



Fall Detection Analysis Using Various Machine Learning Techniques

Ahmad Nazri Ali¹, and Lee Jia Qing¹

¹School of Electrical and Electronic Engineering, University Science Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia.

KEYWORDS

Support Vector Machines
Fall Detection System
Machine Learning

ARTICLE HISTORY

Received 10 Dec 2025
Received in revised form
18 Aug 2026
Accepted 31 Aug 2026
Available online 21 Sep 2026

ABSTRACT

Falls represent a critical health hazard for the elderly demographic, often inducing severe physical trauma and a subsequent decline in psychological well-being that severely compromises their quality of life. To mitigate these risks and foster independent living, the development of a highly dependable and accurate automated fall detection system is of paramount importance. This paper introduces a machine learning-driven fall detection system that utilizes accelerometer data to categorize various activities. The system was trained and evaluated using three machine learning algorithms: Support Vector Machine (SVM), k-Nearest Neighbors (kNN), and Random Forest (RF), employing the publically accessible SisFall dataset. The method employs a low-pass filter in the data preprocessing phase to eliminate noise from sensor readings, subsequently utilizing an overlapping sliding window technique for feature extraction. The findings indicate that all three classifiers attained outstanding performance, with accuracy, sensitivity, and specificity surpassing 90.0%. Of the three algorithms evaluated, Random Forest exhibited the highest performance, rendering it the optimal selection for fall detection systems intended for elderly individuals.

© 2026 The Authors. Published by Penteract Technology.

This is an open access article under the CC BY-NC 4.0 license (<https://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

The World Health Organization (WHO) states that falls are the second leading cause of accidental fatalities globally, with individuals aged 60 and above being more susceptible to severe injuries resulting from falls. Falls impact on individuals not only literally (fractures, disabilities, contusions, edema, etc.) but also psychologically, as the elderly may experience a fear of falling, anxiety, and sadness [1]. The elderly must be continuously observed to guarantee their safety. The Fall Detection System (FDS) is attracting considerable attention in research due to the health risks associated with falls. The primary objective of developing FDS is to swiftly detect and recognize fall incidents and notify companions to reduce rescue time and potential injuries. The duration following a fall is a key factor in assessing the severity of the incident [2].

Prior research conducted by Gharghan and Hashim[3] provided an extensive evaluation of fall detection systems (FDSs) through a thorough comparison, assisting researchers in determining the most accurate and appropriate methodologies for detecting falls in older persons. The wearable-sensor-based system incorporates embedded sensors, including accelerome-

ters and a gyroscope, as an explored strategy for developing a fall detection system in wearable devices. The accelerometer measures acceleration, whereas the gyroscope measures the rate of rotation along the x, y, and z axes.

Fall detection methodologies are classified into two principal categories: threshold-based and machine learning-based. The threshold-based method detects a fall when the values of specific features surpass predetermined thresholds [4-5]. Machine learning (ML) technology identifies patterns in the data and categorizes actions as either activities of daily living (ADL) or falls. Distinguishing between an ADL and a fall is difficult, as certain ADLs exhibit parallels to falls [6]. For instance, sitting abruptly will result in a sudden change in acceleration, as will lying on the floor. A crucial element for fall detection is the ability to organize, analyze, and extract significant information from raw inertial sensor data.

A substantial and expanding corpus of literature has emphasized the utilization of machine learning approaches in fall detection systems, yielding encouraging outcomes and attracting ongoing interest from researchers [7-8]. Consequently, machine learning may provide a solution to the

*Corresponding author:

E-mail address: Ahmad Nazri Ali <nazriali@usm.my>.

<https://doi.org/10.56532/mjsat.v6i3.674>

2785-8901/ © 2026 The Authors. Published by Penteract Technology.

This is an open access article under the CC BY-NC 4.0 license (<https://creativecommons.org/licenses/by-nc/4.0/>).

challenges previously encountered in recognizing fall events. The fall detection paradigm can be viewed as a binary classification issue, distinguishing between activities of daily living (ADL) and falls, utilizing various machine learning classification algorithms.

This paper proposes a fall detection system utilizing accelerometer-derived data through machine learning-based categorization methods. This paper will employ the SisFall dataset, gathered from wearable sensors, for training and testing purposes. A comparative analysis of various prominent machine learning classifier algorithms will be conducted to ascertain the highest-performing algorithm. A thorough comparison analysis is conducted by comparing the findings of this work with those of prior research.

2. RELATED WORK

In recent years, extensive research has been conducted to develop a fall detection system. Fall detection methods are classified into threshold-based and machine learning-based categories. The ML-based approach has been shown to yield superior outcomes when compared to several threshold-based methods [9]. Nonetheless, T. Xu, H. Se, and J. Liu [10] propose a compelling method that integrates both techniques into a unified solution. The solution attained successful outcomes with 97.83%, 96.64%, and 97.02% in sensitivity, specificity, and accuracy, respectively. Rassekh and Snidaro [11] provide a thorough examination of threshold-based and machine learning-based fall detection systems utilizing wearable devices.

Prior research has demonstrated that machine learning-based fall detection systems have yielded promising outcomes [9-10], [12]. The objective of the FDS is to ascertain whether a fall event has occurred, rendering it a binary classification problem (detected fall events yield 1, otherwise 0). Consequently, the supervised learning strategy, adept at classification tasks, is designed to train the model using input data annotated with specific outputs [13].

Zurbuchen et al. [14] chose five machine learning classification techniques for comparison: kNN (k-Nearest Neighbor), SVM (Support Vector Machine), DT (Decision Trees), RF (Random Forest), and GB (Gradient Boosting). Among the algorithms, Random Forest (RF) and Gradient Boosting (GB) exhibited the most optimal performance, achieving sensitivity and specificity levels near 99%. Vallabh et al. [15] suggest more machine learning methods, including kNN, Naïve Bayes, Least Squares Method (LSM), and Artificial Neural Network (ANN). The kNN classifier has proven to be the most effective in this instance.

Moreover, other pertinent studies have demonstrated distinct methodologies in the feature extraction phase to derive significant attributes for classification training input [16-18]. Palmerini et al. [12] presents a window process to detect the window encompassing the peak acceleration of a fall, together with attributes extracted from the resultant falling windows. Pant et al. [19] presented overlapping sliding window approaches to prevent the omission of valuable information. Overlapping windows refer to the intersection of adjacent windows when sensory data is divided into segments of a designated size. Dehghani et al. [20] have demonstrated that the

overlapping window technique enhances the recognition capability of classifiers in a human activity recognition study.

3. METHODOLOGY

3.1 Overview of the Proposed System

Data preparation, feature extraction, feature preprocessing, training and testing, and evaluation of the classifiers are several processes that make up the suggested approach to developing the FDS for senior persons. Gathering acceleration data for different ADLs and falls is the first step in the suggested method's cycle. The next step is to calculate the acceleration after doing data preprocessing to remove noise from the sensory input. The feature extraction stage follows, during which the preprocessed data is scoured for pertinent features; these features will later be utilized for categorization. Following feature extraction, each activity is assigned a label: 0 for daily activities of living and 1 for fall instances.

The processed dataset is subsequently divided into training and testing datasets for the machine learning classifiers. The machine learning classifiers are taught using the training set and assessed with the test set. Various performance indicators are calculated to assess the efficacy of the trained classifiers. This project selects three distinct ML classifiers to evaluate their performance. The chosen classification techniques are SVM (Support Vector Machine), RF (Random Forest), and kNN (k-Nearest Neighbors).

3.2 Dataset

This paper utilizes the publicly accessible SisFall dataset, as its different activities correspond to numerous daily tasks [21]. The information was acquired using two triaxial accelerometers (ADXL345 and MMA8451) and one triaxial gyroscope (ITG3200) at a sampling frequency of 200 Hz. The dataset comprises 19 categories of prevalent Activities of Daily Living (ADLs) and 15 types of falling incidents, conducted by 38 participants, including both elderly and young adults. This experiment will utilize exclusively data acquired from the ADXL345 triaxial accelerometer. This specific study limits its data source to the ADXL345 to evaluate the performance of its model using only a single, primary triaxial acceleration sensor.

3.3 Preprocessing

At this point, the raw acceleration data from the Activities of Daily Living (ADLs) and fall activities, recorded in bits, are translated into units of gravity (g) using Equation 1, given that the range and resolution are ± 16 g and 13 bits, respectively [22].

$$Acceleration(g) = \left(\frac{2 \cdot Range}{2^{\cdot Resolution}}\right)(Acceleration \text{ data in bits}) \quad (1)$$

The acceleration data are processed using a 4th-order Butterworth low-pass filter with a cut-off frequency of 10 Hz. The objective is to eliminate extraneous noise while preserving the integrity of the original data. Initial experiments with various cut-off frequencies are conducted to ascertain the appropriate cut-off frequency for this phase. The acceleration magnitude, A_T , for all actions is subsequently calculated using Equation 2.

$$A_T = \sqrt{A_x^2 + A_y^2 + A_z^2} \quad (2)$$

3.4 Feature Extraction

A 50% overlapping sliding window approach is employed to extract recognizable features from the pre-processed data. Overlapping refers to the intersection of adjacent windows. The data will initially be segmented into time periods of specified duration, from which various features will be retrieved. The identified traits are enumerated in Table 1. An overlapping window technique is employed to avert the loss of information from the data, as the windows do not consistently align with the actual fall segment due to temporal variations. A 50% overlap ensures that any transient signal occurring at the peripheral boundary of a given window is repositioned into the center of the subsequent window. Furthermore, a 50% threshold represents the optimal mathematical compromise between ensuring continuous event capturing and minimizing data redundancy, preventing the computational overhead and overfitting risks associated with hyper-dense (e.g., 75% or 90%) overlapping schemes. The retrieved features are subsequently labeled with their respective activities before being entered into the supervised learning classifier. In Table 1, skewness and kurtosis are chosen as the list of features due to both can provide vital statistical discriminability by evaluating the shape of the signal distribution within each window. Skewness measures acceleration asymmetry; controlled movements like rapid sitting. Concurrently, Kurtosis quantifies the 'peakedness' of the signal.

Table 1. List of features extracted from the data.

Features	Domain
Mean	Time
Minimum	Time
Maximum	Time
Variance	Time
Skewness	Time
Kurtosis	Time

3.5 Model Training

Following the feature extraction phase, the pre-processed and labeled dataset is divided into training and testing sets. The former involves fitting or training the model for classification tasks, while the latter entails verifying the model's performance accuracy. This study designates 0.67 of the dataset for the training set, with the remaining 0.33 allotted to the test set.

K-fold cross-validation is conducted to reaffirm the accuracy of the training model, hence illustrating its robustness. K-fold cross-validation divides the dataset into k pieces of roughly equal size. One fold is designated as a test set, while the remaining k – 1 folds are utilized to train the model. The procedure is executed k times, with distinct folds as the test set for each iteration. In this paper, k is established as 10.

Three distinct classification algorithms will be employed to train on the identically labeled dataset, and their performances will be compared: k-Nearest Neighbor (kNN), Random Forest (RF), and Support Vector Machine (SVM). All these algorithms are prominent supervised learning techniques and vary significantly in methodology. The efficacy of the classification method is also influenced by its corresponding parameters. Consequently, initial evaluations of each algorithm along with their corresponding parameters are performed. The kNN algorithm classifies data by identifying the most similar instances. Preliminary tests are performed using various values of k, commencing with 1 and subsequently including 3, 5, 7,

and 9. Meanwhile, for Random Forest, it executes numerous decision trees on the data samples and averages the outcomes from these trees to ascertain the categorization. Initial tests are performed using varying quantities of trees, n, commencing with 100, 300, 500, 700, and 900. The final classifier, SVM, was executed using various kernels: Polynomial, Linear, Sigmoid, and RBF.

3.6 Performance Evaluation

Upon completion of the training procedure, the classifiers are assessed using the testing sets. The classifiers' performance is assessed using various metrics to demonstrate the trained model's reliability and efficiency. The confusion matrix is a statistic that offers a comprehensive assessment of the model's classification accuracy and the nature of its errors, as illustrated in Table 2. The performance measures, including accuracy, specificity, sensitivity, precision, and F1-score, are calculated using the outcome values from the confusion matrix.

Table 2. Definition of the basic outcomes of the confusion matrix in this case.

Terms	Definition
True Positive (TP)	A fall happens; the model detects it
False Positive (FP)	A fall does not happen; the model reports a fall.
True Negative (TN)	A fall does not happen; the model does not report a fall.
False Negative (FN)	A fall happens; the model misses it.

Accuracy is the metric that quantifies the proportion of correct predictions relative to the total number of classes. Sensitivity, or recall, denotes the proportion of real positive cases that the model predicts appropriately. Specificity refers to the ratio of true negatives that have been accurately identified as negative. Precision indicates the proportion of accurately predicted affirmative cases. This signifies the reliability of the model. The F1-score serves as the metric for assessing performance accuracy. The formulas for accuracy, sensitivity, specificity, precision, and F1-score are shown in Equations 3, 4, 5, 6, and 7, respectively.

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

$$Specificity = \frac{TN}{FP+TN} \quad (4)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (5)$$

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

$$F1\ score = \frac{2*Precision*Recall}{Precision+Recall} \quad (7)$$

4. RESULTS AND DISCUSSION

The experimental outcomes are examined using multiple evaluative methods.

4.1 Preliminary Tests for Determining the Cut-Off Frequency of Low Low-Pass Filter

The falling activity, F01, is characterized by a forward fall during ambulation due to a slip, and the outcomes from the

preliminary tests are examined. The results are illustrated using low-pass filtered data at specific cut-off frequencies, as depicted in Figures 1(a) to 1(e).

Figure 1(c) demonstrates that negligible noise is suppressed while maintaining the fidelity of the raw data, with a cut-off frequency of 10Hz. Figure 1(b) demonstrates that the peaks of data filtered using a 5 Hz cut-off frequency are reduced relative to the unprocessed data shown in Figure 1(a). The peaks are undesirable since they will serve as features to distinguish between activities of daily living (ADL) and fall events in later stages. As shown in Figure 1, a 10 Hz cut-off frequency was chosen for the filter. Testing a lower setting of 5 Hz (Figure 1b) proved to be too restrictive, as it smoothed out and erased the sharp acceleration peaks that represent a real fall impact. The 10 Hz setting successfully removes unwanted background noise while keeping the true shape of the fall intact for accurate detection.

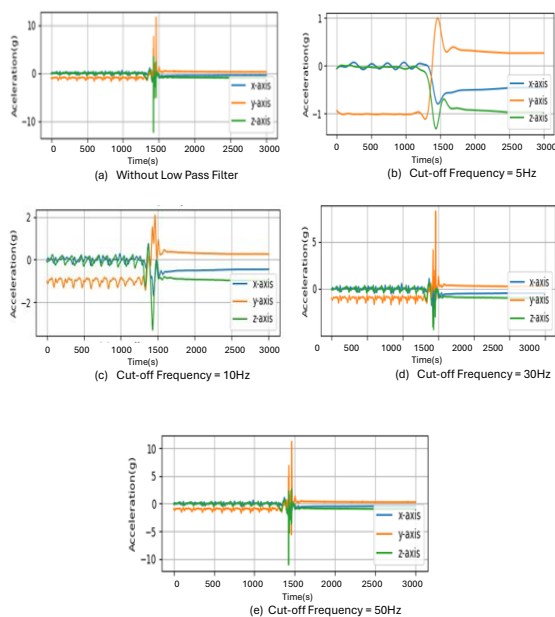


Fig. 1. (a)-(e). The plot of the unfiltered data and low-pass filtered data with cutoff frequencies of 5 Hz, 10 Hz, 30 Hz, and 50Hz.

4.2 Performance of kNN Classifier

Confusion matrices for the kNN classifier with varying values of k are presented in Figure 2 accordingly. As illustrated in Figure 2, about 146 to 183 activities of daily living (ADLs) are erroneously categorized as fall events (false positives), while 178 to 212 fall incidents are misclassified as ADLs (false negatives) in the studies involving variations in the value of k .

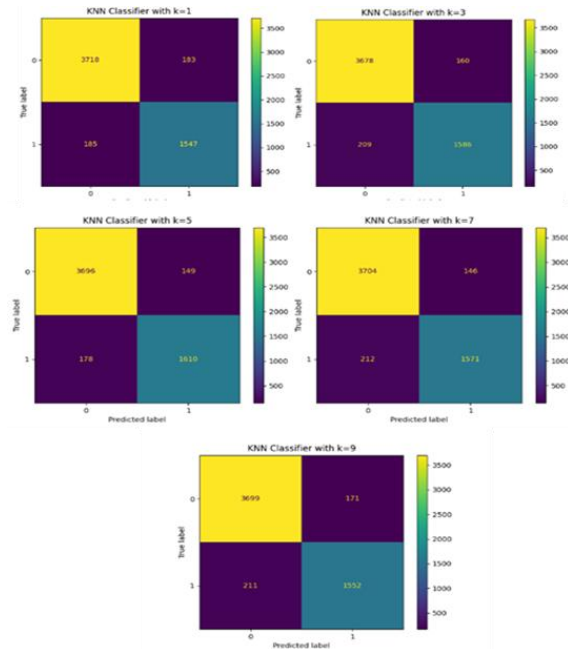


Fig. 2. Confusion matrix with varying number of trees

With these results, for a real-life safety paradigm, a false negative represents a catastrophic system failure where a genuine, potentially injurious fall is completely masked as a benign Activity of Daily Living (ADL). The findings in Table 3 summarize the comprehensive performance, indicating that the kNN classifier with $k=5$ outperformed the other k values, achieving the highest overall cross-validation accuracy. Despite minor variations in performance between k values of 5, 7, and 9, the classifier with $k=5$ attained the highest average accuracy of 93.98%. These data indicate that a k value of 5 is optimal for this dataset.

Table 3. The performance of the kNN algorithm with different values of k and the highest values in bold.

k	SE [%]	SP [%]	Accuracy [%]	Recall [%]	Precision [%]	F1-Score [%]
1	89.32	95.26	87.61	89.32	89.42	89.37
3	88.36	91.62	90.34	88.36	90.84	89.58
5	90.04	95.41	93.98	90.04	91.53	90.78
7	88.11	94.59	93.83	88.11	91.50	89.77
9	88.03	94.60	93.81	88.03	90.08	89.04

SE: Sensitivity; SP: Specificity

4.3 Performance of Random Forest (RF) Classifier

The confusion matrices for the RF classifier with differing tree quantities are presented in Figure 3. Misclassifications were infrequently encountered. Out of 5,633 activities, only 6 ADLs are inaccurately categorized as fall occurrences when $n=100$, 300, or 700, whereas 13 and 8 ADLs are misclassified when $n=500$ and 900, respectively. Furthermore, only approximately 18 to 25 cases have been inaccurately labeled as Activities of Daily Living (ADLs).

The performance indicators are calculated and presented in Table 4 to assess the efficacy of the RF classifier with varying tree values; n . Table 4 demonstrates that there are no significant differences in the performance measures of the RF classifier across varying tree values. An accuracy of 99.47% is achieved, representing the maximum accuracy when the number of trees, n , is set to 900. When the number of trees, n , is set to 500, the

maximum sensitivity and specificity achieved are 98.88% and 99.48%, respectively. The data indicate that the RF classifier with estimator values ranging from 100 to 900 has great reliability.

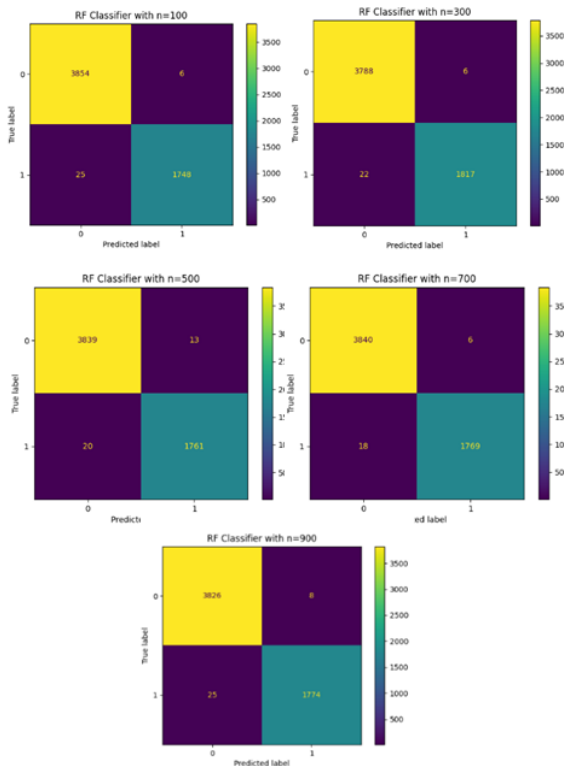


Fig. 3. Confusion matrix with a varying number of trees.

Table 4. The performance of the RF algorithm with different values of k and the highest values in bold.

n	Performance Metrics				
	SE [%]	SP [%]	Accuracy [%]	Precision [%]	F1-Score [%]
100	98.59	99.36	99.45	99.66	99.12
300	98.41	99.28	99.20	99.03	98.72
500	98.88	99.48	99.41	99.66	99.32
700	98.61	99.35	99.41	99.55	99.08
900	98.56	99.32	99.47	99.78	99.17

SE: Sensitivity; SP: Specificity

Table 5. The results of 10-fold cross-validation of the RF classifier with different values of k.

n	Accuracy (%)									
	1	2	3	4	5	6	7	9	10	Avg
1	99.39	99.48	99.39	99.48	99.56	99.56	99.39	99.30	99.39	99.44
3	98.95	99.21	99.21	99.48	99.74	99.56	99.30	99.48	99.65	99.38
5	99.39	99.39	99.30	99.39	99.48	99.39	99.13	99.48	99.13	99.35
7	99.47	99.30	99.30	99.47	99.21	99.30	98.86	99.30	98.95	99.25
9	99.21	99.30	99.39	99.12	99.39	99.56	99.56	99.56	99.21	99.35

n: the 1st digit of the values of trees in hundreds

Avg: Average

4.4 Performance of Support Vector Machines (SVM) Classifier

The confusion matrices for the SVM classifiers with various kernels are presented in Figure 4. As illustrated, the SVM classifier with the "Sigmoid" kernel performed drastically worse than the others, misclassifying 482 activities of daily living (ADLs) as fall incidents and 477 fall incidents as ADLs. This shows that the Sigmoid kernel suffers with a problem to deal with a wide range of data, the Sigmoid function flattens out, causing the algorithm to lose its sensitivity. As a result, it begins treating completely different activities as if they were exactly the same. Consequently, it creates a distorted boundary map that fails to distinguish between falls and ADLs, leading to a poor 70% accuracy. In sharp contrast, the other kernels proved highly robust and forgiving; Figure 4 indicates that only 36, 11, and 17 ADLs are misclassified as fall events for the "Linear," "Polynomial," and "RBF" kernels, respectively, while only 38, 38, and 27 fall events are erroneously classified as ADLs. This further validates that while Linear, Polynomial, and RBF kernels can naturally map out clean, smart boundaries for this dataset, the Sigmoid kernel is highly sub-optimal without flawless calibration.

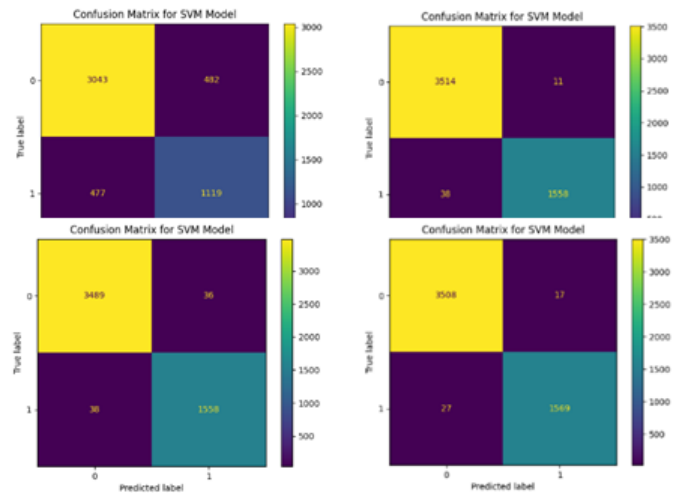


Fig. 4. Confusion matrices with different kernels.

The performance metrics are calculated and shown in Table 6 to substantiate performance across various kernels. Table 7 clearly demonstrates that the SVM classifier utilizing the "Sigmoid" kernel underperforms compared to other kernels by 18% in accuracy, 28% in sensitivity, and 13% in specificity. The results collected demonstrate that the "Sigmoid" kernel is unfavorable. Unlike the RBF kernel, which excels at grouping localized clusters of continuous movement data using radial distances, the Sigmoid kernel creates rigid decision boundaries that cannot cleanly separate intertwined data points. This mathematical incompatibility prevents the SVM from finding an optimal boundary line, resulting in a high rate of misclassifications across all metrics.

Nonetheless, with the SVM classifier utilizing other kernels, no substantial variations in any performance metrics were observed; yet the "RBF" kernel achieved the greatest values across all performance criteria. It achieved an accuracy of 99.14%, a sensitivity of 98.31%, and a specificity of 99.24%.

Table 6. Performance of SVM algorithm with different kernel, with the highest values in bold.

kernel	Performance Metrics				
	SE [%]	SP [%]	Accuracy [%]	Precision [%]	F1-Score [%]
Sigmoid	70.11	86.45	81.27	69.89	70.00
Linear	97.62	98.92	98.55	97.74	97.68
Polynomial	97.62	98.93	99.04	99.30	98.45
RBF	98.31	99.24	99.14	98.93	98.62

Table 7. The results of the 10-fold cross validation of SVM classifier with different kernels.

n	1	2	3	4	5	6	7	8	9	10	AVG
1	81.42	81.51	81.26	80.92	81.26	80.33	79.65	81.74	80.40	82.10	81.06
2	98.83	99.25	98.74	97.41	98.58	98.58	98.74	98.83	98.58	98.66	98.62
3	98.74	99.83	99.08	99.24	99.41	99.41	99.08	99.50	99.41	99.50	99.32
4	98.91	99.58	99.08	99.22	99.58	99.58	99.25	99.50	99.33	99.33	99.14

1: Sigmoid; 2: Linear; 3: Polynomial; 4: RBF

Avg: Average

4.5 Results Comparison

The findings of the three classifiers with their optimal parameters are given in Table 8. Table 8 indicates that all performance measures exceed 90.0%, yielding positive outcomes. In the comparison of classifiers, the kNN classifier trails the other two classifiers by approximately 5.0% in accuracy and specificity and by roughly 8.0% in sensitivity. Although the performance profiles of the RF and SVM classifiers are highly comparable, the RF ensemble achieves marginally superior optimization when configured with $n = 500$ trees. At this hyperparameter setting, the model yields an overall accuracy of 99.41%, supported by a sensitivity of 98.88% and a specificity of 99.48%. In comparison, expanding the ensemble to $n = 900$ trees degrades performance, reducing sensitivity and specificity to 98.56% and 99.32%, respectively. Consequently, the $n = 500$ architecture provides a mathematically superior and more robust compromise between mitigating false alarms and minimizing missed events. In summary, the RF classifier surpassed the kNN classifier, while the SVM classifier exceeded the kNN classifier.

Table 8. Performance of RF, kNN, and SVM classifiers with their respective optimal parameters.

Classifier	Performance Metrics				
	SE [%]	SP [%]	Accuracy [%]	Precision [%]	F1-Score [%]
RF (n=500)	98.88	99.48	99.41	99.66	99.32
kNN (n=5)	90.04	95.41	94.19	91.53	90.78
SVM (RBF)	98.31	99.24	99.14	98.93	98.62

SE: Sensitivity; SP: Specificity

The experimental data reveal a significant precision disparity between the two models, with kNN peaking at 91.53% ($k=5$) compared to the vastly superior 99.66% achieved by RF($n=500$). This gap highlights kNN's inherent susceptibility to generating false fall reports (false positives) when processing high-intensity ADLs, such as abrupt sitting or jumping. Because kNN computes class membership based strictly on local distance metrics, any high-amplitude ADL signal that

maps into a dense cluster of true fall coordinates will inherently trigger a false alarm.

The results achieved in this study are subsequently compared with another previous research utilizing the identical SisFall dataset [23-24], as reported in Table 9. Various classifiers are utilized in the chosen existing study; however, only the findings of three classifiers—kNN, RF, and SVM—are extracted and compared with the suggested method. Table 9 indicates that the suggested approach yields results comparable to those of previous studies across all parameters.

Table 9. Performance comparison between the proposed system in this thesis and the existing study using *SisFall* dataset.

	Our Approach			Jing Zhang et. al [23]			Kocaoglu et. al [24]		
	kNN	RF	SVM	kNN	RF	SVM	kNN	RF	SVM
Acc (%)	94.2	99.4	99.1	94.1	94.0	93.0	98.7	98.6	98.1
SE (%)	90.0	98.9	98.3	91.0	90.0	90.0	98.9	98.8	98.1
SP (%)	95.4	99.5	99.2	96.0	97.0	97.0	98.6	98.3	98.1

5. CONCLUSION

This paper implements a fall detection system utilizing three distinct classification algorithms, effectively differentiating between activities of daily living (ADL) and fall incidents within the SisFall dataset. The encouraging results indicate that the proposed system can be regarded as a reliable and robust solution for fall detection. The goal of creating a precise fall classification system for the elderly has been accomplished. This is achieved by employing a low-pass filter at the data pre-processing stage to eliminate noisy sensory data. The low-pass filter has demonstrated sufficient retention of characteristics for classifying activities of daily living and fall events. The proposed method in this paper has yielded comparable results and surpassed the performance of systems in other current studies.

REFERENCES

- [1] X. Chen, Z. Lin, R. Gao, Y. Yang, and L. Li, "Prevalence and associated factors of falls among older adults between urban and rural areas of Shantou city, China," *International Journal of Environmental Research and Public Health*, vol. 18, no. 13, Jul. 2021, doi: <https://doi.org/10.3390/ijerph18137050>.
- [2] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering Online*, vol. 12, no. 1. BioMed Central Ltd., Jul. 06, 2013, doi: <https://doi.org/10.1186/1475-925X-12-66>.
- [3] S.K. Gharghan and H. A. Hashim, "A comprehensive review of elderly fall detection using wireless communication and artificial intelligence techniques," *Measurement*, 226, 114186, 2024, doi: <https://doi.org/10.1016/j.measurement.2024.114186>.
- [4] T. Xu, H. Se, and J. Liu, "A fusion fall detection algorithm combining threshold-based method and convolutional neural network," *Microprocessors and Microsystems*, Vol. 82, April 2021, 103828, doi: <https://doi.org/10.1016/j.micpro.2021.103828>.
- [5] K. Singh, A. Rajput, and S. Sharma, "Human fall detection using machine learning methods: A survey," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 5, no. 1, pp. 161–180, 2020, doi: <https://doi.org/10.33889/IJMEMS.2020.5.1.014>.

- [6] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, "Latest research trends in fall detection and prevention using machine learning: A systematic review," *Sensors*, vol. 21, no. 15. MDPI AG, Aug. 01, 2021. doi: <https://doi.org/10.3390/s21155134>.
- [7] A. Raza, M.H. Yousaf, W. Ahmad, and S. A. Velastin, "Human fall detection using pose estimation: From traditional machine learning to vision transformers," *Engineering Applications of Artificial Intelligence*, 143, 2025, 109809, doi: <https://doi.org/10.1016/j.engappai.2024.109809>.
- [8] Ch. Gangadhar, P. P. Roy, R. D. Kumar, J. V. N. Ramesh, S. Ravikanth, and N. Akhila, "Wearable sensor-based fall detection for elderly care using ensemble machine learning techniques," *Measurement: Sensors*, 39, 2025, 101870, doi: <https://doi.org/10.1016/j.measen.2025.101870>.
- [9] X. Li, T. Li, Y. Chen, X. Hu, Z. Luo, and S. Xu, "An effective deep learning framework for fall detection: Model development and study design," *Journal of Medical Internet Research*, 2024 Aug 5;26:e56750. doi: <https://doi.org/10.2196/56750>. PMID: 39102661.
- [10] T. Xu, H. Se, and J. Liu, "A two-step fall detection algorithm combining threshold-based method and convolutional neural network," *Metrology and Measurement Systems*, vol. 28, no. 1, pp. 23–40, 2021, doi: <https://doi.org/10.24425/mms.2021.135999>.
- [11] E. Rassekh and L. Snidaro, "Survey on data fusion approaches for fall-detection," *Information Fusion*. 2024 Sep 16;114:102659, doi: <https://doi.org/10.1016/j.inffus.2024.102696>.
- [12] L. Palmerini, J. Klenk, C. Becker, and L. Chiari, "Accelerometer-based fall detection using machine learning: Training and testing on real-world falls," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–15, Nov. 2020, doi: <https://doi.org/10.3390/s20226479>.
- [13] M. Alloghani, D. Al-Jumeily, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science," 2020, pp. 3–21. doi: https://doi.org/10.1007/978-3-030-22475-2_1.
- [14] N. Zurbuchen, A. Wilde, and P. Bruegger, "A machine learning multi-class approach for fall detection systems based on wearable sensors with a study on sampling rates selection," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–23, Feb. 2021, doi: <https://doi.org/10.3390/s21030938>.
- [15] P. Vallabh, R. Malekian, N. Ye, and D. Capeska Bogatinoska, "Fall Detection Using Machine Learning Algorithms," *24th IEEE International Conference on Software, Telecommunications and Computer Networks*, Croatia, 2016, doi: <https://doi.org/10.1109/SOFTCOM.2016.7772142>.
- [16] X. Xi, M. Tang, S. M. Miran, and Z. Luo, "Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable sEMG sensors," *Sensors (Switzerland)*, vol. 17, no. 6, Jun. 2017, doi: <https://doi.org/10.3390/s17061229>.
- [17] E. Escobar-Linero, F. Luna-Perejón, L. Muñoz-Saavedra, J. L. Sevillano, and M. Domínguez-Morales, "On the feature extraction process in machine learning. An experimental study about guided versus non-guided process in falling detection systems," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105170, Sep. 2022, doi: <https://doi.org/10.1016/j.engappai.2022.105170>.
- [18] O. Banos, J. M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors (Switzerland)*, vol. 14, no. 4, pp. 6474–6499, Apr. 2014, doi: <https://doi.org/10.3390/s140406474>.
- [19] S. Pant, J. Kim, and S. Lee, "A fall detection technique using features from multiple sliding windows," *Smart Media Journal*, vol. 7, no. 4, 2018.
- [20] A. Dehghani, O. Sarbishei, T. Glatard, and E. Shihab, "A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors," *Sensors (Switzerland)*, vol. 19, no. 22, Nov. 2019, doi: <https://doi.org/10.3390/s19225026>.
- [21] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors (Switzerland)*, vol. 17, no. 1, Jan. 2017, doi: <https://doi.org/10.3390/s17010198>.
- [22] R. Das, "Accelerated Safety: Revitalizing ADXL345 for Enhanced IoT-enabled Fall Detection," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*. 2023 Dec;11(12), doi: <https://doi.org/10.22214/ijraset.2023.57503>.
- [23] J. Zhang, J. Li, and W. Wang, "A class-imbalanced deep learning fall detection algorithm using wearable sensors," *Sensors*, vol. 21, no. 19, Oct. 2021, doi: <https://doi.org/10.3390/s21196511>.
- [24] S. Kocaoglu and Y. Guven, "Elderly Fall Detection and Fall Direction Decetion via Machine Learning," *Hittite Jorunal of Science and Engineering*, vol. 8, no 3, pp. 197-205, 2021. doi: <https://doi.org/10.17350/HJSE19030000230>.