




# Fusion of Machine Learning and Threshold-Based Approaches for Fall Detection in Healthcare Using Inertial Sensors

Ya Wang<sup>1</sup><sup>a</sup>, Peiman Alipour Sarvari<sup>2</sup><sup>b</sup> and Djamel Khadraoui<sup>2</sup><sup>c</sup>

<sup>1</sup>Faculty of Science, Technology and Medicine, University of Luxembourg, Esch sur Alzette, Luxembourg

<sup>2</sup>IT for Innovative Services, Luxembourg University of Science and Technology, Esch sur Alzette, Luxembourg

**Keywords:** Wearable Fall Detection, Feature Extraction, Threshold, Machine Learning, Inertial Sensors.

**Abstract:** In the healthcare sector, specifically for elderly care, accurate and efficient fall detection is crucial. We present an advanced fall detection methodology tailored for wearable systems. Our approach blends threshold-based screening with machine learning models like Support Vector Machine, K-Nearest Neighbors, Decision Tree, Random Forest, and XGBoost. Utilizing 65 features extracted from the gyroscope and accelerometer data from Inertial Measurement Units, our method addresses the class imbalance often found between Activities of Daily Living and actual fall events. Threshold-based pre-screening serves to mitigate the class imbalance of the fall dataset, making the subsequent machine-learning classification more effective. Validation on two open-source IMU datasets, Sisfall and FallAID, achieving high accuracy rates of 99.55%, 99.68% (wrist), 99.76% (waist), and 99.52% (neck), shows our model surpassing existing solutions in detection accuracy. Furthermore, our strategic feature extraction not only enhances the model's performance but also allows for a fourfold reduction by using the 15 most important features in data transmission without sacrificing accuracy. These findings underscore the efficiency and potential of our methodology, indicating that wearables can indeed be powerful tools for high-precision fall detection with minimal data overhead.


## 1 INTRODUCTION


According to the World Health Organization, falls account for approximately 600,000 global deaths each year, ranking second among unintentional injury-related deaths (WHO, 2023). Alarming, 75% of these fatalities occur in adults over the age of 65 (Vaishya and Vaish, 2020). With the global population aging at an unprecedented rate (WHO, 2022), immediate assistance following falls is vital to minimize medical complications. In fact, prolonged periods of immobility after a fall, often lasting over an hour, have been shown to increase the risk of mortality and lead to severe health issues such as dehydration and pneumonia (Fleming and Brayne, 2008).


Given these concerns, there's been a surge in interest in cost-effective Fall Detection Systems (FDSs). Telecare and remote biosignal monitoring offer innovative pathways for these systems. Since 2010, both research articles and patents in automatic FDSs

have seen a significant uptick, underscoring the field's growing importance (Tanwar et al., 2022).

Fall detection systems (FDSs) primarily fall into two categories: Context-Aware Systems (CAS) and wearable FDSs. CAS systems utilize sensors like microphones, cameras, and radars placed in a predefined area surrounding the individual. However, the need for customization, high installation and maintenance costs, and limited coverage areas restrict their applicability outside controlled environments such as nursing homes. Contrarily, wearable FDSs use inertial measurement units (IMUs) directly attached to the individual, allowing for location-independent fall detection. These systems offer numerous advantages including cost-efficiency, easier installation, privacy preservation, and simpler design and configuration (Hashim et al., 2020). This feature, combined with the widespread use of smartwatches and sports bands, enhances the feasibility and accessibility of wearable FDSs, making them particularly suitable for urban areas with reliable mobile connectivity."

<sup>a</sup> <https://orcid.org/0000-0002-4542-1074>

<sup>b</sup> <https://orcid.org/0000-0003-1235-2102>

<sup>c</sup> <https://orcid.org/0000-0003-1054-1612>

## 2 RELATED WORK

Algorithms of Wearable FDS can be broadly classified into three types: (i) threshold-based approach (ii) machine learning-based approach and (iii) deep learning-based approach. Table 1 summarizes representative studies of the above three categories of wearable sensor-based fall detection.

**Threshold-Based Approach.** Threshold-based methods hinge on predefined threshold values to detect falls by comparing specific sensor data—namely, acceleration (Saadeh et al., 2019), angular velocity (Bourke and Lyons, 2008), and body angle (Sorvala et al., 2012). It operates on the premise that falls exhibit distinct differences in body position and velocity compared to activities of daily living (ADLs). The system's effectiveness largely depends on the accuracy of these preset values. For example, de Sousa et al. (de Sousa et al., 2021) introduced a low-power wearable system for fall detection that utilized a threshold-based approach and achieved a sensitivity of 92.6% and specificity of 97.7%. Jung et al. (Jung et al., 2020) employed thresholds based on the sum vector magnitude of acceleration, the sum vector magnitude of angular velocity, and the vertical angle. They reported a sensitivity of 100% and specificity of 97.54%. However, the approach has limitations in terms of generalizing across different settings and populations, causing decreased performance, particularly in specificity, when tested on complex datasets like SisFall (Wang et al., 2020)(Sucerquia et al., 2017).

**Machine Learning-Based Approach.** Machine learning techniques offer a flexible and adaptive alternative to threshold-based fall detection methods, often yielding improved performance in diverse scenarios. For example, Giuffrida et al. used a Support Vector Machine (SVM) model trained on a curated set of features, which significantly optimized the system's parameters (Giuffrida et al., 2019). In a similar vein, Yu et al. adopted a Hidden Markov Model (HMM) for fall detection that circumvented the need for manual feature selection altogether. Their approach processed raw acceleration data and achieved an impressive sensitivity of 99.2% and specificity of 99.0% (Yu et al., 2017). To assess the general efficacy of machine learning in this domain, Martinez-Villaseor et al. compared four key machine learning classifiers: Random Forest (RF), SVM, Multilayer Perceptron (MLP), and k-Nearest Neighbors (KNN). These algorithms were evaluated for their ability to differentiate falls from

fall-like activities, adding a layer of complexity to the detection problem (Martinez-Villaseñor and Ponce, 2020). Despite their promising results, machine learning-based approaches do face a bottleneck in feature selection. The process of identifying the most relevant features for fall detection is non-trivial and can affect the algorithm's overall performance.

**Deep Learning-Based Approach.** The advent of powerful computational hardware has propelled the utilization of deep learning algorithms in fall detection (Yu et al., 2020). These algorithms automatically identify important features from raw sensor data, eliminating the need for manual feature engineering. Remarkable performance metrics, such as a sensitivity of 99.3% and specificity of 91.86% using ResNet architecture, have been reported (Zhang et al., 2021). Moreover, the ConvLSTM model was shown to excel in both sensitivity and specificity, achieving 99.32% and 99.01% (Yu et al., 2022).

To address this challenge of fall detection and enhance computational efficiency, we introduce a two-tiered hybrid algorithm that integrates threshold-based and machine-learning methods for wearable Fall detection systems.

## 3 METHODOLOGY

Fall datasets often exhibit a class imbalance between Activities of Daily Living (ADL) and genuine fall events. This imbalance skews the performance of models trained on such datasets, particularly affecting their ability to accurately identify falls, which are the minority class. To address this challenge and to enhance computational efficiency, we introduce a two-tiered hybrid algorithm that integrates threshold-based and machine-learning methods. The flow chart in Figure 1 illustrates the overall process of the fusion approach.

### 3.1 Data Processing

#### 3.1.1 Dataset

In this study, the open source datasets, SisFall and FallAIIID were utilized to validate the effectiveness of three different fall detection approaches, after analyzing the various fall detection datasets.

**SisFall Dataset.** SisFall (Sucerquia et al., 2017) collected data from the IMU attached to the waist. The IMU includes sensors such as accelerometers,

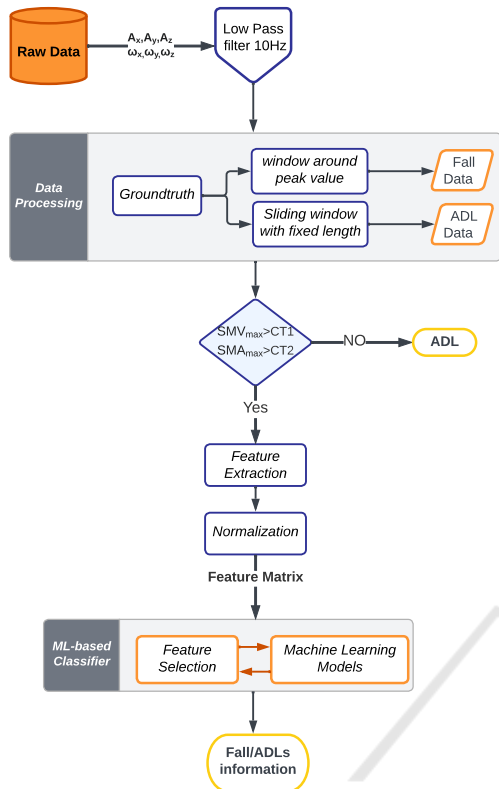


Figure 1: Flow chart of the designed fall detection approach.

gyroscopes, and magnetometers. It consists of 19 ADLs and 15 fall types performed by 23 young adults, 15 ADL types performed by 14 healthy and independent participants over 62 years old, and data from one participant of 60 years old that performed all ADLs and falls. The dataset includes motion data captured by an inertial measurement unit (IMU) placed on the waist at a sampling rate of 200Hz. The SisFall dataset provides a substantial number of fall and ADL trials, making it suitable for evaluating fall detection algorithms.

**FallAIIID Dataset.** The FallAIIID dataset, proposed by (Saleh et al., 2020), utilizes an IMU placed on the neck, chest, and waist to measure movement during experiments. The dataset was obtained from 15 subjects (eight males and seven females), defined as containing 35 falls and 44 ADL types. The waist and neck acquired ADL and fall data for 14 and 12 subjects, respectively, whereas the wrist sensor acquired ADL and fall data for 13 and 9 subjects.

Figure 2 illustrates the coordinates of the accelerometer and the angular velocity measurements in our research.

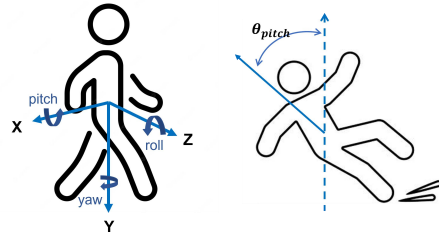


Figure 2: Body accelerator and angular velocity measurement system.

### 3.1.2 Low-Pass Filtering

The raw data collected from the Inertial Measurement Unit (IMU) may contain electronic noise or other types of artifacts that can affect the accuracy and reliability of the measurements. To mitigate these disturbances, a filtering process is often applied to the data (Yu et al., 2022)(Jung et al., 2020) (Shi et al., 2020). In this research, we utilized a fourth-order low-pass Butterworth filter with a 2-pass digital implementation to remove noise and artifacts from the accelerometer data and angular velocity data. The chosen cut-off frequency for the filter was set at 10Hz. This value was selected because the relevant frequency spectra of human motion typically fall within the range of 0 to 10Hz (Winter, 2009).

### 3.1.3 Data Refinement and Segmentation

The six-axis inertial signals from the IMU sensor were defined as  $A_x, A_y, A_z, \omega_x, \omega_y$  and  $\omega_z$ . During a fall event, as the body makes contact with the ground, it typically causes sudden and pronounced peaks in acceleration. Figure 3 illustrates a representative diagram displaying the 3-axis acceleration and angular velocity data of a forward fall event starting from the initial state position. Such peaks, captured by the Signal Magnitude Vector (SMV), serve as the primary indicators for fall events. Mathematically, for each measurement during a particular fall instance, SMV is defined as:

$$SMV_i = \sqrt{A_{x_i}^2 + A_{y_i}^2 + A_{z_i}^2} \quad (1)$$

#### Segmentation Strategy Using Signal Magnitude Vector (SMV):

1. **Fall Events:** Fall instances are pinpointed by observing the 2-second window surrounding the peak SMV values within the recorded data. Ground truth information is used to validate and label these samples as falls. The definition of the time of peak SMV ( $t_{SMV_{max}}^i$ ) for the  $i$ th fall event is defined as:

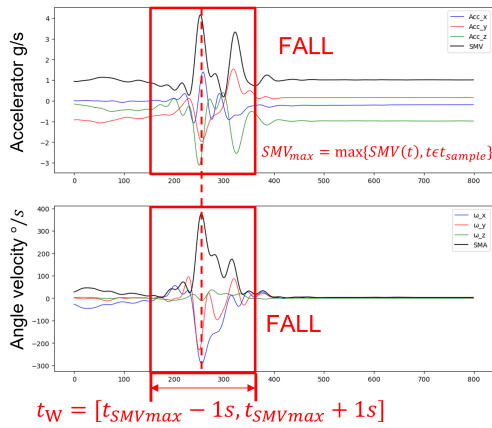


Figure 3: Fall accident from a static posture. ( $t_W$ : length of the time window for the fall segment).

$$t_{SMV_{max}}^i = \arg \max_t \{SMV_i(t) : SMV_i(t) \in \text{ith fall}\} \quad (2)$$

- ADL Events:** To diversify the ADLs dataset, a sliding window method is employed. This mechanism uses a consistent window of 2 seconds, progressing at steps of 0.2 seconds across the data timestamps. This method captures data fragments and categorizes them as non-fall events, ensuring a more encompassing representation of typical movements.

However, there are instances where high-intensity activities, such as rapid walking or jumping, can produce acceleration patterns similar to falls, posing challenges in accurate detection. To address this conundrum, gyroscope data is incorporated to assess and determine the subject's posture. For every data point during the  $i$ -th fall event, the Signal Magnitude Vector of Angular Velocity (SMA) is leveraged to measure alterations in angular velocity. This metric is articulated as:

$$SMA_i = \sqrt{\omega_{x_i}^2 + \omega_{y_i}^2 + \omega_{z_i}^2} \quad (3)$$

### 3.1.4 Adaptive Time-Window Size

Fall detection algorithms commonly utilize temporal windows to analyze inertial signals where a fall event might occur. These windows typically span between 0.2 and 2 seconds. Optimal window durations for fall and human activity detection have been subject to research. Banos et al. recommended a 1-2 second window for general human activities, balancing recognition speed with accuracy (Banos et al., 2014). The intrinsic dynamics of a fall, characterized by abrupt and unexpected movements, usually occur within a 1-3 second timeframe (Yu, 2008). The most critical

phases of a fall, including the free-fall and impact periods, happen within an even narrower range of 0.5-0.85 seconds (Huynh et al., 2013). Eduardo Casilari et al. fine-tuned this by proposing a 2-second window, centered around the peak of the fall signal, capturing the most relevant features of a fall event (Casilari and Silva, 2022). In our analysis, we adopt this 2-second observation window for optimal fall detection.

### 3.1.5 Data Split

After the data processing, the whole data set is split into the training dataset and test dataset with a ratio of 75/25. The flow chart in Figure 4 illustrates the training process and testing process.

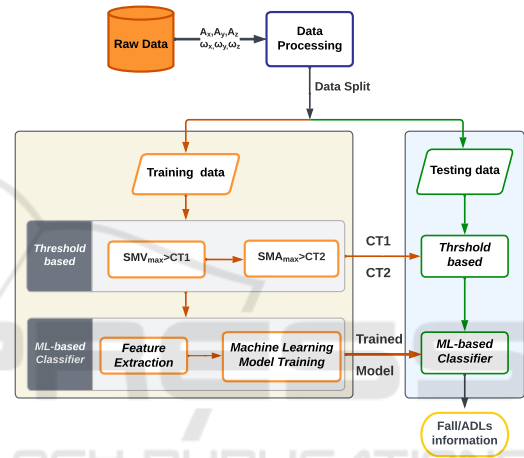


Figure 4: Flow chart of the train process and test process.

## 3.2 Threshold-Based Methods

The first tier acts as an initial filter using threshold-based criteria to swiftly differentiate potential fall events from routine activities. This approach allows for rapid processing, screening out most ADL instances and forwarding only suspected fall events to the second tier for detailed analysis. Specifically, the algorithm employs two thresholds calculated from sensor measurements of the training data: Signal Magnitude Vector (SMV) and Signal Magnitude Vector of Angular Velocity (SMA). These thresholds are defined as follows:

$$CT1 = \min\{SMV_{max} : SMV_{max} \in \text{fall training data}\} \quad (4)$$

where  $SMV_{max} = \max\{SMV(t) : t \in t_W\}$

$$CT2 = \min\{SMA_{max} : SMA_{max} \in \text{fall training data}\} \quad (5)$$

where  $SMA_{max} = \max\{SMA(t) : t \in t_W\}$

Within the predefined time window  $t_W$ , if both  $SMV_{max}$  and  $SMA_{max}$  exceed their respective thresholds, the data is forwarded to the second tier; otherwise, it is disregarded.

### 3.3 Machine Learning-Based Approach

The second tier capitalizes on machine learning algorithms to meticulously classify the dataset that's been pre-screened by the initial tier. This dual-stage approach amplifies the algorithm's precision in detecting falls, while also boosting computational speed.

#### 3.3.1 Feature Extraction

Before applying machine learning classifiers, we focus on feature extraction to accurately represent the underlying data patterns. We compute a set of eight statistical features that encapsulate information from accelerometer and gyroscope readings. These features include metrics such as angular velocity, Signal Magnitude Vector (SMV), and Signal Magnitude Vector of Angular Velocity (SMA). For a detailed summary, refer to Table 1. These statistical features are universally accepted in the domains of Human Activity Recognition (HAR) and Fall Detection Systems (FDS) (Sucerquia et al., 2017; Giuffrida et al., 2019; Martinez-Villaseñor and Ponce, 2020; Casilari and Silva, 2022).

We denote the human acclivity feature derived from the raw data of the IMU sensor by  $S$ , which is defined as

$$S = [A_x, A_y, A_z, \omega_x, \omega_y, \omega_z, SMV, SMA, N_{Ang}], \quad (6)$$

where  $A_x, A_y, A_z$  represent the accelerometer readings along the X, Y, and Z axes, respectively, and  $\omega_x, \omega_y, \omega_z$  represent the angular velocity readings along the X, Y, and Z axes, respectively,  $SMV$  and  $SMA$  denote the Signal Magnitude Vector and Signal Magnitude Vector of Angular Velocity, respectively,  $N_{Ang}$  represent the attitude change during the fall.

The attitude angle change during the fall  $N_{Ang}$  is defined as follows:

**Pitch Angle.** The pitch angle represents the forward angle of the sensor during a fall.

$$Pitch = \arccos \frac{|A_z|}{\sqrt{A_z^2 + A_y^2}} \quad (7)$$

**Roll Angle.** The roll angle represents the sideward angle of the sensor during a fall. By incorporating these pitch and roll angles, we can capture the

changes in the forward and sideward attitudes of individuals during a fall.

$$Roll = \arccos \frac{|A_x|}{\sqrt{A_x^2 + A_y^2}} \quad (8)$$

The attitude angle change during the fall is defined as:

$$N_{Ang} = Pitch + Roll \quad (9)$$

The selected statistic features are analytically defined as follows:

**Maximum (Peak) of Feature  $S$ .** This feature represents the peak or maximum value of a specific data feature ( $S$ ) during the fall window. It serves as a meaningful descriptor of the force of the impact against the ground. Mathematically, it can be defined as:

$$S_{max} = \max\{S(t) : t \in t_W\} \quad (10)$$

**Minimum of Feature  $S$ .** This feature indicates the minimum value achieved by the data feature ( $S$ ) during the fall window. It is a key element in describing the fall.

$$S_{min} = \min\{S(t) : t \in t_W\} \quad (11)$$

**Mean of Feature  $S$ .** The mean provides information about the average body motion intensity during the fall. It is computed as the average of the feature values over the observation window ( $t_W$ ) containing  $N_W$  feature samples.

$$\mu_S = \frac{1}{N_W} \sum_{t \in t_W} S \quad (12)$$

where  $N_W$  is defined as:

$$N_w = 2\left[\frac{T}{2} f_s\right] + 1 \quad (13)$$

**Standard Deviation of Feature  $S$ .** This feature describes the variability of the feature ( $S$ ) during the observation window. It is calculated as the square root of the average squared deviation from the mean.

$$\sigma_S = \sqrt{\frac{1}{N_W} \sum_{t \in t_W} (S - \mu_S)^2} \quad (14)$$

**Skewness of Feature  $S$ .** Skewness characterizes the symmetry of the distribution of feature values. It indicates whether the distribution is skewed to the left or right.

$$\gamma_S = \frac{1}{\sigma_S^3 N_W} \sum_{t \in t_W} (S - \mu_S)^3 \quad (15)$$



Table 1: Descriptive statistics of 65 features.

Statistic	Equation	$Acc$	$Ang$	$SMV$	$SMA$	$N_{Ang}$
Maximum	$S_{max} = \max\{S(t) : t \in t_W\}$	$A_{xmax}$ $A_{ymax}$ $A_{zmax}$	$\omega_{xmax}$ $\omega_{ymax}$ $\omega_{zmax}$	$SMV_{max}$	$SMA_{max}$	$N_{Ang_{max}}$
Minimum	$S_{min} = \min\{S(t) : t \in t_W\}$	$A_{xmin}$ $A_{ymin}$ $A_{zmin}$	$\omega_{xmin}$ $\omega_{ymin}$ $\omega_{zmin}$	$SMV_{min}$	$SMA_{min}$	$N_{Ang_{min}}$
Mean	$\mu_S = \frac{1}{N_W} \sum_{t \in t_W} S$	$\mu_{A_x}$ $\mu_{A_y}$ $\mu_{A_z}$	$\mu_{\omega_x}$ $\mu_{\omega_y}$ $\mu_{\omega_z}$	$\mu_{SMV}$	$\mu_{SMA}$	$\mu_{N_{Ang}}$
Standard Deviation	$\sigma_S = \sqrt{\frac{1}{N_W} \sum_{t \in t_W} (S - \mu_S)^2}$	$\sigma_{A_x}$ $\sigma_{A_y}$ $\sigma_{A_z}$	$\sigma_{\omega_x}$ $\sigma_{\omega_y}$ $\sigma_{\omega_z}$	$\sigma_{SMV}$	$\sigma_{SMA}$	$\sigma_{N_{Ang}}$
Skewness	$\gamma_S = \frac{1}{\sigma_S^3 N_W} \sum_{t \in t_W} (S - \mu_S)^3$	$\gamma_{A_x}$ $\gamma_{A_y}$ $\gamma_{A_z}$	$\gamma_{\omega_x}$ $\gamma_{\omega_y}$ $\gamma_{\omega_z}$	$\gamma_{SMV}$	$\gamma_{SMA}$	$\gamma_{N_{Ang}}$
Kurtosis	$Kurt_S = \mathbb{E}[(\frac{S - \mu_S}{\sigma_S})^4]$	$Kurt_{A_x}$ $Kurt_{A_y}$ $Kurt_{A_z}$	$Kurt_{\omega_x}$ $Kurt_{\omega_y}$ $Kurt_{\omega_z}$	$Kurt_{SMV}$	$Kurt_{SMA}$	$Kurt_{N_{Ang}}$
valley-to-peak range	$R_S = S_{max} - S_{min}$	$R_{A_x}$ $R_{A_y}$ $R_{A_z}$	$R_{\omega_x}$ $R_{\omega_y}$ $R_{\omega_z}$	$R_{SMV}$	$R_{SMA}$	NAN
valley-to-peak time	$T_S = t_{Smax} - t_{Smin}$	NAN	NAN	$T_{SMV}$	$T_{SMA}$	NAN

<sup>1</sup>  $A_x, A_y, A_z$  represent the accelerometer readings along the X, Y, and Z axes, respectively,

<sup>2</sup>  $\omega_x, \omega_y, \omega_z$  represent the angular velocity readings along the X, Y, and Z axes,

<sup>3</sup> SMV and SMA denote the Signal Magnitude Vector and Signal Magnitude Vector of Angular Velocity

**Kurtosis of Feature  $S$ .** Kurtosis measures the tailedness (frequency of outliers) of the distribution of feature values.

$$Kurt_S = \mathbb{E}[(\frac{S - \mu_S}{\sigma_S})^4] \quad (16)$$

**Valley-to-Peak Range ( $R_S$ ).** This feature represents the value of the interval between the minimum ( $S_{min}$ ) and maximum ( $S_{max}$ ) of the feature  $S$ . It is calculated as:

$$R_S = S_{max} - S_{min} \quad (17)$$

**Valley-to-Peak Time ( $T_S$ ).** This feature indicates the duration of the interval between the minimum ( $S_{min}$ ) and maximum ( $S_{max}$ ) of the feature  $S$ .

$$T_S = t_{Smax} - t_{Smin} \quad (18)$$

### 3.3.2 Model Implementation

With the features extracted, we proceed to apply machine learning models for classification. In this study, we explore multiple classifiers including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and XGBoost. These classifiers are trained and evaluated

based on their ability to accurately distinguish between fall and non-fall events.

Our dual-layer approach, incorporating these selected features, is rigorously tested on the Sisfall and FallAID datasets. By applying the chosen statistical features to both categories, we aim to critically assess and validate the performance of our machine learning-based fall detection model.

## 4 RESULTS AND DISCUSSION

### 4.1 Performance Evaluation Metrics

Various methods have been developed to evaluate the performance of different classifiers. These methods rely on the outcomes obtained from the classifiers, which are represented in the form of a confusion matrix (Figure 10). The confusion matrix provides a visual representation of the classifier's performance, including true positives, true negatives, false positives, and false negatives.

- True positive (TP): The ADL events have been correctly classified.
- True negative (TN): The fall events have been correctly detected.

- False positive (FP): Fall events that have not been detected.
- False negative (FN): A false alarm situation occurs.

One commonly used method to assess classifier performance is accuracy, which calculates the proportion of correctly classified samples overall. However, accuracy has certain limitations, such as being susceptible to the influence of large abnormal data and potentially misleading results in class-imbalanced training data scenarios. To address these limitations, alternative evaluation methods are selected to evaluate class-imbalanced classifiers effectively. For fall detection tasks with imbalanced classes, sensitivity, specificity, F-score, and receiver operating characteristic (ROC) are commonly utilized to assess the classifier's ability to differentiate falls from a large number of activities of daily living (ADL) events.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Sensitivity measures the proportion of correctly identified positive samples. Specificity, on the other hand, measures the proportion of correctly identified negative samples.

$$Sensitivity = \frac{TP}{TP + FN} \quad (20)$$

$$Specificity = \frac{TN}{TN + FP} \quad (21)$$

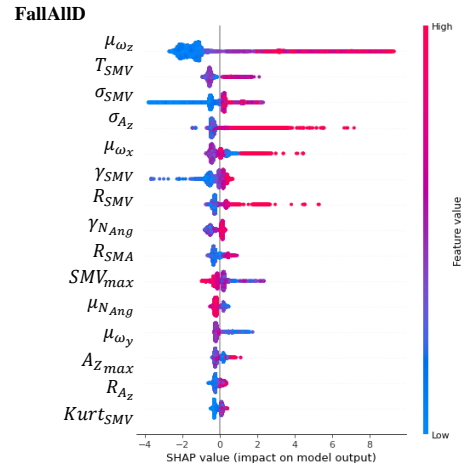
The F1 is a robust evaluation metric that balances sensitivity and specificity. In large-scale datasets, sensitivity and specificity often have a trade-off relationship, and the F-score provides a comprehensive measure of their discrimination. The parameter  $\beta$  in the F-score equation allows adjusting the weight between sensitivity and specificity. Setting  $\beta$  to 0.5 assigns a higher weight to specificity, which is crucial in fall detection as it reflects the detection of all fall signals in the data.

$$F1 = (1 + \beta^2) \frac{Sensitivity \times Specificity}{\beta^2 \times (Sensitivity + Specificity)} \quad (22)$$

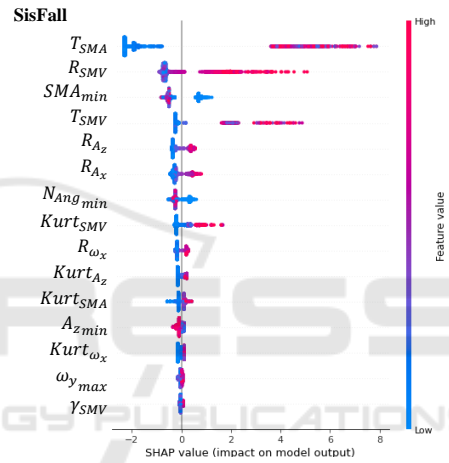
In summary, sensitivity, specificity, and F-score provide comprehensive measures for evaluating the performance of fall detection classifiers, particularly in scenarios with imbalanced class distributions. These metrics address the limitations of accuracy and offer a more nuanced assessment of the classifier's ability to distinguish falls from ADL events.

## 4.2 Results and Discussion

Our analyses underscored the varying importance of features across the two datasets—Sisfall and FallAIID. Notably, features like  $T_{SMV}$ ,  $T_{SMA}$ , and  $R_{SMV}$



(a) Shap values for FallAIID.



(b) Shap values for SisFall.

Figure 5: The Shap values of the top 15 features of the dataset (a) FallAIID; (b) SisFall.

consistently ranked among the top ten most important features when both datasets were integrated, emphasizing their critical role in accurately detecting falls and Activities of Daily Living (ADLs). These findings are corroborated by Table 2, which offers an exhaustive evaluation of different machine learning classifiers based on the feature sets. In which, feature (1) utilizes all 65 extracted features and features (2) focuses on the 15 most important features as determined by feature integration. The model's robustness was evident from its high accuracy rates across varying types of falls and ADLs. The results show that the model is robust to the input of different types of falls/ADLs and achieved superior performance. The FallAIID dataset is collated from 3 different locations with limited data size, it has been shown that the position of the sensors also plays an important role in the fall detection models. And we can also see that

Table 2: Fall detection results for Sisfall and FallAIID: the unit is %, feature (1) utilizes all 65 extracted features, and feature (2) focuses on the 15 most important features.

Dataset	Model	Sensitivity	Specificity	Accuracy	F1
SisFall <sup>1</sup>	SVM	99.47	99.35	99.45	98.22
	KNN	98.06	98.91	98.96	94.27
	DT	98.88	99.38	99.28	98.45
	RF	99.44	98.75	99.54	99.03
	XGB	99.62	98.81	99.55	99.14
SisFall <sup>2</sup>	SVM	98.67	99.03	98.62	96.43
	KNN	97.61	98.63	98.05	86.44
	DT	97.29	99.23	98.70	97.98
	RF	99.01	97.87	99.26	98.88
	XGB	98.79	98.33	99.16	98.92
FallAIID <sup>1</sup> (wrist)	SVM	99.36	1	99.84	99.68
	KNN	98.71	99.58	99.37	98.71
	DT	98.87	99.61	99.55	98.82
	RF	98.89	99.88	98.95	99.24
	XGB	99.62	99.47	99.68	99.47
FallAIID <sup>2</sup> (Wrist)	SVM	98.27	99.40	99.37	97.55
	KNN	97.67	97.44	97.49	94.02
	DT	98.88	99.38	99.18	98.9
	RF	96.44	98.75	99.66	99.4
	XGB	98.62	98.51	99.55	98.7
FallAIID <sup>1</sup> (waist)	SVM	99.77	99.03	99.34	99.27
	KNN	98.06	98.91	98.96	98.69
	DT	97.88	99.02	99.18	99.22
	RF	99.34	98.75	99.76	99.07
	XGB	99.32	98.31	99.35	99.23
FallAIID <sup>2</sup> (waist)	SVM	98.63	97.51	96.74	98.58
	KNN	96.85	95.81	97.39	97.41
	DT	96.86	99.84	97.68	97.88
	RF	98.62	98.80	99.44	98.47
	XGB	97.82	98.82	98.84	98.64
FallAIID <sup>1</sup> (neck)	SVM	96.53	98.94	99.15	91.25
	KNN	89.06	98.91	95.96	87.10
	DT	98.28	99.38	99.12	98.32
	RF	98.24	99.47	99.52	99.40
	XGB	98.88	99.46	99.47	99.51
FallAIID <sup>2</sup> (neck)	SVM	95.81	99.03	99.35	89.55
	KNN	79.66	91.34	86.90	86.79
	DT	97.48	99.38	98.18	97.44
	RF	97.44	98.75	98.66	97.96
	XGB	98.42	98.51	98.55	98.73

the tree-based models show superior and robust performance in different datasets.

In Table 3, we present a comparative analysis of fall detection capabilities between our proposed model and existing models, all evaluated using the same dataset. Our model’s robustness and enhanced performance are evident; it consistently identifies a range of falls and Activities of Daily Living (ADLs) with remarkable accuracy.

## 5 CONCLUSIONS

In this study, we proposed a wearable fall detection model that combines the threshold and ma-

Table 3: Comparison of results between the proposed and previous research models, the unit is %.

	Algorithm	Sensitivity	Specificity	Accuracy
Yu et al. (2020)	ResNet10	97.91	72.89	96.22
	SMOTE	99.17	89.98	97.54
	Sisfall	99.33	91.86	97.52
Santoyo et al. (2022)	CNN	85.97	96.79	NAN
	FallAIID waist			
Jeong et al. (2023)	LightGBM	91.04	96.38	94.86
Proposed Sisfall	Fusion method	99.62	98.81	99.55
Proposed FallAIID wrist	Fusion method	99.62	99.47	99.68
Proposed FallAIID waist	Fusion method	99.34	98.75	99.76
Proposed FallAIID neck	Fusion method	98.24	99.47	99.52

Note: In this comparison, while the same open-source datasets are used, the training and testing datasets for the fall detection model differ due to variations in data processing methods, like filter frequency and sample window size. Hence, the results should be viewed as indicative rather than conclusive.

chine learning approach benchmarked against Sisfall and FallAIID datasets. Employing a suite of 65 rigorously selected statistical features (as shown in table1) extracted from inertial sensors, the study leveraged tree-based ensemble models to achieve unprecedented accuracy rates: 99.55%, 99.68% (wrist), 99.76% (waist), and 99.52% (neck) across the examined datasets. This level of performance substantially outperforms existing benchmarks documented in the scholarly literature.

SHAP value analysis was instrumental in distilling the feature set down to the top 15 most influential features. Comparative analysis indicated that the reduced feature set incurred a statistically insignificant diminution in performance metrics—less than a 1% deviation relative to the exhaustive feature set. The hybrid model architecture, ingeniously combining threshold-based and machine learning algorithms, facilitates minimal data transference from the wearable device to the computational node while sustaining high fidelity in fall detection outcomes.

While the current study’s accomplishments are manifold, it is imperative to acknowledge its limitations. The absence of real fall data in the utilized datasets denotes an opportunity for future work to further validate the model’s performance under invalid conditions. In light of the latter, future research endeavors will be directed toward the integration of this validated model architecture into wearable technology platforms, emphasizing the necessity of feature selection optimization for real-time fall detection.



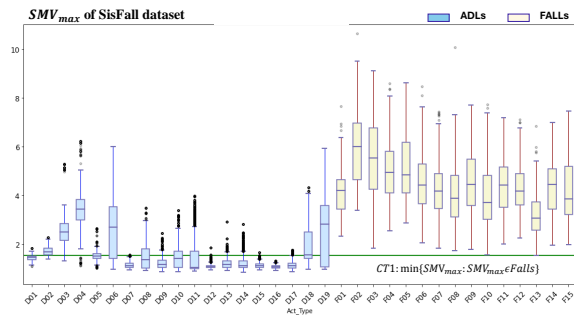
## ACKNOWLEDGEMENTS

This work was adapted and extended from the master's thesis titled 'Analysis and Comparison of Different Types of Algorithms for Fall Detection in Fall Alerting Systems' completed at the University of Luxembourg, supported by the European Active and Assisted Living 2021(AAL) Programme, the Luxembourg National Research Fund (FNR), and the Luxembourg Institute of Science and Technology (LIST). This research is part of the AGAPE project, with the code AAL-2021-8-124-CP, and titled 'ADVANCING INCLUSIVE HEALTH & CARE SOLUTIONS FOR AGEING WELL IN THE NEW DECADE.

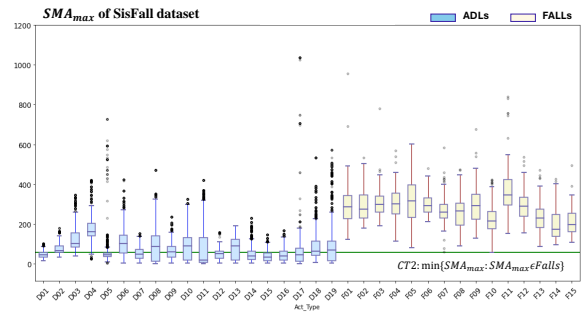
## REFERENCES

- Banos, O., Galvez, J.-M., Damas, M., Pomares, H., and Rojas, I. (2014). Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499.
- Bourke, A. K. and Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. In *Medical engineering & physics*, volume 30, pages 84–90. Elsevier.
- Casilari, E. and Silva, C. A. (2022). An analytical comparison of datasets of real-world and simulated falls intended for the evaluation of wearable fall alerting systems. *Measurement*, 202:111843.
- de Sousa, F. A. S. F., Escriba, C., Bravo, E. G. A., Brossa, V., Fourniols, J., and Rossi, C. (2021). Wearable pre-impact fall detection system based on 3d accelerometer and subject's height. In *IEEE Sensors Journal*, volume 22, pages 1738–1745. IEEE.
- Fleming, J. and Brayne, C. (2008). Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. In *Bmj*, volume 337. British Medical Journal Publishing Group.
- Giuffrida, D., Benetti, G., Martini, D. D., and Facchinetti, T. (2019). Fall detection with supervised machine learning using wearable sensors. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 253–259. IEEE.
- Hashim, H. A., Mohammed, S. L., and Gharghan, S. K. (2020). Accurate fall detection for patients with parkinson's disease based on a data event algorithm and wireless sensor nodes. In *Measurement*, volume 156, page 107573. Elsevier.
- Huynh, Q. T., Nguyen, U. D., Tran, S. V., Nabili, A., and Tran, B. Q. (2013). Fall detection system using combination accelerometer and gyroscope. In *Proc. of the Second Int. I Conf. on Advances in Electronic Devices and Circuits (EDC 2013)*.
- Jung, H., Koo, B., Kim, J., Kim, T., Nam, Y., and Kim, Y. (2020). Enhanced algorithm for the detection of preimpact fall for wearable airbags. 20(5):1277.
- Martinez-Villaseñor, L. and Ponce, H. (2020). Design and analysis for fall detection system simplification. *JoVE (Journal of Visualized Experiments)*, (158):e60361.
- Saadeh, W., Butt, S. A., and Altaf, M. A. B. (2019). A patient-specific single sensor iot-based wearable fall prediction and detection system. In *IEEE transactions on neural systems and rehabilitation engineering*, volume 27, pages 995–1003. IEEE.
- Saleh, M., Abbas, M., and Le Jeannes, R. B. (2020). Fallall: An open dataset of human falls and activities of daily living for classical and deep learning applications. *IEEE Sensors Journal*, 21(2):1849–1858.
- Shi, J., Chen, D., and Wang, M. (2020). Pre-impact fall detection with cnn-based class activation mapping method. *Sensors*, 20(17):4750.
- Sorvala, A., Alasaarela, E., Sorvoja, H., and Myllylä, R. (2012). A two-threshold fall detection algorithm for reducing false alarms. In *2012 6th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–4. IEEE.
- Sucerquia, A., López, J. D., and Vargas-Bonilla, J. F. (2017). Sisfall: A fall and movement dataset. 17(1):198.
- Tanwar, R., Nandal, N., Zamani, M., and Manaf, A. A. (2022). Pathway of trends and technologies in fall detection: a systematic review. In *Healthcare*, volume 10, page 172. Multidisciplinary Digital Publishing Institute.
- Vaishya, R. and Vaish, A. (2020). Falls in older adults are serious. In *Indian journal of orthopaedics*, volume 54, pages 69–74. Springer.
- Wang, X., Ellul, J., and Azzopardi, G. (2020). Elderly fall detection systems: A literature survey. 7:71.
- WHO (2022). *Ageing and health*. WHO, N/A, n/a edition.
- WHO (2023). *World health statistics 2023: monitoring health for the SDGs, sustainable development goals*. WHO, N/A, n/a edition.
- Winter, D. A. (2009). *Biomechanics and motor control of human movement*. John Wiley & Sons.
- Yu, S., Chen, H., and Brown, R. A. (2017). Hidden markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring. *IEEE Journal of Biomedical and Health Informatics*, 22(6):1847–1853.
- Yu, X. (2008). Approaches and principles of fall detection for elderly and patient. In *HealthCom 2008-10th International Conference on e-health Networking, Applications and Services*, pages 42–47. IEEE.
- Yu, X., Koo, B., Jang, J., Kim, Y., and Xiong, S. (2022). A comprehensive comparison of accuracy and practicality of different types of algorithms for pre-impact fall detection using both young and old adults. *Measurement*, 201:111785.
- Yu, X., Qiu, H., and Xiong, S. (2020). A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors. *Frontiers in Bioengineering and Biotechnology*, 8:63.
- Zhang, J., Li, J., and Wang, W. (2021). A class-imbalanced deep learning fall detection algorithm using wearable sensors. *Sensors*, 21(19):6511.

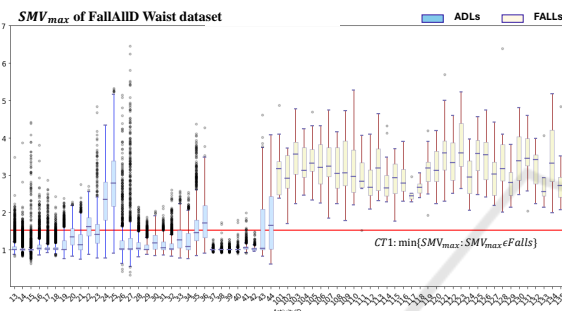
## APPENDIX



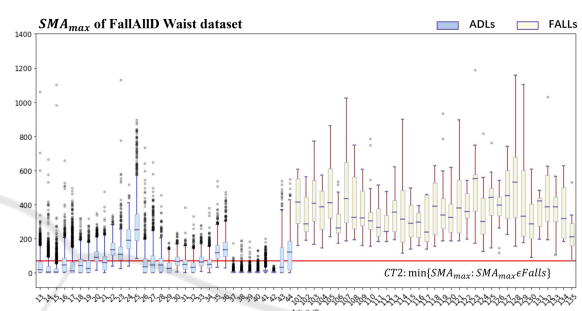
(a) Boxplot of  $SMV_{max}$  for SisFall dataset



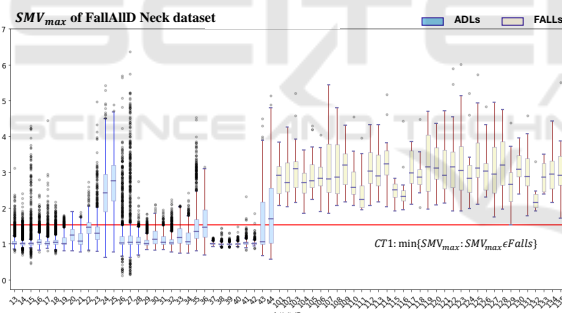
(b) Boxplot of  $SMA_{max}$  for SisFall dataset



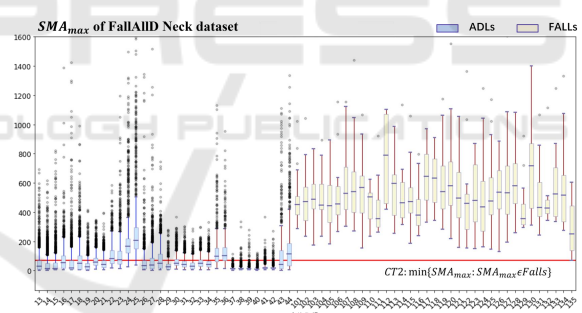
(c) Boxplot of  $SMV_{max}$  for FallAIID Waist dataset



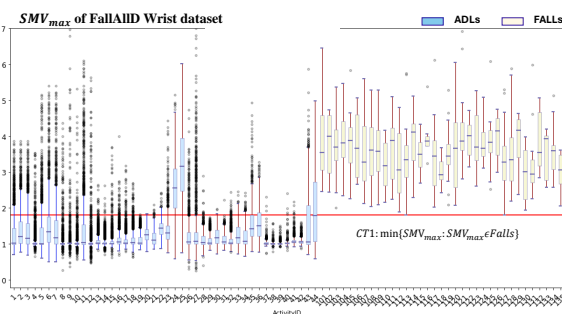
(d) Boxplot of  $SMA_{max}$  for FallAIID Waist dataset



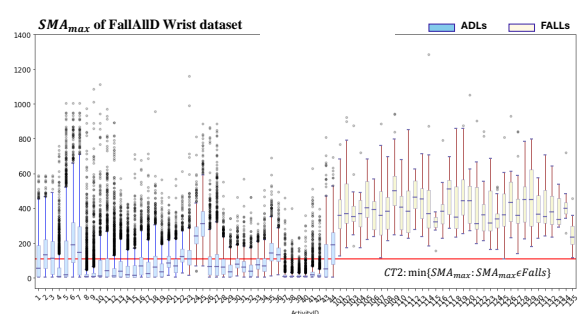
(e) Boxplot of  $SMV_{max}$  for FallAIID Neck dataset



(f) Boxplot of  $SMA_{max}$  for FallAIID Neck dataset



(g) Boxplot of  $SMV_{max}$  for FallAIID Wrist dataset



(h) Boxplot of  $SMA_{max}$  for FallAIID Wrist dataset

Figure 6: The boxplots of statistical features of the dataset provide a visual representation of their distribution and the thresholds.