Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

A Comparative Study of Machine Learning Models for Fall Detection on the WEDA Dataset

Sarah Madi[#1], Kaouthar Madi[*2]

[#]*LRPE, USTHB,*

 # Department of computer Science, University of Algiers1
Algiers, Algeria

[*]*Department of Mechanical and Aerospace engineering, UAEU*
Alain-UAE

[1]s.madi@univ-alger.dz

[2]201735186@uaeu.ac.ae

***Abstract*— Falls are a leading cause of injury and the second most common cause of unintentional injury-related deaths worldwide, particularly among the elderly, posing significant health risks and challenges for timely assistance. Effective fall detection systems are crucial for rapid intervention to reduce injury severity. However, accurate and reliable detection remains a challenge due to the variability in fall types and daily activities. This study focuses on developing a robust fall detection system using wrist-worn sensors from the WEDA fall dataset, extracting key features such as minimum, maximum, mean, and variance to create a labeled dataset suitable for machine learning. We explored and applied various preprocessing techniques and machine learning algorithms to evaluate their performance in fall detection accuracy. Our results indicate that minimal preprocessing combined with key feature extraction yields high accuracy ranging from 97% to 99%, demonstrating that wrist-based data and selected algorithms can provide reliable fall detection and facilitate future hardware implementation for real-world use. These findings underscore the potential of wrist-worn sensor data and appropriate machine learning pipelines to address the critical need for effective fall detection in elderly care.**

***Keywords*— Fall Detection, Sensor data processing, Machine Learning, SVM, KNN**

I. INTRODUCTION

Falls are a major cause of injury worldwide and the second most common cause of unintentional injury-related deaths, with elderly people being particularly affected [1]. Data from the Centers for Disease Control and Prevention (CDC) shows that annually, over 25% of adults aged 65 and older experience a fall, and more than 20% of these falls result in serious injuries [2]. Efficient Fall Detection is a key measure to minimize the time until help arrives and reduce the consequences of a fall, enabling healthcare workers or family members to notice the event and intervene quickly [3]. Even if no external injuries are witnessed and the person who experienced the fall got up and completed their task, risk of internal injuries is reduced if appropriate care is given succeeding a fall if detected.

Machine learning (ML), particularly supervised learning, is an important method for detection based on robust datasets. Given the variety of available algorithms and their high accuracy, fall detection is effectively achievable. Datasets are widely available in the literature and typically comprise sensor readings gathered mainly from accelerometers, gyroscopes, and magnetometers. Sometimes timestamps are also recorded for further processing. The placement of sensors can influence outcomes and affect user comfort. Some datasets prefer multiple position placement such as neck, wrist, ankle and waist. Other datasets focus on one sensor position to be more convenient when applied in real life. Then comes the preprocessing of raw data captured from sensors, feature extraction and knowledge preparation to be presented in the form of inputs and labelled outputs if ML is the goal [1,9].

The WEDA Fall Dataset (Wrist Elderly Daily Activity and Fall Dataset) proposed in [1] was gathered using a Fitbit wristband. Sensor attachment at the wrist is suitable and provides readings that effectively assist in fall

ISSN: 2961-6611

Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

detection. This dataset encompasses a variety of activities and fall types, including elderly participants in their trials.

The purpose of this work is to apply ML algorithms to the WEDA dataset for fall detection with minimal preprocessing and simple features/attributes. The dataset provides only raw readings from accelerometers, gyroscopes, and magnetometers along with their timestamp information. Applying ML algorithms will require further preprocessing, such as feature extraction and data transformation using standardization and normalization, which are essential for some algorithms. Choosing the appropriate hyperparameters for each tested ML algorithm can also be time-consuming and challenging.

The main contribution of this paper is achieving high accuracy fall detection by extracting key basic features, then applying algorithms to both raw and pre-processed features to make comparisons and draw conclusions. The accuracies achieved (97-99%) compete with the outcomes of original work that involved more extensive preprocessing, including frequency variations, time window analysis, and more.

II. RELATED WORK

Fall detection using sensor readings from wearable devices (wrist, waist, head, etc.) has been gaining more popularity, especially compared to visual sensors, which have issues related to privacy and high computational cost [4]. The location of the device on the user's body affects the final classification performance, and many studies have been conducted over the past years to find the optimal location [[3], [5] [6], [7], [8], [9]]. Previously, studies focused on waist-based devices [[1], [10], [11], [12]] due to the large contrast in waist movements during daily activities (minimal) versus falling (sudden and large), which makes distinguishing falls easier [4]. However, for users, the most comfortable and commonly used location for wearable devices is the wrist, which has led to an increase in research focusing on wrist-based wearable devices for fall detection, especially to improve the detection of falls since it is more complicated compared to waist-based devices due to the diversity and frequency of wrist movements.

In the context of wrist-based devices for fall detection, the authors in [13] developed and validated a fall detection system using a wrist-worn tri-axial accelerometer (3DACC). They considered data from different datasets (UMAFALL, DaLiAc, etc.) and used a scheme to extract eight fall dynamics-related features (Maximum Peak Index, Peak Duration Index, etc.) whenever a peak was detected in the acceleration data. Different machine learning models, including Decision Trees (DT), Support Vector Machines (SVM), Neural Networks (NN), and Rutherford backscattering Spectrometry (RBS), were evaluated based on various metrics (accuracy, sensitivity, precision, etc.), and it was found that all models had comparable performance, with some performing better for certain metrics and others excelling in different ones.

de Quadros et al. [3] utilized a wrist-based device with various sensors (gyroscope, accelerometer, magnetometer) to compare the performance of threshold-based and ML-based methods for fall detection. They used Madgwick's decomposition to fuse data from the sensors and estimate the device's spatial orientation, using it to derive high-level signals such as vertical acceleration, which were then used to extract features directly related to fall events, leading to enhanced overall detection. For the threshold-based method, the best accuracy obtained was 91.1% when Madgwick's decomposition was used. All ML methods (k-NN, DT, SVM, etc.) outperformed the threshold-based methods, with k-NN showing the best overall performance, achieving an accuracy of 99%.

Marques & Moreno [1] addressed a significant gap in previous studies on wrist-based fall detection by introducing the WEDA-FALL dataset, which includes fall and daily activity data from both younger and elderly individuals. They conducted comprehensive offline experiments using low-computational features (maximum, minimum, variance, mean) with different machine learning models (k-NN, SVM, DT) to find the parameters (sensors considered, learning algorithm, feature computation, window size) that yield the best fall prediction while considering the watch's battery consumption and adding the capability of receiving user feedback when the watch is online to further enhance performance.

III. DATASET DESCRIPTION

The work in [1] proposes the WEDA Fall Dataset (Wrist Elderly Daily Activity and Fall Dataset) [14] gathered using a Fitbit wristband. It includes various types of falls (forward, backward, and lateral) occurring during daily life activities. This selection was based on the frequency of real-life activities that could generate false positives. The dataset also contains multiple Activities of Daily Living (ADLs), such as walking, jogging,

ISSN: 2961-6611

and sitting down before standing up. The authors ensured a diverse group of participants, ranging in age from 20 to 88 years, hence, including both elderly and young individuals. The dataset is organized to differentiate between falls and ADLs, indicating the round number and whether the participant is young or elderly. Some actions were repeated across multiple rounds while ensuring participant safety. Data from accelerometers, gyroscopes, and orientation sensors were captured at a frequency of 50 Hz. The dataset primarily comprises the raw data from these sensor readings, and the authors applied filters to reduce frequencies to 40 Hz, 25 Hz, 10 Hz, and 5 Hz. All files are included in the final dataset. Additionally, the authors performed basic preprocessing and applied machine learning models to the raw data from young participants to gain initial insights into the system's behavior. The study was subsequently extended to include various window sizes and additional analyses.

All raw data was organized into folders indicating whether it pertains to a fall or an ADL (Activities of Daily Living). Movements from different rounds and participants are labelled and grouped in designated folders. Each individual movement has separate files for accelerometer, gyroscope, and orientation raw readings. Some movements are accompanied by an additional file containing vertical acceleration raw data.

The acceleration and gyroscope raw data consist of four readings: time, and readings along the x, y, and z axes. The orientation raw data includes time and four other different readings. We decided to focus on raw data for both young and elderly participants, while excluding time information from the study.

IV. METHODOLOGY

This research commenced with extracting features and preparing the final dataset file. Next, we studied distribution information for each feature to determine which preprocessing techniques to apply. Only then, our data was ready to start exploring various ML algorithms and conclude the more suitable and accurate to be applied with wrist-based Fall detection systems.

A. Feature Extraction

A Python script was used to process multi-sensor data by scanning folders, reading accelerometer, gyroscope, and orientation files, and extracting statistical features for fall detection. For each session, the script computes the minimum, maximum, mean, and variance of each sensor axis, assigns a class label (Fall or nonFall), and aggregates the results into a single dataset. The final dataset contains 969 instances with 40 features, providing an efficient and widely accepted statistical representation suitable for machine learning analysis. A recent study on low-cost environmental sensor data shows that statistical characterization enhances model predictions by stabilizing noisy raw readings and assisting in algorithm calibration [15]. Although these basic features have limitations, such as their inability to capture temporal dynamics or complex structural patterns, they are crucial for initial data processing and are often supplemented by advanced modeling techniques to improve accuracy. Next, we check the distribution and nature of individual features to apply the appropriate preprocessing techniques.

B. Pre-processing

Machine learning models are sensitive to feature scale, and large differences in value ranges can bias learning toward dominant features. Analysis of the extracted features revealed substantial variation across sensors and statistical measures, with some features spanning wide ranges while others remain narrowly bounded. This scale disparity highlights the necessity of preprocessing techniques such as standardization or normalization to balance feature influence and improve model performance and stability in fall detection tasks [16].

The dataset was preprocessed using standardization, normalization, and a combined approach that applies standardization followed by normalization. These techniques make features comparable by controlling scale and variance, preventing large-magnitude features from dominating learning. Standardization is especially important for distance-based and margin-based models such as KNN and SVM, while normalization ensures consistent feature ranges. Overall, these preprocessing steps improve model accuracy, stability, and training efficiency. Kim et al. [17] investigated the impact of data normalization on energy consumption prediction using neural networks, demonstrating that normalization significantly improved model convergence speed and prediction accuracy by reducing bias from differently scaled features. Similarly, the research presented by [18], emphasized that data standardization enhances data preparation quality, leading to more robust downstream models by mitigating scale-related distortions in the data.

ISSN: 2961-6611

Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

Empirical evidence from multiple models evaluated, showed that models trained on standardized datasets had higher stability, better generalization, and faster convergence than those trained on unscaled data, confirming standardization as a best practice for diverse algorithms including Support Vector Machines and k-Nearest Neighbors [19]. In the next subsection, we list the main ML algorithms used along with the reason of each choice.

C. Machine learning algorithms

K nearest neighbors (kNN) was chosen for its simplicity, less affected by noise and the fact that is has no model building, hence the best training time. It has also proven to achieve good accuracy in previous fall detection work [1].

Decision Trees (DT) will be used for its high interpretability, providing insights into which features are most important and the underlying decision logic. This approach aligns with recent studies where decision trees have demonstrated strong performance in fall detection tasks, providing a balance of interpretability, accuracy, and computational efficiency [20], [21]. The multi-criteria evaluation helps identify the best criterion for the specific sensor feature data, contributing to robust classifier design.

Random Forest (RF) is a widely used machine learning method in fall detection due to its robustness and high accuracy, as shown in recent studies where it outperformed other algorithms like SVM and KNN by effectively handling complex sensor data and reducing false positives and negatives [22], [20]. Its ensemble nature enhances generalization by combining multiple decision trees, making it suitable for real-time fall detection applications on wearable sensors. RFs mitigate overfitting which is a common issue with DTs and to enhance model robustness. It is also less sensitive to noise and outliers in the data.

Finally, Support Vector Machines (SVM) are an excellent option for fall detection because of their high accuracy and ability to capture complex patterns in sensor data, as demonstrated by recent research. Furthermore, SVM models generally provide faster inference times than other classifiers, making them ideal for real-time, time-sensitive fall detection applications where a prompt response is essential [16]. The following table summarizes the hyperparameters for each algorithm and how they were tuned.

TABLE I
ALGORITHMS AND HYPERPARAMETERS USED

Model	Tuned Hyperparameters	Values/Options Tested or Used	Notes
K-Nearest Neighbors (KNN)	Number of neighbors (k)	3, 5, 7, 9	Explored multiple k values to optimize accuracy
	Distance metric	Default (Euclidean)	Mostly default used
Support Vector Machine (SVM)	Kernel	Radial Basis Function (RBF), Polynomial	Both kernels tested
	Kernel parameters	Standardized data used; kernel parameters not explicitly tuned but compared	RBF kernel with standardized data performed best
Decision Tree (DT)	Splitting criterion	gini, entropy, log_loss	Criterion compared for best performance
	Other parameters	Default values	No other hyperparameters tuned
Random Forest	Number of estimators	Default 100	Fixed at 100
	Maximum tree depth	None (fully grown)	No explicit tuning done

Models like KNN and SVM had specific parameters explored, while Decision Tree and Random Forest tuning focused mostly on criterion and default values, respectively.

Now, we present the metrics used to assess the performance of each ML algorithm and their corresponding description.

D. Evaluation Metrics

Evaluation metrics are crucial for assessing the performance of machine learning models in classification tasks. The statistics from the testing, which included the number of correctly detected falls (TP), falsely detected falls (FP), correctly detected non-falls (TN), and falsely detected non-falls (FN), were obtained and used to calculate different performance metrics, including accuracy, precision, and sensitivity. These metrics are essential because they provide a clearer picture of performance when the data is imbalanced, which is often the case in fall detection. Finally, cross-validation accuracy represents the average performance across

ISSN: 2961-6611

Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

multiple training folds, offering a robust estimate of generalization capability. In this work, the number of folds was set to 5.

V. RESULTS

The study evaluates several ML models with different hyperparameters and preprocessing methods for fall detection. **KNN** performance is strongly influenced by preprocessing, with standardized data and smaller k values (3–5) yielding the best results (up to 98.63% accuracy), confirming the importance of feature scaling. **Decision Trees** show stable but moderate performance (~90–91% accuracy), largely unaffected by preprocessing or splitting criteria. **SVM** achieves the best overall results when using standardized data with an RBF kernel, reaching about 99.3% accuracy and excellent sensitivity and specificity, highlighting the critical role of preprocessing and kernel choice. **Random Forest** demonstrates consistently high and robust performance (~97.6% accuracy) across all preprocessing methods, showing strong generalization and low sensitivity to feature scaling. Overall, preprocessing is crucial for KNN and SVM, while Random Forest remains robust regardless of scaling. Here is a final table summarizing the best outcome from each ML algorithm:

TABLE II
SUMMARY OF RESULTS

Model	**Best Preprocessing**	**Key Metrics**	**Test Accuracy**	**CV Accuracy Mean**
KNN	Standardized (k=3)	Precision ~0.97, Recall ~0.99, F1 ~0.98	0.9863	0.9676
SVM	Standardized (RBF kernel)	Precision ~0.99, Recall ~0.99, F1 ~0.99	~0.9931	~0.9675
RF	Raw / Standardized / Normalized / Combined	Precision ~0.96, Recall ~0.97, F1 ~0.97	0.9759	~0.9587
DT	Raw / Standardized / Normalized / Combined	Precision ~0.87, Recall ~0.88, F1 ~0.88	~0.9141	~0.8894

As noticed from this concluding table that SVM achieved the highest accuracy and classification metrics, particularly with the RBF kernel and standardized preprocessing, reaching nearly 99% accuracy. KNN followed closely, especially when properly tuned (k=3) and standardized, delivering strong and competitive results. Random Forest exhibited robust and stable high performance across all preprocessing methods, maintaining accuracy around 97.6%. The Decision Tree, while consistent and stable across different preprocessing types, had the lowest accuracy (~91%) but remains a reliable and interpretable baseline. Overall, SVM and KNN with standardized data are the top performers for fall detection, with Random Forest also very effective, and Decision Tree serving as a simpler alternative.

VI. CONCLUSION

This study conducted a comprehensive evaluation of sensor-based fall detection using multiple machine learning models, with a strong focus on feature extraction, data preprocessing, and model optimization. Raw sensor signals were transformed into meaningful statistical features and preprocessed using standardization, normalization, and a combined approach to improve learning stability. Several classifiers were trained and tuned, including Decision Tree, Random Forest, K-Nearest Neighbors, and Support Vector Machines.

Among all models, the SVM using an RBF kernel with standardized data delivered the best overall performance, achieving a test accuracy of 99.31%, a cross-validation accuracy of 97%, and an excellent balance between sensitivity and specificity (both 0.99). While other models also produced robust and reliable results, their performance was slightly lower, emphasizing the critical impact of appropriate preprocessing and kernel selection. These results are consistent with recent validation studies and slightly exceed the accuracy reported by the dataset creators. Future research may explore real-time implementation and the integration of deep learning techniques, such as CNNs and LSTMs combined with sensor fusion, to further improve accuracy and robustness in real-world fall detection systems.

ISSN: 2961-6611

Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

REFERENCES

[1] J. Marques and P. Moreno, "Online Fall Detection Using Wrist Devices," *Sensors*, vol. 23, no. 3, p. 1146, Jan. 2023, doi: 10.3390/s23031146.

[2] CDC, "Older Adult Falls Data," Older Adult Fall Prevention. [Online]. Available: https://www.cdc.gov/falls/data-research/index.html

[3] T. De Quadros, A. E. Lazzaretti, and F. K. Schneider, "A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device," *IEEE Sensors J.*, vol. 18, no. 12, pp. 5082–5089, Jun. 2018, doi: 10.1109/JSEN.2018.2829815.

[4] S. Li, "Fall Detection With Wrist-Worn Watch by Observations in Statistics of Acceleration," *IEEE Access*, vol. 11, pp. 19567–19578, 2023, doi: 10.1109/ACCESS.2023.3249191.

[5] A. Bagnasco, A. M. Scapolla, and V. Spasova, "Design, implementation and experimental evaluation of a wireless fall detector," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Barcelona Spain: ACM, Oct. 2011, pp. 1–5. doi: 10.1145/2093698.2093763.

[6] T. R. Bennett, J. Wu, N. Kehtarnavaz, and R. Jafari, "Inertial Measurement Unit-Based Wearable Computers for Assisted Living Applications: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 28–35, Mar. 2016, doi: 10.1109/MSP.2015.2499314.

[7] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic Fall Monitoring: A Review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, Jul. 2014, doi: 10.3390/s140712900.

[8] Jian Yuan, Kok Kiong Tan, Tong Heng Lee, and G. C. H. Koh, "Power-Efficient Interrupt-Driven Algorithms for Fall Detection and Classification of Activities of Daily Living," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1377–1387, Mar. 2015, doi: 10.1109/JSEN.2014.2357035.

[9] N. Noury, P. Rumeau, A. K. Bourke, G. ÓLaighin, and J. E. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM (Innovations et Applications en Biologie et Médecine)*, vol. 29, no. 6, pp. 340–349, 2008.

[10] M. Saleh and R. L. B. Jeannes, "Elderly Fall Detection Using Wearable Sensors: A Low Cost Highly Accurate Algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 2019, doi: 10.1109/JSEN.2019.2891128.

[11] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "UP-Fall Detection Dataset: A Multimodal Approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019, doi: 10.3390/s19091988.

[12] F. Hussain, F. Hussain, M. Ehatisham-ul-Haq, and M. A. Azam, "Activity-Aware Fall Detection and Recognition Based on Wearable Sensors," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019, doi: 10.1109/JSEN.2019.2898891.

[13] S. B. Khojasteh, J. R. Villar, C. Chira, V. M. González, and E. De La Cal, "Improving Fall Detection Using an On-Wrist Wearable Accelerometer," *Sensors*, vol. 18, no. 5, p. 1350, Apr. 2018, doi: 10.3390/s18051350.

[14] João Marques and Plinio Moreno, "WEDA-FALL." 2022. [Online]. Available: https://github.com/joaojtmarques/WEDA-FALL

[15] T. Barrett and A. K. Mishra, "Statistical Study of Sensor Data and Investigation of ML-based Calibration Algorithms for Inexpensive Sensor Modules: Experiments from Cape Point," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–10, 2024, doi: 10.1109/TIM.2024.3372211.

[16] T. Y. Koffi, Y. Mourchid, M. Hindawi, and Y. Dupuis, "Machine Learning and Feature Ranking for Impact Fall Detection Event Using Multisensor Data," in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2023, pp. 1–6. doi: 10.1109/MMSP59012.2023.10337682.

[17] Y.-S. Kim, M. K. Kim, N. Fu, J. Liu, J. Wang, and J. Srebric, "Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models," *Sustainable Cities and Society*, vol. 118, p. 105570, Jan. 2025, doi: 10.1016/j.scs.2024.105570.

[18] E. Lai, Y. Lou, B. Youngmann, and M. Cafarella, "Toward Standardized Data Preparation: A Bottom-Up Approach." OpenProceedings.org, 2025. doi: 10.48786/EDBT.2025.49.

[19] K. Mahmud Sujon, R. Binti Hassan, Z. Tusnia Towshi, M. A. Othman, M. Abdus Samad, and K. Choi, "When to Use Standardization and Normalization: Empirical Evidence From Machine Learning Models and XAI," *IEEE Access*, vol. 12, pp. 135300–135314, 2024, doi: 10.1109/ACCESS.2024.3462434.

[20] S. Almeraikhi and M. Al-Rajab, "Machine Learning Algorithms for Early Fall Detection of Elderly People," *JITM*, vol. 14, no. 2, Apr. 2022, doi: 10.22059/jitm.2022.86925.

[21] F.-Y. Leu, C.-Y. Ko, Y.-C. Lin, H. Susanto, and H.-C. Yu, "Fall Detection and Motion Classification by Using Decision Tree on Mobile Phone," in *Smart Sensors Networks*, Elsevier, 2017, pp. 205–237. doi: 10.1016/B978-0-12-809859-2.00013-9.

[22] L. Afuan, "Enhanced Fall Detection using Optimized Random Forest Classifier on Wearable Sensor Data," *J. Appl. Data Sci.*, vol. 6, no. 1, pp. 213–224, Jan. 2024, doi: 10.47738/jads.v6i1.498.

ISSN: 2961-6611

Proceedings of Book Series PBS -Vol 31, pp.8-15 **Advances in Control Systems and Information Technology**

ISSN: 2961-6611