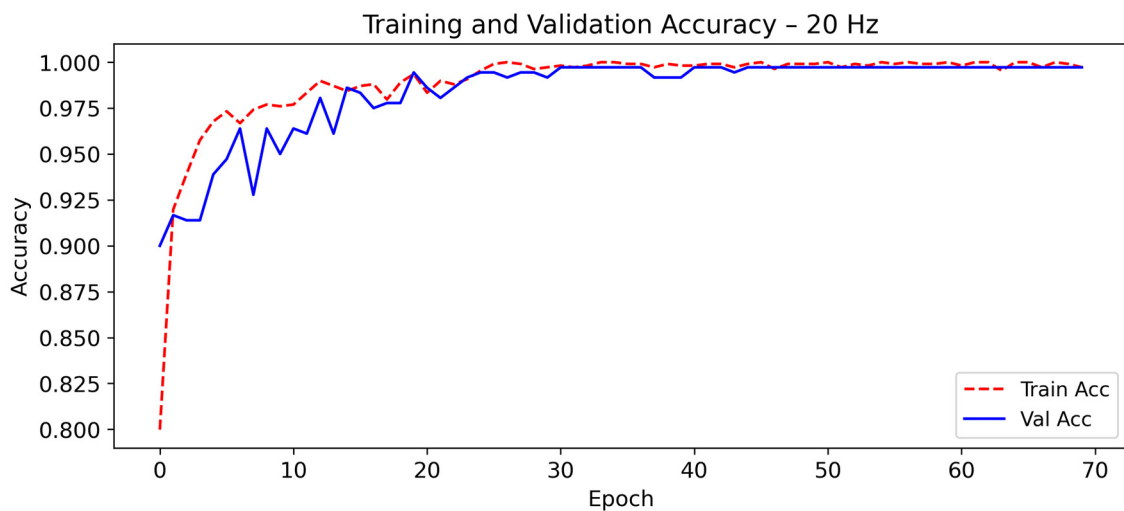
Model	Sampling Rate (Hz)	Best Accuracy (%)	Sensitivity (%)	Specificity (%)
CNN-LSTM	20 Hz	98.89	96.67	99.63
LSTM + BatchNorm	20 Hz	98.06	95.56	98.89
CNN	10 Hz	97.78	96.67	98.15
SVM	10 Hz	95.56	93.33	96.30
KNN (5 neighbors)	20 Hz	95.00	83.33	98.89
KNN (15 neighbors)	10 Hz	94.17	83.33	97.78

As shown, deep learning models maintained superior performance across all sampling rates, with the CNN-LSTM at 20 Hz achieving the most balanced results (accuracy = 98.89%, sensitivity = 96.67%, specificity = 99.63%). The LSTM + Batch Normalization model followed closely, confirming that architectures capable of modeling temporal dependencies

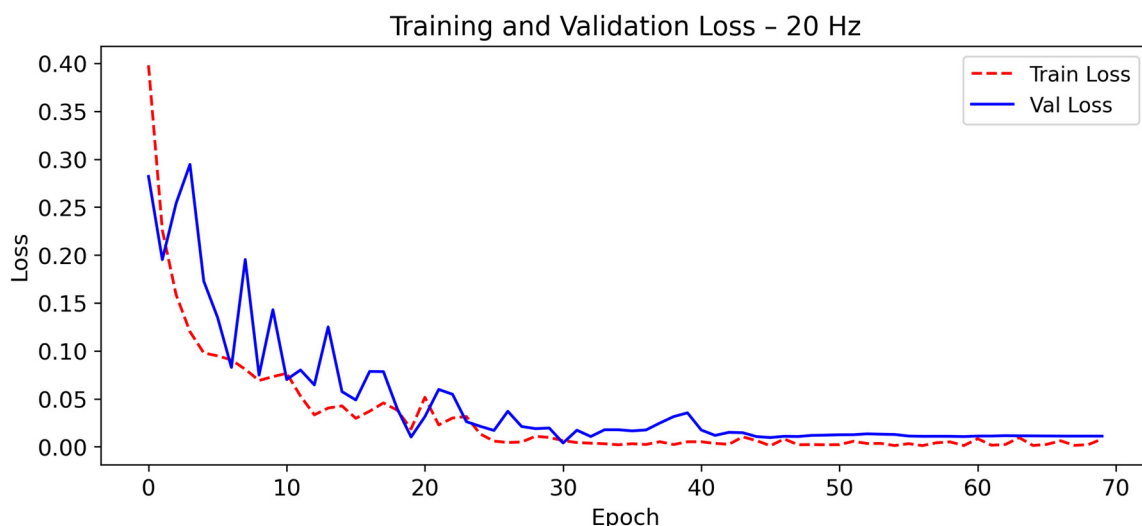
outperform traditional classifiers such as SVM and k-NN, which exhibit noticeable degradation in sensitivity. These findings are consistent with previous studies, where hybrid and recurrent models demonstrated improved robustness and generalization in fall-detection tasks [44,45].

The predominance of intermediate frequencies (10–20 Hz) among the best configurations supports prior evidence that most discriminative motion dynamics related to falls occur below 20 Hz, where signal information remains sufficiently rich while computational and energy costs are reduced [25]. Consequently, the CNN-LSTM at 20 Hz was selected for an in-depth evaluation of its learning behavior, convergence stability, and classification reliability.

As shown in Figures 4 and 5, the model exhibits rapid convergence during the first 15 epochs. After this point, both training and validation curves stabilize with minimal fluctuations.



**Figure 4.** Evolution of the training and validation accuracy of CNN-LSTM model at 20 Hz as a function of the number of epochs during training.

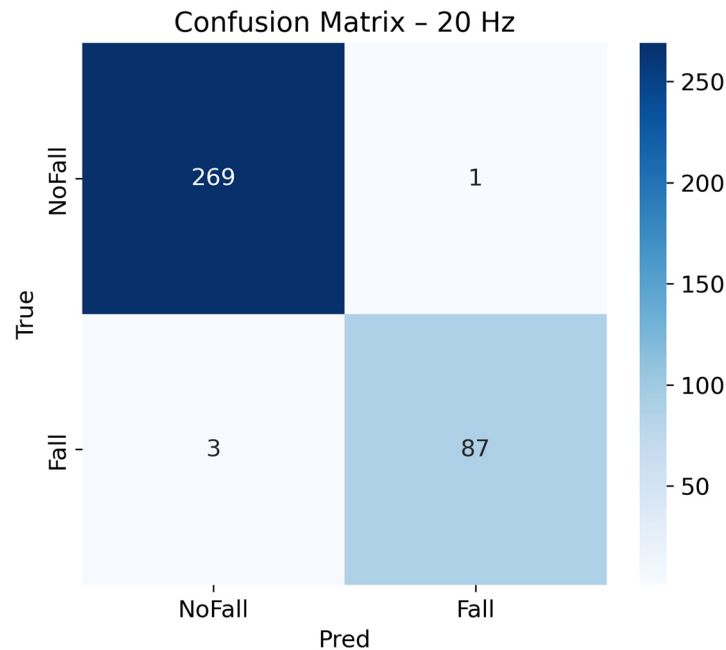


**Figure 5.** Evolution of the training and validation loss of CNN-LSTM model at 20 Hz as a function of the number of epochs during training.

The close alignment between training and validation curves confirms the strong generalization capability of the model and the absence of overfitting. This stability is typical of well-regularized CNN-LSTM architectures, where convolutional layers extract

discriminative motion features and LSTM units capture temporal dependencies necessary to distinguish falls from ADL. Similar robustness has been reported in multimodal activity-recognition networks [50,51], confirming the suitability of this hybrid topology for noise-tolerant and energy-efficient fall detection [52].

The confusion matrix in Figure 6 further corroborates these findings. Out of 270 non-fall activities in the test set, 269 were correctly classified, and out of 90 fall events, 87 were correctly identified. Thus, the classifier only generated one false positive and three false negatives.



**Figure 6.** Confusion matrix of CNN-LSTM at 20 Hz on the test set.

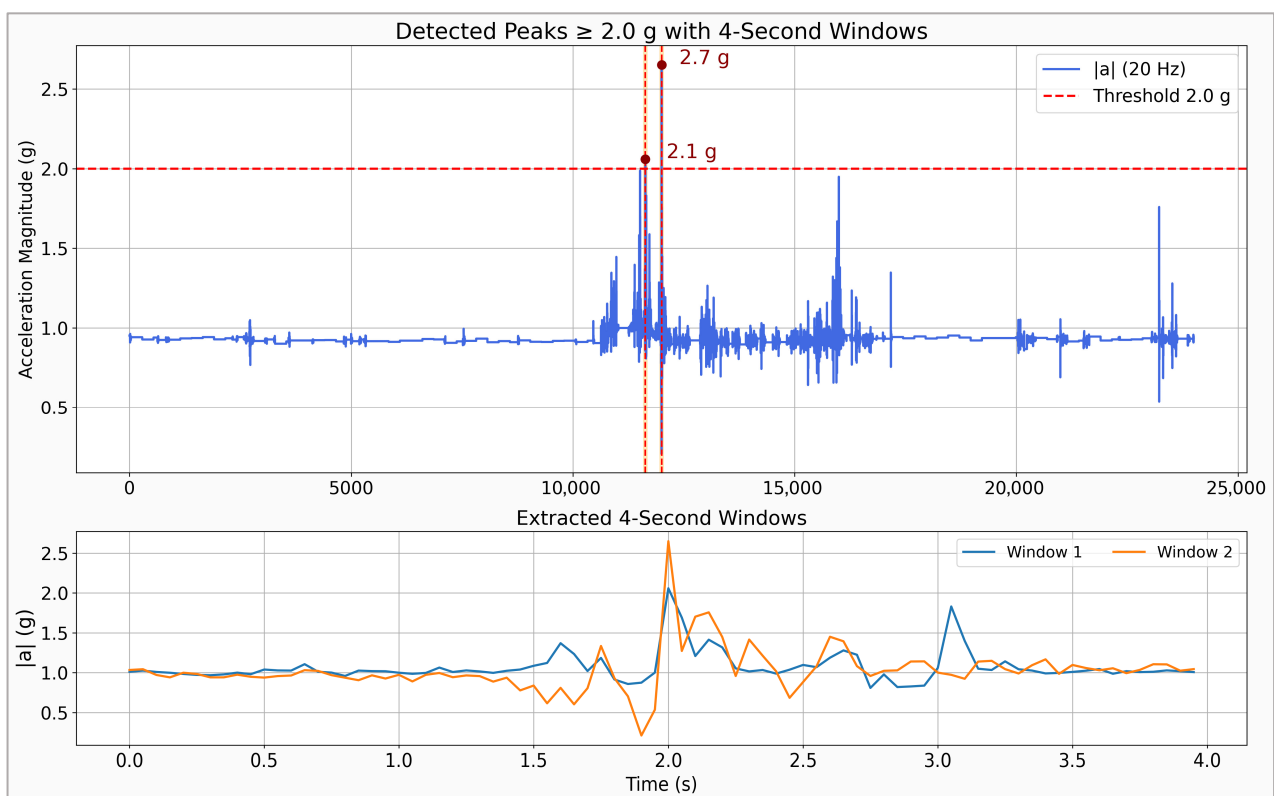
These results yield a precision (PPV) and negative predictive value of  $\approx 98.9\%$ , with an F1-score of 97.8%, confirming the strong discriminative capability of the CNN-LSTM model. The low false-alarm rate is particularly relevant in remote-monitoring scenarios, where excessive false positives can cause caregiver alert fatigue [53]. Overall, the empirical optimum at  $\approx 20$  Hz aligns with previous studies and represents the best trade-off between accuracy and energy efficiency for IoT-based wearable fall-detection systems [25,54,55].

### 3.3. Validation on External Datasets (FARSEEING and FFF)

To assess the generalization capability of the model beyond the training environment, an additional cross-dataset evaluation was conducted using two external repositories containing real fall events: FARSEEING [36] and FFF (Free From Falls) project [48] datasets. In this phase, the optimal configuration of the CNN-LSTM model, previously trained and validated on the SisFall dataset, was applied at a sampling frequency of 20 Hz, which had yielded the most balanced performance across all tested rates. For this purpose, the original series in these two datasets were resampled to the common frequency of 20 Hz following the procedure described in Section 2.1. The experiments were executed in Google Colab (v2024.10) [47], automating the segmentation of windows, inference, and computation of performance metrics. Both datasets include accelerometer signals recorded under real-life conditions but differ in population characteristics and acquisition protocols, allowing the evaluation of the model's robustness against variations in environment and sensor placement.

The FARSEEING dataset [36], developed by the European consortium of the same name, comprises more than 200 real falls recorded between 2012 and 2015 in older adults

during daily life activities. The sensors, one triaxial accelerometer positioned on the subject's lower back or thigh, operated at sampling rates of 20 Hz or 100 Hz. Figure 7 illustrates one of the movements from the FARSEEING dataset, highlighting the detected peaks and the corresponding 4 s windows that were extracted based on those peaks to test the model. Following this procedure on 22 extended recordings, 104 four-second windows with accelerations equal to or above 2 g were extracted. Due to missing timestamps, the absolute maximum peak per recording was labeled as the fall, whereas secondary peaks were treated as ADLs. These samples were analyzed using the CNN-LSTM model with a probabilistic threshold of  $p > 0.4$ . The model correctly detected 18 of the 22 real falls (sensitivity = 81.8%), produced three false detections, and correctly classified 79 of 82 non-fall windows, achieving an overall accuracy of 93.3% and a specificity of 96.3%. The false alarms corresponded to abrupt but non-fall movements, whereas the missed events involved lower-amplitude acceleration patterns.



**Figure 7.** Examples of the detected maxima of the acceleration module and 4 s windows extracted from one trace in FARSEEING dataset. The dotted horizontal line indicates the acceleration threshold, while the dotted vertical lines mark acceleration peaks that exceeded this threshold.

The Free From Falls (FFF) project dataset [48] contains the traces captured during real fall episodes from individuals with multiple sclerosis, recorded during eight weeks of continuous monitoring in their homes using a triaxial accelerometer with a sampling rate of 50 Hz. The sensors were uniformly placed on the subjects' lower back, ensuring consistent signal acquisition across participants. To process the continuous stream, a pre-selection threshold of 1.1 g was applied to extract candidate four-second windows. Additionally, a minimum separation of 40 s was enforced between detections to prevent counting the same fall event multiple times. Using a classification threshold of 0.85 for the CNN-LSTM model, 690 four-second windows extracted from the 49 FFF traces were analyzed. The test with the trained model yielded 48 true positives, one false negative, and seven false positives, resulting in an accuracy of 98.3%, a sensitivity of 97.9%, and a specificity of 98.9%.



confirming its high reliability under controlled experimental conditions and traces captured during real falls.

Compared to FARSEEING (accuracy = 93.3%, sensitivity = 81.8%), this superior performance is mainly attributed to the greater homogeneity in sensor placement and recording conditions in FFF, while the variability in sensor locations and sampling rates in FARSEEING may have introduced kinematic diversity and a slightly reduced sensitivity.

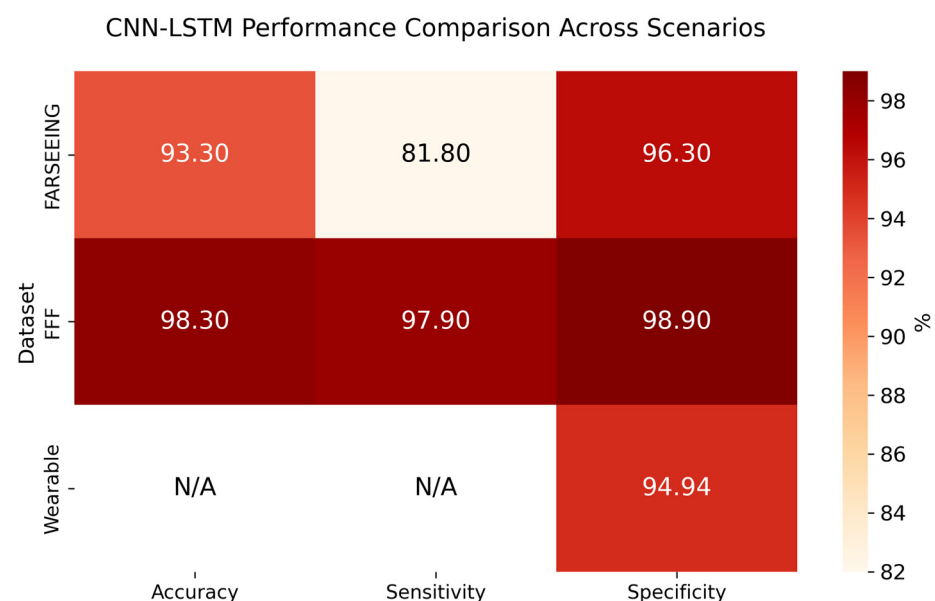
### 3.4. Experimental Validation on the Wearable Device

Following the evaluation with external datasets containing real fall events, an additional validation was conducted to examine the system's stability and the occurrence of false alarms under real-world usage conditions. To this end, the model was implemented on the wearable prototype based on the Arduino Nano 33 BLE Sense Rev2 presented in [11]. The device incorporated a triaxial accelerometer positioned at the waist and was configured to sample data at 20 Hz with an threshold of 2 g for the measured acceleration module. Each time this threshold was exceeded, the system automatically generated a 4 s window, which was subsequently processed by the CNN-LSTM model using a probabilistic decision threshold of 0.5.

The real-life validation was performed over seven consecutive days, during which the participant transporting the detector engaged in routine low-intensity daily activities such as walking, brief runs to cross streets, climbing stairs, riding a motorcycle, traveling as a taxi passenger, working in an office environment, eating, and performing common household tasks. During this period, the device analyzed 1147 observation windows that were suspected of being caused by a fall after exceeding the acceleration threshold. Of these, 58 were incorrectly classified as falls, yielding a false-positive rate of 5.06% and a specificity of 94.94%.

Although no actual falls occurred during the trial, the results demonstrate the model's ability to maintain stable classification performance when exposed to typical daily life movements and its low tendency to trigger false alarms, confirming its reliability in real-world continuous-monitoring scenarios.

As shown in Figure 8, the CNN-LSTM model preserved stable and consistent performance across all validation scenarios. The comparison among FARSEEING, FFF, and wearable evaluations highlights its robustness to variations in data acquisition and its reliability under continuous monitoring conditions.



**Figure 8.** CNN-LSTM performance across external validation and real-life data from the wearable.

## 4. Discussion

The obtained results show that the sampling frequency may exert a noteworthy influence on the performance of wearable Fall Detection Systems (FDS), revealing an optimal point around 20 Hz, where the highest detection capability is achieved without generating unnecessary data volume. Consistent with previous studies, we found that sampling rates above approximately 50–100 Hz yield marginal or negligible improvements in accuracy. Moreover, in some deep learning-based models, increasing the sampling rate can even lead to a degradation in the quality metrics of the detection decision. Santoyo-Ramón et al. [25] reported that a frequency of 15–20 Hz is sufficient to maintain sensitivities and specificities above 95%, while Liu et al. [31] found that 22 Hz was enough to achieve ~97% accuracy across multiple classifiers. Our experiments fully support these conclusions: increasing temporal resolution beyond ~20 Hz did not enhance model performance and, in the case of the pure LSTM architecture, even caused a sharp drop in accuracy at 100 Hz. This phenomenon suggests that, beyond a certain threshold, the inclusion of additional samples may hinder the correct modeling of temporal dependencies without providing relevant kinematic information. In fact, higher sampling frequencies reduce the parsimony of the system by increasing the dimensionality of the input space, which most likely impacts the model's ability to extract discriminative features. Thus, higher-rate data may introduce redundancy that causes overfitting, which may ultimately challenge the network's generalization capacity. In contrast, moderate reductions in sampling rate did not compromise fall detection, indicating that the distinctive dynamics of these events are predominantly concentrated in low frequencies. In fact, several models exhibited stable performance even at 10 Hz, consistent with the findings of Maurer et al. [24], who observed that movement recognition hit rate in a HAR system using with an eWatch improved up to ~15–20 Hz before reaching an accuracy plateau above 20 Hz. Although our CNN achieved its best sensitivity at 10 Hz (slightly higher than at 20 Hz), the difference was minimal, suggesting that even 10 Hz may be viable in certain scenarios. Nevertheless, selecting ~20 Hz provides a safety margin to capture the rapid transients of falls without incurring a significant computational cost, aligning with the trade-off between performance and energy efficiency reported in the literature [25,31,55].

A second critical aspect concerns the performance gap between deep learning architectures and traditional machine learning methods, as well as their practical implications. In our experiments, the CNN-LSTM architecture clearly outperformed SVM and k-NN at 20 Hz (98.89% accuracy versus 93.06% and 95.00%, respectively), reflecting its superior ability to capture the nonlinear and temporal patterns characteristic of falls. This behavior has also been reported in previous studies, such as that of Afuan et al. [56]. These deep-learning approaches typically require balanced datasets and extensive preprocessing. Given these computational demands, a hybrid pipeline was implemented wherein the wearable locally stores 4 s windows upon detecting peaks  $> 2$  g, which are subsequently analyzed externally using the CNN-LSTM model. Operating at 20 Hz, this configuration yielded a false-positive rate below 1.5% in controlled tests and a cumulative error of 5% during seven-day continuous monitoring. This equates to approximately 8 false alarms per day, a rate consistent with most clinical trials [37]. It is worth noting that most false positives originated from abrupt but non-fall activities such as climbing stairs quickly, riding a motorcycle, running, or traveling by taxi, yet the model filtered out the vast majority of these events.

From a practical standpoint, our results indicate that it is feasible to design wearable fall detection systems that are both highly accurate and energy-efficient by limiting the sampling frequency to approximately 20 Hz. To quantify this efficiency, our empirical measurements using an ERASMUS EMT-500 multimeter revealed a ~44% reduction in

sensor current (2.616 mA at 100 Hz vs. 1.459 mA at 20 Hz). Furthermore, the trained model size dropped by ~62% (2705 KB to 1030 KB) and the transmission payload by 80% (15 KB to 3 KB). These combined factors directly translate to a lower computational and transmission load, resulting in a significant extension of the device's battery life [25,57]. At the same time, the false alarm rate remained low (specificity ~95% during daily life testing), which is a critical factor for system acceptance since excessive false alarms can lead to desensitization among caregivers and users [53,58]. In our external validations, the CNN-LSTM model trained with the SisFall dataset demonstrated remarkable generalization capability, achieving about 82% sensitivity in detecting real falls from the FARSEEING repository of acceleration signals captured during actual falls (with ~96% specificity). This performance is consistent with that reported by Bourke et al. [58], who, using a machine learning algorithm, obtained 88% sensitivity and 87% specificity with real data from the same dataset. Similarly, in the FFF repository, which features more homogeneous conditions, the model achieved nearly 98% sensitivity, confirming its robustness when the sensor configuration remains consistent. Overall, this study provides a comprehensive evaluation that unifies and extends previous findings, offering quantitative guidelines to optimize the sampling rate without compromising effectiveness. This is particularly valuable for the development of wearable IoT devices, as it enables more durable and reliable telemonitoring solutions capable of improving the safety and quality of life of vulnerable populations.

## 5. Conclusions

This study provides clear experimental evidence that the sampling frequency strongly shapes the performance and stability of wearable fall-detection models. The obtained results show that a CNN-LSTM architecture trained at 20 Hz achieved the most consistent balance between sensitivity and specificity across synthetic and real-world datasets, including data captured during a seven-day continuous monitoring campaign with a functional prototype. These findings highlight that moderate sampling frequencies can effectively capture the kinematic signatures of falls while maintaining model reliability under everyday conditions. Overall, the results establish a practical reference for developing next-generation wearable systems for fall detection, combining accuracy, robustness, and long-term operational feasibility within IoT-based monitoring environments.

**Author Contributions:** Conceptualization, M.V. and E.C.; Methodology, M.V.; Formal analysis, M.V. and E.C.; Investigation, M.V.; Resources, M.V. and E.C.; Writing—original draft preparation, M.V. and E.C.; Writing—review and editing, M.V. and E.C.; Supervision, E.C.; Project administration, E.C.; Funding acquisition, E.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Spanish Ministry of Science, Innovation, and Universities (MCIN/AEI/10.13039/501100011033) and NextGenerationEU/PRTR Funds under grant ED2021-130456B-I00, by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech (grant B4-2023-12), DIANA PAIDI research group and Universidad de Investigación y Desarrollo (UDI, Bucaramanga, Colombia).

**Institutional Review Board Statement:** The study was conducted in accordance with the Declaration of Helsinki and approved by the Ethics Committee of the Universidad de Investigación y Desarrollo (UDI) (Act No. CEI0104032025, approval date: 4 March 2025).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study are openly available in Zenodo at <https://doi.org/10.5281/zenodo.17776345>.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. World Health Organization. Ageing and Health. Available online: <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health> (accessed on 10 April 2023).
2. World Health Organization. Falls. Available online: <https://www.who.int/en/news-room/fact-sheets/detail/falls> (accessed on 26 February 2023).
3. United Nations. World Population Ageing 2017: Highlights. Available online: <https://www.un-ilibrary.org/content/books/9789213627457> (accessed on 31 August 2025).
4. World Health Organization. WHO Global Report on Falls Prevention in Older Age. Available online: <https://apps.who.int/iris/handle/10665/43811> (accessed on 26 February 2023).
5. Peel, N.M. Epidemiology of Falls in Older Age. *Can. J. Aging* **2011**, *30*, 7–19. [[CrossRef](#)] [[PubMed](#)]
6. Tinetti, M.E.; Liu, W.L.; Claus, E.B. Predictors and Prognosis of Inability to Get Up After Falls Among Elderly Persons. *JAMA* **1993**, *269*, 65–70. [[CrossRef](#)]
7. Wang, X.; Ellul, J.; Azzopardi, G. Elderly Fall Detection Systems: A Literature Survey. *Front. Robot. AI* **2020**, *7*, 71. [[CrossRef](#)] [[PubMed](#)]
8. Qian, Z.; Lin, Y.; Jing, W.; Ma, Z.; Liu, H.; Yin, R.; Li, Z.; Bi, Z.; Zhang, W. Development of a Real-Time Wearable Fall Detection System in the Context of Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 21999–22007. [[CrossRef](#)]
9. Karar, M.E.; Shehata, H.I.; Reyad, O. A Survey of IoT-Based Fall Detection for Aiding Elderly Care: Sensors, Methods, Challenges and Future Trends. *Appl. Sci.* **2022**, *12*, 3276. [[CrossRef](#)]
10. Casamassima, F.; Farella, E.; Benini, L. Context aware power management for motion-sensing body area network nodes. In Proceedings of the 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 24–28 March 2014; pp. 1–6. [[CrossRef](#)]
11. Villa, M.; Casilari, E. Energy-Efficient Fall-Detection System Using LoRa and Hybrid Algorithms. *Biomimetics* **2025**, *10*, 313. [[CrossRef](#)]
12. Salah, O.Z.; Selvaperumal, S.K.; Abdulla, R. Accelerometer-based elderly fall detection system using edge artificial intelligence architecture. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 4430–4438. [[CrossRef](#)]
13. Huynh, Q.T.; Nguyen, U.D.; Tran, B.Q. A Cloud-Based System for In-Home Fall Detection and Activity Assessment. In *Proceedings of the IFMBE Proceedings*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 69, pp. 103–108. [[CrossRef](#)]
14. Chang, W.J.; Chen, L.B.; Chen, M.C.; Su, J.P.; Sie, C.Y.; Yang, C.H. Design and Implementation of an Intelligent Assistive System for Visually Impaired People for Aerial Obstacle Avoidance and Fall Detection. *IEEE Sens. J.* **2020**, *20*, 10199–10210. [[CrossRef](#)]
15. Li, Y.; Lin, Z.; Huang, Z.; Cai, Z.; Huang, L.; Wei, Z. A Channel Hopping LoRa Technology Based Emergency Communication System for Elderly People Living Alone. In Proceedings of the 2022 21st International Symposium on Communications and Information Technologies, ISCIT 2022, Xi'an, China, 27–30 September 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022; pp. 19–26. [[CrossRef](#)]
16. Wu, Y.; Zeng, P.; Ge, H. A Research of Fall Detection Module Based on NB-IOT. In Proceedings of the 2022 7th International Conference on Computer and Communication Systems, ICCCS 2022, Seoul, Republic of Korea, 3–5 November 2022; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022; pp. 197–201. [[CrossRef](#)]
17. Liu, P.; Pan, J.; Zhu, H.; Li, Y. A Wearable Fall Detection System Based on 1D CNN. In Proceedings of the Proceedings—2021 2nd International Conference on Artificial Intelligence and Computer Engineering, ICAICE 2021, Hangzhou, China, 5–7 November 2021; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2021; pp. 200–203. [[CrossRef](#)]
18. Pena Queralta, J.; Gia, T.N.; Tenhunen, H.; Westerlund, T. Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks. In Proceedings of the 2019 42nd International Conference on Telecommunications and Signal Processing, TSP 2019, Budapest, Hungary, 1–3 July 2019; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2019; pp. 601–604. [[CrossRef](#)]
19. Cai, W.Y.; Guo, J.H.; Zhang, M.Y.; Ruan, Z.X.; Zheng, X.C.; Lv, S.S. GBDT-Based Fall Detection with Comprehensive Data from Posture Sensor and Human Skeleton Extraction. *J. Healthc. Eng.* **2020**, *2020*, 8887340. [[CrossRef](#)]
20. Linhares, Í.; Andrade, R.M.C.; Júnior, E.C.; Oliveira, P.A.; Oliveira, B.; Aguilar, P. Lessons Learned from the Development of Mobile Applications for Fall Detection. In Proceedings of the GLOBAL HEALTH 2020: The Ninth International Conference on Global Health Challenges, Nice, France, 25–29 October 2020; pp. 10–25.
21. Yuan, J.; Tan, K.K.; Lee, T.H.; Koh, G.C.H. Power-efficient interrupt-driven algorithms for fall detection and classification of activities of daily living. *IEEE Sens. J.* **2015**, *15*, 1377–1387. [[CrossRef](#)]
22. Xu, T.; Se, H.; Liu, J. A two-step fall detection algorithm combining threshold-based method and convolutional neural network. *Metrol. Meas. Syst.* **2021**, *28*, 23–40. [[CrossRef](#)]
23. Danilenka, A.; Sowiński, P.; Rachwał, K.; Bogacka, K.; Dąbrowska, A.; Kobus, M.; Baszczyński, K.; Okrasa, M.; Olczak, W.; Dymarski, P.; et al. Real-Time AI-Driven Fall Detection Method for Occupational Health and Safety. *Electronics* **2023**, *12*, 4257. [[CrossRef](#)]

24. Maurer, U.; Smailagic, A.; Siewiorek, D.P.; Deisher, M. Activity recognition and monitoring using multiple sensors on different body positions. In Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks, Cambridge, MA, USA, 3–5 April 2006; Volume 2006, pp. 113–116. [[CrossRef](#)]
25. Santoyo-Ramón, J.A.; Casilari, E.; Manuel Cano-García, J. A study of the influence of the sensor sampling frequency on the performance of wearable fall detectors. *Measurement* **2022**, *193*, 110945. [[CrossRef](#)]
26. Nguyen, H.P.; Ayachi, F.; Lavigne-Pelletier, C.; Blamoutier, M.; Rahimi, F.; Boissy, P.; Jog, M.; Duval, C. Auto detection and segmentation of physical activities during a Timed-Up-and-Go (TUG) task in healthy older adults using multiple inertial sensors. *J. Neuroeng. Rehabil.* **2015**, *12*, 36. [[CrossRef](#)]
27. Mathie, M.J.; Coster, A.C.F.; Lovell, N.H.; Celler, B.G. Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiol. Meas.* **2004**, *25*, R1. [[CrossRef](#)] [[PubMed](#)]
28. Winter, D.A. *Biomechanics and Motor Control of Human Movement*, 4th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2009; pp. 1–370. [[CrossRef](#)]
29. Antonsson, E.K.; Mann, R.W. The frequency content of gait. *J. Biomech.* **1985**, *18*, 39–47. [[CrossRef](#)]
30. Liu, K.C.; Hsieh, C.Y.; Hsu, S.J.P.; Chan, C.T. Impact of Sampling Rate on Wearable-Based Fall Detection Systems Based on Machine Learning Models. *IEEE Sens. J.* **2018**, *18*, 9882–9890. [[CrossRef](#)]
31. Sanchez-Iborra, R. LPWAN and Embedded Machine Learning as Enablers for the Next Generation of Wearable Devices. *Sensors* **2021**, *21*, 5218. [[CrossRef](#)]
32. Sucerquia, A.; López, J.D.; Vargas-Bonilla, J.F. SisFall: A fall and movement dataset. *Sensors* **2017**, *17*, 198. [[CrossRef](#)]
33. Zurbuchen, N.; Wilde, A.; Bruegger, P. A Machine Learning Multi-Class Approach for Fall Detection Systems Based on Wearable Sensors with a Study on Sampling Rates Selection. *Sensors* **2021**, *21*, 938. [[CrossRef](#)]
34. SciPy Developers. *scipy.signal.firwin—SciPy v1.16.2 Manual*. Available online: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html> (accessed on 28 November 2025).
35. SciPy Developers. *scipy.signal.resample\_poly—SciPy Reference Guide*. Available online: [https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample\\_poly.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample_poly.html) (accessed on 28 November 2025).
36. Casilari, E.; Lora-Rivera, R.; García-Iagos, F. A Study on the Application of Convolutional Neural Networks to Fall Detection Evaluated with Multiple Public Datasets. *Sensors* **2020**, *20*, 1466. [[CrossRef](#)] [[PubMed](#)]
37. Alizadeh, J.; Bogdan, M.; Classen, J.; Fricke, C. Support Vector Machine Classifiers Show High Generalizability in Automatic Fall Detection in Older Adults. *Sensors* **2021**, *21*, 7166. [[CrossRef](#)]
38. Özdemir, A.T.; Barshan, B. Detecting Falls with Wearable Sensors Using Machine Learning Techniques. *Sensors* **2014**, *14*, 10691–10708. [[CrossRef](#)]
39. Mauldin, T.R.; Canby, M.E.; Metsis, V.; Ngu, A.H.H.; Rivera, C.C. SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning. *Sensors* **2018**, *18*, 3363. [[CrossRef](#)]
40. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [[CrossRef](#)]
41. Wu, J.; Wang, J.; Zhan, A.; Wu, C. Fall Detection with CNN-Casual LSTM Network. *Information* **2021**, *12*, 403. [[CrossRef](#)]
42. Xu, J.; He, Z.; Zhang, Y. CNN-LSTM Combined Network for IoT Enabled Fall Detection Applications. *J. Phys. Conf. Ser.* **2019**, *1267*, 012044. [[CrossRef](#)]
43. Wang, F.T.; Chan, H.L.; Hsu, M.H.; Lin, C.K.; Chao, P.K.; Chang, Y.J. Threshold-based fall detection using a hybrid of tri-axial accelerometer and gyroscope. *Physiol. Meas.* **2018**, *39*, 105002. [[CrossRef](#)]
44. Klenk, J.; Schwickert, L.; Palmerini, L.; Mellone, S.; Bourke, A.; Ihlen, E.A.F.; Kerse, N.; Hauer, K.; Pijnappels, M.; Synofzik, M.; et al. The FARSEEING real-world fall repository: A large-scale collaborative database to collect and share sensor signals from real-world falls. *Eur. Rev. Aging Phys. Act.* **2016**, *13*, 8. [[CrossRef](#)]
45. Mosquera-Lopez, C.; Wan, E.; Shastry, M.; Folsom, J.; Leitschuh, J.; Condon, J.; Rajhbeharrysingh, U.; Hildebrand, A.; Cameron, M.; Jacobs, P.G. Automated Detection of Real-World Falls: Modeled from People with Multiple Sclerosis. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 1975–1984. [[CrossRef](#)] [[PubMed](#)]
46. Arduino Nano 33 BLE Sense Rev2 | Arduino Documentation. Available online: <https://docs.arduino.cc/hardware/nano-33-ble-sense-rev2/> (accessed on 3 September 2024).
47. Li, C.; Liu, M.; Yan, X.; Teng, G. Research on CNN-BiLSTM Fall Detection Algorithm Based on Improved Attention Mechanism. *Appl. Sci.* **2022**, *12*, 9671. [[CrossRef](#)]
48. Galvão, Y.M.; Ferreira, J.; Albuquerque, V.A.; Barros, P.; Fernandes, B.J.T. A multimodal approach using deep learning for fall detection. *Expert. Syst. Appl.* **2021**, *168*, 114226. [[CrossRef](#)]
49. Musci, M.; De Martini, D.; Blago, N.; Facchinetti, T.; Piastra, M. Online Fall Detection Using Recurrent Neural Networks on Smart Wearable Devices. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1276–1289. [[CrossRef](#)]
50. Thakur, N.; Han, C.Y. A Study of Fall Detection in Assisted Living: Identifying and Improving the Optimal Machine Learning Method. *J. Sens. Actuator Netw.* **2021**, *10*, 39. [[CrossRef](#)]



51. González-Cañete, F.J.; Casilari, E. Consumption Analysis of Smartphone based Fall Detection Systems with Multiple External Wireless Sensors. *Sensors* **2020**, *20*, 622. [[CrossRef](#)]
52. Yamane, T.; Kimura, M.; Morita, M. Effects of Sampling Frequency on Human Activity Recognition with Machine Learning Aiming at Clinical Applications. *Sensors* **2025**, *25*, 3780. [[CrossRef](#)]
53. Google LLC. Google Colaboratory. Available online: <https://colab.research.google.com/> (accessed on 28 November 2025).
54. Afuan, L.; Isnanto, R.R. 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 Conf.* **2025**, *605*, 03051. [[CrossRef](#)]
55. Casilari, E.; Silva, C.A. An analytical comparison of datasets of Real-World and simulated falls intended for the evaluation of wearable fall alerting systems. *Measurement* **2022**, *202*, 111843. [[CrossRef](#)]
56. Rezaie, H.; Ghassemian, M. Implementation study of wearable sensors for activity recognition systems. *Healthc. Technol. Lett.* **2015**, *2*, 95–100. [[CrossRef](#)] [[PubMed](#)]
57. Bagalà, F.; Becker, C.; Cappello, A.; Chiari, L.; Aminian, K.; Hausdorff, J.M.; Zijlstra, W.; Klenk, J. Evaluation of Accelerometer-Based Fall Detection Algorithms on Real-World Falls. *PLoS ONE* **2012**, *7*, e37062. [[CrossRef](#)] [[PubMed](#)]
58. Bourke, A.K.; Klenk, J.; Schwickert, L.; Aminian, K.; Ihlen, E.A.F.; Mellone, S.; Helbostad, J.L.; Chiari, L.; Becker, C. Fall detection algorithms for real-world falls harvested from lumbar sensors in the elderly population: A machine learning approach. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2016, Orlando, FL, USA, 16–20 August 2016; pp. 3712–3715. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.