Journal of Information Systems Engineering and Management
2025, 10(7s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Towards Safer Aging: A Hybrid KNN Model for Pre-Impact Fall Detection Enhanced by Class Balancing

R Ranjith[1], Anju S Pillai[2*], Krishna Priya R[3], Alagan Anpalagan[4]
[1,2]Department of Electrical and Electronics Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India.
[3]Research and Consultancy Department, University of Technology and Applied Sciences, Musandam, Khasab, Governorate of Musandam, Sultanate of Oman.
[4]Department of Electrical and Computer Engineering, Toronto Metropolitan University, Canada.
* Corresponding Author: s_anju@cb.amrita.edu

ARTICLE INFO

Received: 12Oct 2024
Revised: 13Dec 2024
Accepted: 25Dec 2024

ABSTRACT

Falls among the elderly are a significant public health issue, often resulting in serious injuries or even fatalities. Detecting falls before they happen (pre-impact) can enable timely interventions and help minimize the severity of injuries. This research focuses on creating an effective pre-impact fall detection system for elderly individuals by utilizing class balancing techniques and a hybrid k-nearest neighbors (KNN) approach. To address the challenges posed by imbalanced datasets, methods like SMOTE, ADASYN, and RUS were employed, while the hybrid KNN model was used for accurate classification. The system was tested using the KFall dataset, which includes sensor data from accelerometers and gyroscopes. The hybrid KNN model demonstrated an impressive 99.8% accuracy in detecting pre-impact falls, surpassing traditional KNN and other standard methods. Moreover, the use of class balancing significantly enhanced the detection of minority classes. This study highlights the potential of combining class balancing techniques with hybrid KNN models for reliable pre-impact fall detection. These findings could pave the way for developing wearable devices aimed at preventing falls and improving safety for elderly individuals.

Keywords: Class balancing, Fall detection systems, Hybrid KNN model, Pre-impact fall detection, Random under sampling (RUS), Synthetic minority oversampling techniques (SMOTE)

INTRODUCTION

A fall is an event that causes a person to unintentionally fall on the floor or another lower level. Such incidents can lead to injuries, chronic pain, or disabilities. According to statistics from the World Health Organization (WHO), around 6,84,000 fatal falls occur each year (World Health Organization, 2024). Age is a significant factor contributing to falls, with approximately 3 million older adults visiting hospital emergency departments annually due to falls (Centers for Disease Control and Prevention, 2024). If a fall is not promptly attended to, it can result in severe medical complications. Injuries from falls can range from moderate to severe, affecting the individual's normal life. Therefore, identifying falls quickly and providing timely assistance can reduce the severity of injuries and speed up recovery. A serious consequence of fall is a situation known as 'long lie', inability to get after the event of fall. This would increase the injury or lead to loss of mobility or sometimes death (Blackburn et al., 2022). In these kinds of situations, Fall Detection Systems (FDS) would play a crucial role to enable timely medical treatment (Yu et al., 2023).

Fall detection systems involves different techniques to monitor the human continuously. Such techniques include the usage of wearable devices, video surveillance, ambient sensors, etc., Wearable devices employ sensors such as accelerometers, gyroscopes and magnetometers to constantly monitor the movement of an individual and draw an inference out of the data. Video surveillance involves usage of camera to continuously monitor the movement of the individual within an environment and employ computer vision-based techniques to identify fall event. Usage of ambient sensors to detect falls involves different methods such as floor pressure sensors, radar sensors, etc, within a room where the person is monitored for fall (Wang et al., 2023). Even though these methods are non-invasive in terms of signal measurements, video surveillance and ambient sensors would seriously affect the privacy of the person being monitored. Wearable devices are a better choice for FDS because it just monitors the movement of the person; it does not consider of the environment in which the person is present. Unlike video surveillance and ambient sensor-based methods, the person need not be always present in a confined environment to monitor the fall. The person can easily carry the wearable device and is independent to move to any place while continuously being monitored for the occurrence of the fall.

Research in FDS consists of two streams: post-fall identification and pre-impact fall identification. The former would help to minimize the duration of long lie by initiating appropriate calls to family members and medical personnel. While the later would play a crucial role in the minimization of injuries. Pre-impact fall detection can be used for identification of the fall event before the actual occurrence of fall. So suitable protection mechanism like airbags can be inflated around the person where he would get the direct hit. With the help of protection mechanisms and pre-impact fall identification serious injuries like bone fractures, joint dislocations can be minimized to greater extent possible (Liu et al., 2021).

This paper explores the possibilities of identifying the pre-impact fall event based on machine learning (ML) approach. Standard dataset like KFall (Yu et al., 2021), provides comprehensive sensor data recorded for different types of activities and fall. The dataset also includes time stamp to indicate the start of fall. One of the issues faced is, duration of pre-impact fall is very less when compared to the duration of other activities and post-fall. As a result, there exists an inherent class imbalance which needs to be addressed for training the suitable ML model for better classification. Over the years, several over sampling techniques have evolved like random oversampling, synthetic minority oversampling techniques (SMOTE) and its variants, adaptive synthetic sampling (ADASYN) to increase the number of samples in the minority class. There are several under sampling techniques like random under sampling (RUS) to under sample the majority class (Hasib et al., 2020). This paper analyses different resampling techniques that would enable the ML model to learn the pattern of data and classify them in a better way.

The primary contributions of this paper are Comparative analysis of resampling approaches, including oversampling, and a hybrid approach that combines both under sampling and oversampling to achieve class balance, performance evaluation of resampled data to assess the improvements in the machine learning model's performance when utilizing resampled data and a hybrid machine learning techniques that enhance the classification of pre-impact falls, particularly in scenarios with class imbalance.

LITERATURE REVIEW

This section examines relevant literature, highlighting significant findings, and methodologies. To improve the quality of life of elder people, various fall detection systems have been developed. The system that aids the elderly people from falling can be categorised as fall detection, fall prevention and fall protection systems. FDS involves identifying the fall event by means of pattern recognition. The patterns include ADL and different fall events. Fall prevention system can avoid the risk of fall by suggesting regular health check-ups, exercises and clothing that would prevent the occurrence of fall. It also involves generating warning signals when the system predicts the possibility of a fall. Fall protection system involves providing timely medical services after the person falls, actuating suitable protecting air bags to minimize the injury and providing suggestions to the person to avoid the falls (Purwar & Chawla, 2024).

There are different approaches in the implementation of the FDS namely computer vision, ambient sensor and wearable device-based systems (Mrozek et al., 2020). Computer vision-based FDS uses single or multiple cameras to monitor individuals, with multiple cameras preferred for capturing diverse poses. Deep learning techniques enhance fall detection by predicting falls in various orientations. Depth sensors like Kinect address challenges under low light, enabling pose estimation and fall detection even in dim conditions. Classification accuracy ranges from 95.8% (Kinect) to 99.73% (camera-based systems) (Vishnu et al., 2021, Alam et al., 2022, Chen et al., 2020, Mansoor et al., 2022, Saidin & Shukor, 2020). Ambient sensor-based FDS uses technologies like capacitive floor sensors, 2D LiDAR, millimetre waves, and radar to monitor movements in indoor environments. Capacitive sensors track real-time movements with minimal calibration, while LiDAR and radar systems rely on data-driven methods, such as machine learning and deep learning, to detect falls with high accuracy (>98.5%). These systems maintain privacy and are unaffected by light conditions but are limited to confined spaces. To overcome this, wearable devices provide continuous monitoring across various environments, ensuring no fall events are missed (Faulkner et al., 2020, Miawarni et al., 2020, Wang et al., 2020, Wang et al., 2021, Shastri et al., 2022). Wearable FDS use kinematic sensors (accelerometers, gyroscopes) and physiological sensors (ECG, EEG, etc.), commonly found in devices like smartphones and smartwatches. Custom devices can also combine these sensors to monitor falls effectively. Techniques like threshold-based, machine learning, and deep learning approaches are employed for detection, often using datasets like SisFall, WEDA fall, UP Fall or MobiFall for training. Once a fall is detected, the system can alert emergency services and family members (Liu et al., 2023, Zhang et al., 2021, Sharma & Gupta, 2022, Patel & Shah, 2021, Ahmad & Khan, 2020).

The fall of a human body can be classified into three phases in time domain namely, pre-impact fall, fall and post-fall. Detection of the pre-impact fall is beneficial as this phase occurs before the fall event. Pre-impact fall identification requires precise time information for classifying the three phases mentioned before. KFall dataset is a comprehensive dataset that contains the sensor data along with the time information for the fall related events. The data recorded covers 21 activities of daily living and 15 simulated falls (Yu et al., 2021). The data was recorded using custom hardware containing kinematic sensors. The time labels are also based on the video recorded simultaneously along with the sensor data. However there exists an intrinsic class imbalance among the labelled classes in the dataset. Training a classification model requires a balanced dataset or else the majority class would influence the result more which might be misleading. So, balancing the classes before training the ML model is essential.

A straightforward method for balancing datasets is manual under-sampling or over-sampling, where data from the majority class is reduced, or data from the minority class is duplicated. While easy, under-sampling may lower model performance by removing useful data, whereas over-sampling can extend training time without affecting accuracy. More advanced methods, such as SMOTE, ADASYN, etc, improve balance by using statistical techniques to generate synthetic data. Another common approach is random sampling, which removes majority samples or replicates minority samples to create a more balanced dataset, often resulting in better model performance (Wang & Lee, 2021). This paper deals about analysing the performance of different classification model by balancing the classes using hybrid technique.

METHODS

The technical steps involved in the proposed method is shown in Figure 1. More details of the steps and analysis is explained in the following sections.

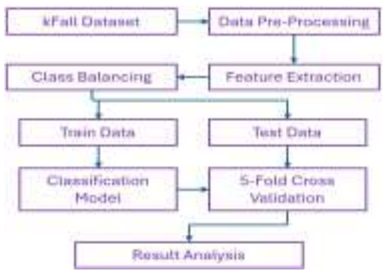

Figure 1. Technical steps – proposed method

1. Dataset

Classification models require data for getting trained. Before the deployment into hardware, model training can be performed with available standard datasets. KFall dataset was chosen as it contains sensor data collected form 32 individuals while performing different tasks. The data includes measurements from accelerometer, gyroscope and magnetometer sensors sampled at 100 Hz. The tasks performed by the individuals include 21 types of ADLs and 19 types of simulated falls (Yu et al., 2021). In total the dataset comprises of 5,075 files containing sensor data for ADL and fall motions. Each file contains 9 features from the sensors mentioned above with time stamp and frame counter columns. The main advantage of KFall dataset is that it contains labels indicating the change of states especially for fall motions. It clearly segregates the states of fall onset, pre-impact fall and post fall in each fall types.

2. Pre-processing

The sensor data is subjected to a moving averaging filter to remove the noise components. A simple moving average algorithm (Wu & Yang, 2022) to smoothen the observations in time series data with a moving window of fixed size is shown in (1).

$$\hat{x_t} = \frac{1}{2n+1}\sum_{i=-n}^{n} x_{i+t} \qquad (1)$$

In equation (1), $\hat{x_t}$ is the predicted value of x_t, the true value. The fraction represents the window, and the variable n represents the window size. The sensors used for recording the user's movement provides output in different ranges i.e. three-axis accelerometer (±16G), a three-axis gyroscope (±2,000◦/s), and a three-axis magnetometer (±16G) (Yu et al., 2021). To rescale the sensor value for data standardisation, scaler-fit transform (Kim & Park, 2021) of scikit learn library in Python has been used.

3. Feature extraction and Class Labelling

The data can be categorised into ADL, pre-fall, fall and post-fall classes. Sensor data with respect to 21 types of ADLs can be made as a single class "ADL". Whereas the data with respect to 19 types of fall motions contains 3 sequential phases. The phases are pre-fall, falling and post-fall as shown in the Figure 2 (Yu et al., 2021). Pre-fall phase contains the data recorded till the fall-onset time instance labelled as "Pre-Fall" class. As shown in the Figure 2 it is a longer phase and contains large number of data points. Data recorded after fall-impact time instance can be labelled as "Post-Fall" class. Like pre-fall class, post-fall class also contains large number of data points. The between Pre-fall and Post-fall class is labelled as "Fall" class. Its duration is small and hence less data points compared to other classes.

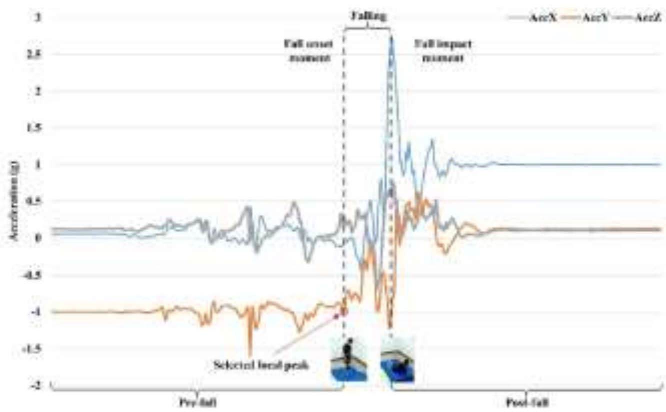

Figure 2. Labelling of the sensor data [7]

The features were extracted from the labelled data using window method. Statistical features such as mean, standard deviation, root mean square (RMS), zero crossing rate (ZCR), absolute difference and first 5-FFT coefficients were determined for all the nine sensor features (Huang & Li, 2023).

- ZCR is the sample count whose value are over the mean of window (Huang & Li, 2023).
- Absolute difference $\sum \frac{|x_i - \mu|}{n}$, where x_i represent each individual sample value, μ represent the mean of the sample values and n is the total number of samples (Huang & Li, 2023).
- First 5-FFT coefficients represent the first five frequency domain coefficients computed using FFT (Huang & Li, 2023).

4. Class Balance

The ADL class contains 2.8 million data points and fall motion contains 1.78 million data points in total as shown in Figure 3. Within the fall motion, pre-fall and post-fall classes contain approximately 0.7 million data points each whereas the fall class contains 0.17 million data points.

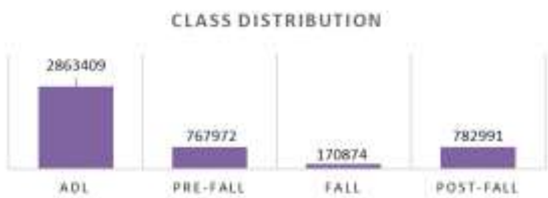

Figure 3. Class Imbalance in KFall Dataset

To balance the classes, hybrid balancing technique is used with bottom-top approach. Class Balancing-1 (CB1) was applied to Pre-fall and Post-fall class. CB2 was applied to balance fall class with pre-fall/post-fall class. The combination of CB1 and CB2 created a balanced fall motion class. Finally, CB3 was applied to ADL class to balance with fall motion class as shown in Figure 4.

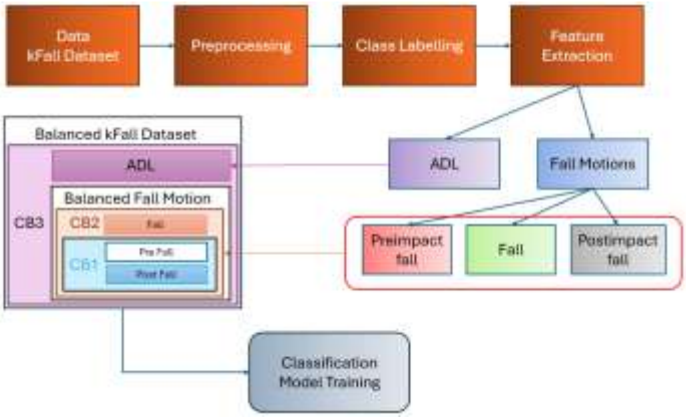

Figure 4. Class Balancing Approach to Train a Classification model

a. Random Under-Sampling (RUS): A simple technique used for discarding random number of samples from majority class (Lee & Choi, 2023). This technique is used for CB1. Post-fall is a majority class when compared with Pre-fall class. As the post-fall class contains large number of data points recorded after the user has fallen on the floor, removing random samples would aid in class balancing.

Oversampling (CB2) was done to the minority 'Fall' class to balance with Pre-fall/Post-fall class. Several techniques were used to determine a suitable one that would aid in the classification of the 3 fall motion classes.

b. Synthetic Minority Over-Sampling Technique (SMOTE): Synthetic samples are created by interpolation of the k nearest neighbors around each minority sample (Lee & Choi, 2023, Kumar & Sharma, 2022).
c. Borderline SMOTE: SMOTE to borderline samples that are misclassified by their nearest neighbors (Lee & Choi, 2023).
d. Support Vector Machine (SVM) SMOTE: SVM classifier is used for predicting new instances of minority class along the borderline (Lee & Choi, 2023).
e. Adaptive Synthetic (ADASYN): Generates synthetic samples for the minority class that are difficult to be learnt for classification using weighted distribution method (Lee & Choi, 2023).

The newly augmented data with original and resampled data was subjected to classification models and SVM-SMOTE technique-based data was exhibiting better performance in classification. The comparison of performance metrics is discussed in the results section.

Under Sampling (CB3) was done to balance ADL class with Fall motion class. It was more of a manual approach in discarding the data from multiple trials for the ADL class.

5. Classification Models

i) k-Nearest Neighbors

KNN technique classifies a new data X based on the majority votes of the similar data samples in the training dataset. K data points around the new data are identified. The Euclidean distance as in equation (2) is calculated between the new data and the samples in the training dataset. The class is assigned to the new data to which most of its closest neighbors reside as in (3) (Singh & Rana, 2021, Yu & Xiong, 2021). Eighty percent of the datasets was allocated for training the model and the remaining was used to test the model. The training process was done using classification learner app of MATLAB.

$$d(X_i, X_j) = \sqrt{\sum_{p=1}^{n} (x_{ip} - x_{jp})^2} \qquad (2)$$

$$R_k^X = \{X \in R^n, d(X, X_i) \leq \ d(X, X_k)\} \qquad (3)$$

Depending on the number of neighbors considered, there are different KNN variants. With the number of neighbors set to 1, fine KNN makes finely detailed distinction ns between the classes. With the number of neighbors set to 10, medium KNN defines medium distinctions between the classes whereas cosine KNN uses cosine distance metric to define the medium distinctions. Similarly for coarse KNN, the number of neighbors is set to 100. Ensemble subspace KNN trains multiple KNN models with combinations of different subset of features (Li & Zhao, 2020).

ii) Support Vectors Machines

The SVM model uses hyperplane optimization to distinguish target classes with maximum margins. A new dimensional space is defined upon which the input data gets mapped. Between the points of the different classes, the largest possible margin is used for determining the decision boundary. Finally, the SVM model classifies the testing data according to the decision boundary (Sun & Yu, 2022, Yu & Xiong, 2021). SVM is primarily used for binary problems. When handling more than two classes, SVM is extended to a multi-class approach by combining multiple binary classifiers. The input data represented by vector X is mapped into a higher-dimensional space Y using a transformation function $\varphi(X)$. The resulting decision function is expressed as (Fang & Li, 2021):

$$F = \ w^T \varphi(X) + b \qquad (4)$$

where w is a weight vector, and b represents a bias term. These parameters are learned from the training data to establish a boundary between classes. To allow for flexibility in separating classes, a slack variable ξ is introduced. The kernel function (Xi, Yj) is a crucial component in SVM, enabling the model to perform complex classifications without explicitly transforming the data. Gaussian kernel measures the similarity between data points. This kernel is defined as (Fang & Li, 2021):

$$k(X_i, Y_j) = \ e^{\left(-\frac{\left\|X_i - Y_j\right\|^2}{2\sigma^2}\right)} \qquad (5)$$

The Gaussian kernel allows SVM to draw non-linear boundaries, making it suitable for classifying data with intricate patterns, such as in fault detection tasks.

iii) Proposed Hybrid Classification Model

Classification Learner App of MATLAB was used to identify suitable classification technique for the FDS. Initial comparison of different models with imbalanced data proved that Fine-KNN performed better than other models based on evaluation criteria mentioned in the next section. However, the data was subjected to hybrid resampling to achieve class balance. The usage of hybrid technique led to the proposed hybrid architecture of the classification approach as shown in the Figure 5. The class balance was achieved in two major steps. Initial balance was achieved among the fall motion sub classes followed by the balance between ADL and fall motion class.

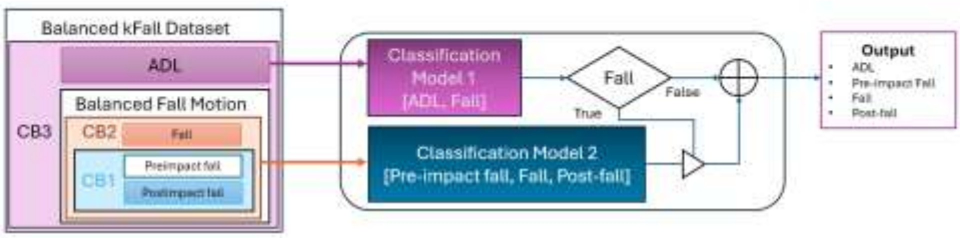

Figure 5. Hybrid model trained with Balanced data

The proposed hybrid model contains two classification models working in parallel with the supplied data. The 'Classification model 1' does binary classification of ADL and fall motions. Whereas the 'Classification model 2' does the classification of fall motion classes. If the output belongs to ADL class, the result is directed for the final output of the hybrid model. If the output belongs to the fall motion class, then the output of the model 2 is directed to the final output of the hybrid model. Thus, the hybrid model classifies the input data into four possible classes. The evaluation criteria and the results of various class balancing techniques and classification models are presented in the following sections.

6. Evaluation Criteria

Experimentation on the dataset was done in four ways. Initial training of the model was carried out with the class imbalance present in the data. Second experimentation was done with the balanced fall classes leading to model 2. Third experimentation was done with balanced data of ADL and fall classes leading to model 1. Fourth experimentation was done with a hybrid model, made of model 1 and model 2. In each case, 80% of the data was used for training and 20% of the data was used for testing the machine learning models. Performance of the models were compared using sensitivity, specificity, precision and accuracy (Yu et al., 2021). These parameters were calculated with the results of classification which includes TP (True positive), FP (False positive), TN (True negative) and FN (False negative) as shown in equations 6 to 9.

$$Sensitivity = \frac{TP}{TP + FN} \quad (6)$$

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (9)$$

RESULTS

1. Classification Performance with Imbalanced Data

Various machine learning algorithms were trained and tested for classification of the four classes with imbalanced data as input using MATLAB Classification Learner app. Based on the test, confusion matrix parameters were computed. Table-1 summarises the results.

Table 1. Performance comparison of Classification models with class imbalance

		Sensitivity	Specificity	Accuracy	Precision
Fine KNN	ADL	0.996122	0.990926	0.994199	0.994675
	Pre-Fall	0.989633	0.997374	0.99623	0.984869
	Fall	0.914691	0.997873	0.994300	0.950738
	Post-Fall	0.986425	0.997021	0.995115	0.986425
Weighted KNN	ADL	0.99611	0.981917	0.990840	0.989385
	Pre-Fall	0.986197	0.995941	0.994504	0.976760
	Fall	0.874704	0.997447	0.992163	0.939086
	Post-Fall	0.973003	0.997390	0.992977	0.988006
Medium KNN	ADL	0.994331	0.977383	0.988012	0.986659
	Pre-Fall	0.996165	0.995061	0.995225	0.972326
	Fall	0.834286	0.998514	0.991507	0.961581
	Post-Fall	0.975972	0.998162	0.994128	0.991598
Medium Gaussian SVM	ADL	0.987722	0.953839	0.975089	0.97297
	Pre-Fall	0.955486	0.990457	0.985269	0.94577
	Fall	0.699524	0.996307	0.983644	0.894096
	Post-Fall	0.970049	0.99548	0.990857	0.979463
Coarse KNN	ADL	0.98072	0.950188	0.969339	0.970691
	Pre-Fall	0.983016	0.979410	0.979945	0.892661
	Fall	0.487619	0.998153	0.976369	0.921692
	Post-Fall	0.95786	0.992351	0.986081	0.965311

Sensitivity ensures true positives is as true as possible, which is vital in fields like healthcare where missing a real case could be dangerous. Specificity is about reducing false alarms, which helps to avoid unnecessary actions and costs by correctly identifying true negatives. Precision measures the accuracy of positive predictions, which is crucial in scenarios were acting on a false positive is costly or inconvenient. From Figure 6, it is evident that models were unable to deliver a common performance across all the classes due to the presence of imbalance. The accuracy of the models cannot be considered as a metric for comparison in the presence of class imbalance. Table 1 summarizes the model performance with other metrics.

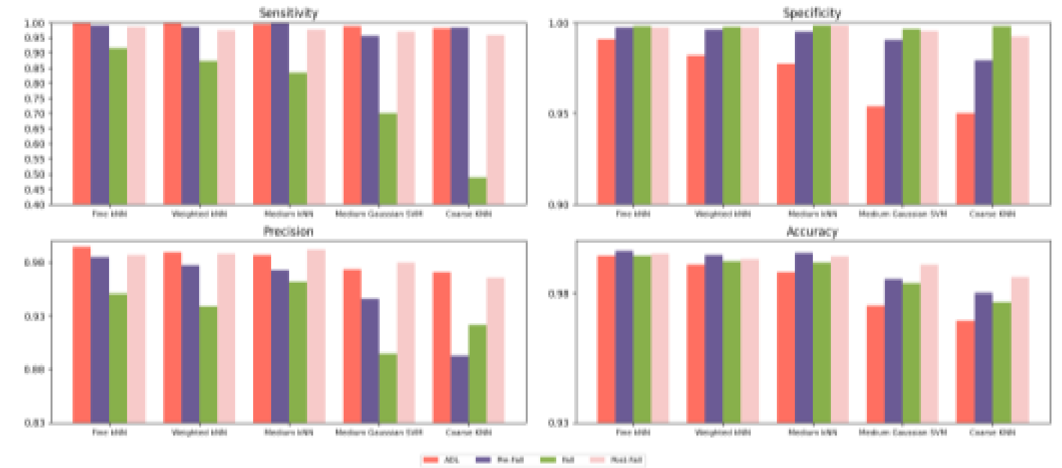

Figure 6. Performance comparison of Machine Learning Models with Class Imbalanced Data

From the Table 1, it is observed that, Fine-KNN technique provides better classification results compared to other techniques. On a close observation of KNN, sensitivity varies across minority classes. The class imbalance leads to difference in the sensitivity in the result of classification. Accuracy cannot be used as a metric for comparison of model performance due to the class imbalance present in the training data. So, class balancing was done to balance the classes.

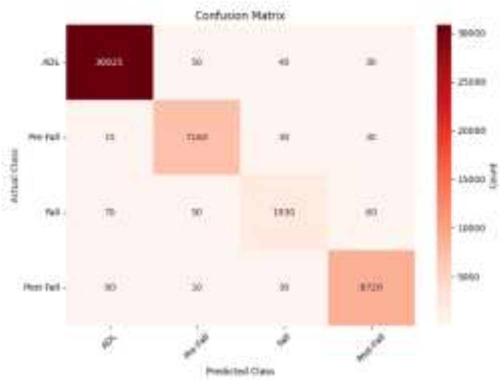

Figure 7. Confusion matrix of Fine-KNN model with class imbalance

Above figure illustrates the confusion matrix for the Fine-KNN model. The matrix clearly explains the data imbalance of the classes.

2. Identification of Resampling techniques for Fall motion classes (CB2)

Class balancing was done in 3 stages CB1, CB2 and CB3 as shown in Figure_4. CB1 and CB3 were based on RUS to balance the majority class with minority class. Whereas the CB2 is based on Over Sampling to balance the three fall classes. Figure 8 shows the performance comparison of the over sampling techniques, SMOTE, Borderline-SMOTE, SVM-SMOTE and ADASYN.

Table 2. Performance Comparison of Class Balancing within Fall class

		Sensitivity	Specificity	Accuracy	Precision
SMOTE	Pre-Fall	0.998006	0.998773	0.998517	0.997547
	Fall	0.998006	0.997316	0.997546	0.99465
	Post-Fall	0.995245	0.99954	0.998108	0.999076

Borderline SMOTE	Pre-Fall	0.997698	0.998204	0.998033	0.996475
	Fall	0.996977	0.996164	0.996429	0.992085
	Post-Fall	0.993252	0.999609	0.997464	0.999229
SVM SMOTE	Pre-Fall	0.998309	0.999014	0.998779	0.998028
	Fall	0.998873	0.997112	0.997699	0.994254
	Post-Fall	0.994929	0.999929	0.998263	0.999858
ADASYN	Pre-Fall	0.998003	0.999386	0.998925	0.99877
	Fall	0.999078	0.995853	0.996928	0.991767
	Post-Fall	0.992934	0.99977	0.997491	0.999536

For each combination of RUS with different over sampling techniques, the performance of class balance was verified by subjecting the augmented data to classification models. Fine-KNN model was effective in classifying the data based on the SMOTE and Borderline-SMOTE methods. Whereas Ensemble method with preset as Subspace KNN was effective for the data augmented using SVM—SMOTE and ADASYN methods. The results are listed in the table 2 and the corresponding plot is shown in Figure 8.

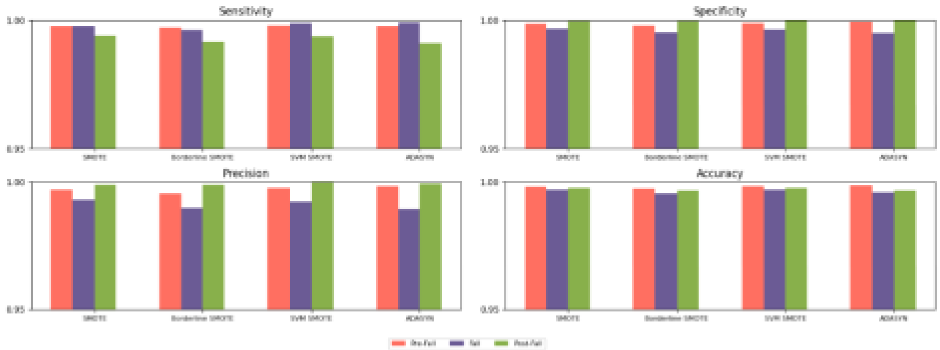

Figure 8. Performance Comparison of Class Balancing Techniques for CB2

Based on Sensitivity and Specificity, SVM-SMOTE and ADASYN methods are exhibiting similar performance and better than the other methods. Based on Precision and Accuracy, SVM-SMOTE was chosen as suitable method for CB2 as shown in Figure 4. The model Ensemble method with preset as Subspace KNN was chosen as the classification model 2 as shown in Figure 5. Figure 9, shows the confusion matrix of the selected model.

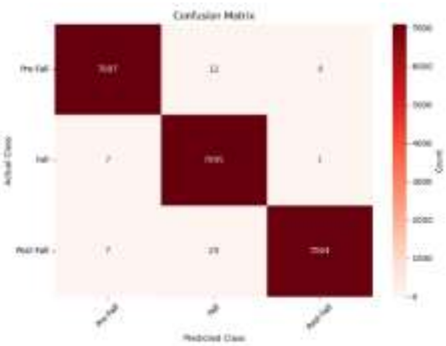

Figure 9. Confusion matrix of Ensemble Model – Classification Model 2

The trained model can then be used as Classification Model 2 in the hybrid system. The Classification Model 2 performs slightly better than the TinyFallNet model in pre-fall detection. Where the overall accuracy of TinyFallNet is 98% with sensitivity of 86.67% and a specificity of 97.97% (Zhang & Chen, 2020).

3. Identification of Class Balancing technique for Fall and ADL classes (CB3)

For classifying the ADL and fall motions, a binary classification model 1 as in Figure 5 was required. This model, upon predicting fall motions would enable the classification model 2 to predict the type of fall motion. For training the classification model 1, all the fall motions were categorized as 'Fall' class. As there are more types of ADL than fall, there exists a class imbalance between ADL and fall motions.

There are several trials for certain ADLs. Reduction in trial count was performed to balance the class. The binary class data used for training and testing the classification model 1 produced the results as shown in Figure 10.

Table 3. Performance Comparison of Models for Classification Model 1

		Sensitivity	Specificity	Accuracy	Precision
Fine KNN	ADL	0.99807	0.997035	0.997552	0.997037
	Fall	0.997035	0.99807	0.997552	0.998069
Ensemble Subspace KNN	ADL	0.999082	0.994223	0.996655	0.994262
	Fall	0.994223	0.999082	0.996655	0.999075
Medium KNN	ADL	0.988469	0.991053	0.98976	0.991048
	Fall	0.991053	0.988469	0.98976	0.988476
Cosine KNN	ADL	0.980713	0.990798	0.98575	0.990722
	Fall	0.990798	0.980713	0.98575	0.980869
Medium Gaussian SVM	ADL	0.969896	0.961196	0.965551	0.961605
	Fall	0.961196	0.969896	0.965551	0.969574

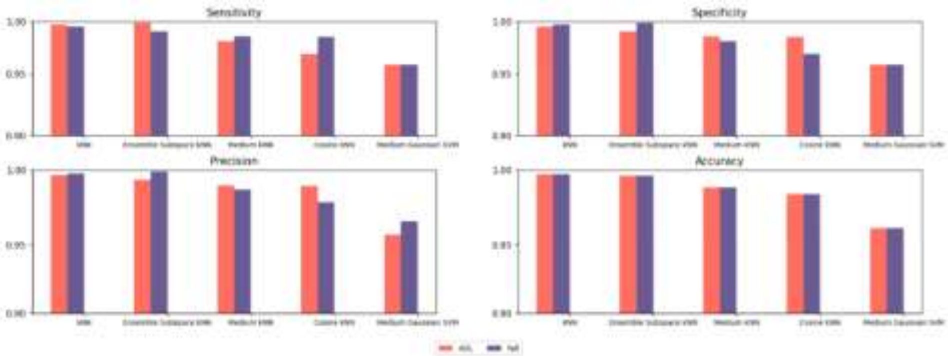

Figure 10. Performance comparison of Classification Models with Balanced ADL and Fall

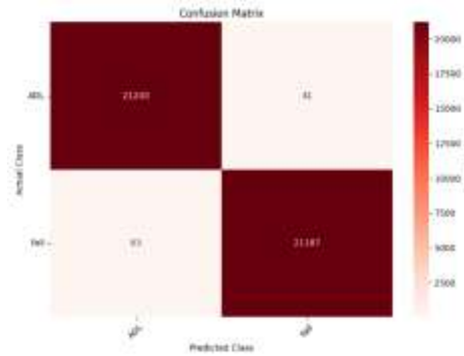

Figure 11. Confusion matrix of KNN Model – Classification Model 1

Fine-KNN model provides better performance for the binary classification. The confusion matrix of the selected model is shown in Figure 11. The output can be either ADL or fall motion. The performance of the model is comparable to results achieved in (Singh & Rana, 2021) where accuracy is 99.84%.

4. Performance evaluation of proposed Hybrid Classification Model

The output ADL class of classification model 1 is directed to the final system output. The output fall motion is used to trigger the classification model 2 for fall motion classification. The output of classification model 2 is directed to the final system output as shown in Figure 5. The hybrid system performs in a way like a single model that predicts single ADL class and 3 fall motion classes. The performance of the hybrid system is shown in Figure 8.

Table 4. Classification Performance of the Hybrid System with Four Class Data

		Sensitivity	Specificity	Accuracy	Precision
Hybrid System	ADL	0.997826	0.995118	0.996883	0.997392
	Pre-Fall	0.99711	0.999843	0.999575	0.998553
	Fall	0.997748	0.998488	0.998441	0.977925
	Post-Fall	0.988654	0.999128	0.997166	0.996189

The hybrid system was trained using the balanced data. As seen in Figure 12, there is a performance improvement compared with the model trained using imbalanced data.

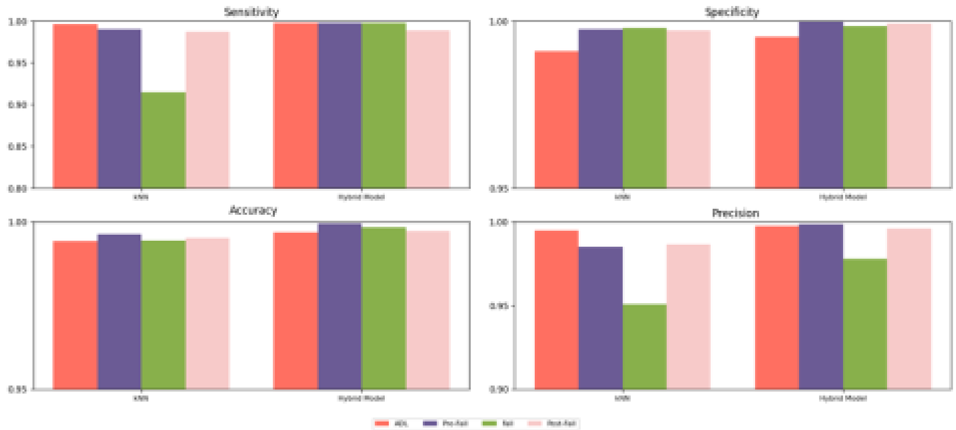

Figure 12. Performance comparison of Hybrid Model with Initial model of KNN

Performance metrics for the minority fall motion classes have improved significantly with the hybrid resampling and hybrid classification model. The hybrid model was tested using data with class imbalance. Table 4 summarizes the performance of model with respect to the four classes.

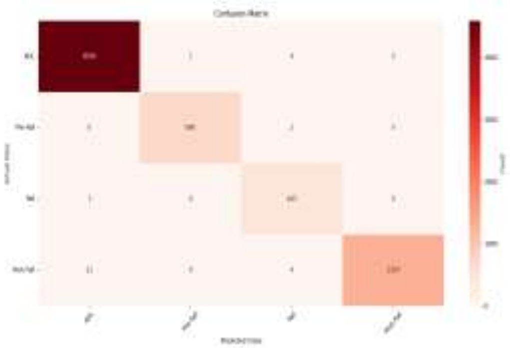

Figure 13. Confusion matrix of the proposed Hybrid Classification Model

Figure 13 illustrates the confusion matrix of the hybrid model when tested using data with class imbalance. The entire process of data pre-processing, class balancing and training the classification model could be done using a cloud server. The trained model can be deployed into an Edge Computing Device like a Single Board Computer (SBC). The SBC is paired to the user wearing the sensing module to receive the live data. Upon processing the SBC can generate necessary signalling to the injury mitigation system like airbag. The SBC also can receive Over the Air (OTA) update from the cloud server whenever the model gets updated in the cloud as shown in Figure 14. Thus, ensuring the user monitored with the latest trained model.

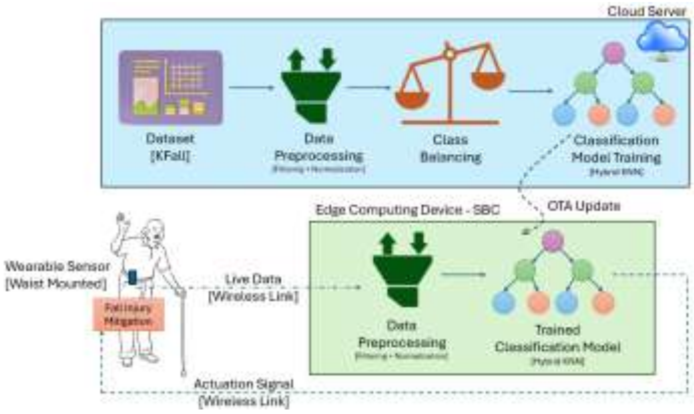

Figure 14. System implementation

DISCUSSION

The proposed study successfully addresses the critical challenge of pre-impact fall detection for elderly individuals, a pressing public health concern with significant implications for safety and injury prevention. By integrating class balancing techniques, such as SVM-SMOTE, with a hybrid in model, the proposed system effectively tackles the limitations of imbalanced datasets. This approach not only enhances the overall accuracy of fall detection but also ensures better detection rates for minority classes, which are often overlooked in conventional methods. The system achieved an accuracy of 99.8%, outperforming traditional and other baseline models, making it a promising solution for real-world applications. The KFall dataset, comprising sensor data from accelerometers and gyroscopes, provided a comprehensive and realistic foundation for training and testing the model. The results highlight the robustness and reliability of the proposed approach in identifying pre-impact falls, which is a critical window for initiating preventive interventions. Such timely detection can enable protective measures, such as activating wearable airbags, significantly reducing the risk of severe injuries or fatalities.

This research not only demonstrates the potential of combining advanced data balancing methods with machine learning techniques but also sets the stage for the development of wearable devices tailored to elderly care. These devices could serve as proactive tools for fall prevention, enhancing the quality of life for elderly individuals and reducing the burden on caregivers and healthcare systems. Future work could expand on this foundation by incorporating additional sensor data, exploring real-time deployment in wearable technologies, and evaluating the system's performance in diverse and uncontrolled environments. This would further refine and solidify its applicability in ensuring the safety and well-being of elderly populations.

REFRENCES

[1] World Health Organization. (2024). *Falls*. Retrieved July 2024, from https://www.who.int/news-room/fact-sheets/detail/falls

[2] Centers for Disease Control and Prevention. (2024). *Older adult fall prevention*. Retrieved July 2024, from https://www.cdc.gov/falls/data-research/facts-stats/index.html

[3] Blackburn, J., Ousey, K., Stephenson, J., & Lui, S. (2022). Exploring the impact of experiencing a long lie fall on physical and clinical outcomes in older people requiring an ambulance: A systematic review. *International Emergency Nursing, 62,* 101148. https://doi.org/10.1016/j.ienj.2022.101148

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

[5] Wang, P., Li, Q., Yin, P., Wang, Z., Ling, Y., Gravina, R., & Li, Y. (2023). A convolution neural network approach for fall detection based on adaptive channel selection of UWB radar signals. *Neural Computing and Applications, 35,* 15967–15980. https://doi.org/10.1007/s00521-021-06795-w

[6] Liu, N., Zhang, D., Su, Z., & Wang, T. (2021). Preimpact fall detection for elderly based on fractional domain. Mathematical Problems in Engineering, 2021, 6661034. https://doi.org/10.1155/2021/6661034

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

[8] Hasib, K. M., Iqbal, M. S., Shah, F. M., Al Mahmud, J., Popel, M. H., Showrov, M. I. H., Ahmed, S., & Rahman, O. (2020). A survey of methods for managing the classification and solution of data imbalance problem. Journal of Computer Science, 16(11), 1546–1557. https://doi.org/10.3844/jcssp.2020.1546.1557

[9] Purwar, I., & Chawla, P. (2024). A systematic review on fall detection systems for elderly healthcare. Multimedia Tools and Applications, 83, 43277–43302. https://doi.org/10.1007/s11042-023-17190-z

[10] Mrozek, D., Koczur, A., & Małysiak-Mrozek, B. (2020). Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge. Information Sciences, 537, 132–147. https://doi.org/10.1016/j.ins.2020.05.070

[11] Vishnu, C., Datla, R., Roy, D., Babu, S., & Mohan, C. K. (2021). Human fall detection in surveillance videos using fall motion vector modeling. IEEE Sensors Journal, 21(15), 17162–17170. https://doi.org/10.1109/JSEN.2021.3082180

[12] Alam, E., Sufian, A., Dutta, P., & Leo, M. (2022). Vision-based human fall detection systems using deep learning: A review. Computers in Biology and Medicine, 146, 105626. https://doi.org/10.1016/j.compbiomed.2022.105626

[13] Chen, W., Jiang, Z., Guo, H., & Ni, X. (2020). Fall detection based on key points of human-skeleton using OpenPose. Symmetry, 12, 744. https://doi.org/10.3390/sym12050744

[14] Mansoor, M., Amin, R., Mustafa, Z., et al. (2022). A machine learning approach for non-invasive fall detection using Kinect. Multimedia Tools and Applications, 81, 15491–15519. https://doi.org/10.1007/s11042-022-12113-w

[15] Saidin, N. A., & Shukor, S. A. A. (2020). An analysis of Kinect-based human fall detection system. 2020 IEEE 8th Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia, 220–224. https://doi.org/10.1109/ICSPC50992.2020.9305797

[16] Faulkner, N., Parr, B., Alam, F., Legg, M., & Demidenko, S. (2020). CapLoc: Capacitive sensing floor for device-free localization and fall detection. IEEE Access, 8, 187353–187364. https://doi.org/10.1109/ACCESS.2020.3029971

[17] Miawarni, H., et al. (2020). Fall detection system for elderly based on 2D LiDAR: A preliminary study of fall incident and activities of daily living (ADL) detection. 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 1–5. https://doi.org/10.1109/CENIM51130.2020.9298000

[18] Wang, B., Guo, L., Zhang, H., & Guo, Y.-X. (2020). A millimetre-wave radar-based fall detection method using line kernel convolutional neural network. IEEE Sensors Journal, 20(22), 13364–13370. https://doi.org/10.1109/JSEN.2020.3006918

[19] Wang, Y., Liu, H., Cui, K., Zhou, A., Li, W., & Ma, H. (2021). m-Activity: Accurate and real-time human activity recognition via millimeter wave radar. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 8298–8302. https://doi.org/10.1109/ICASSP39728.2021.9414686

[20] Shastri, A., et al. (2022). A review of millimeter wave device-based localization and device-free sensing technologies and applications. IEEE Communications Surveys & Tutorials, 24(3), 1708–1749. https://doi.org/10.1109/COMST.2022.3177305

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

[22] Ranjith, R., & Pillai, A. S. (2019). Classification of activities of daily living using smartphones. International Journal of Recent Technology and Engineering (IJRTE, 8(3), 2984–2988. https://doi.org/10.35940/ijrte.c4822.098319

[23] Badgujar, S., & Pillai, A. S. (2020). Fall detection for elderly people using machine learning. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–4. https://doi.org/10.1109/ICCCNT49239.2020.9225494

[24] Garimella, K. S., & Pillai, A. S. (2022). Comparison of fall risk prediction of patients using tree algorithms implemented on a wearable belt. 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 88–95. https://doi.org/10.1109/ICOSEC54921.2022.9951918

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

[26] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., ... Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. IEEE Access, 4, 7940–7957. https://doi.org/10.1109/ACCESS.2016.2619719

[27] Yu, X., Koo, B., Jang, J., Kim, Y., & 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. https://doi.org/10.1016/j.measurement.2022.111785

[28] Wang, X., & Wang, C. (2020). Time series data cleaning: A survey. IEEE Access, 8, 1866–1881. https://doi.org/10.1109/ACCESS.2019.2962152

[29] Scikit-learn. (2024). sklearn.preprocessing.StandardScaler. Scikit-learn. Retrieved from https://scikit-learn.org/dev/modules/generated/sklearn.preprocessing.StandardScaler.html

[30] Khushi, M., Khamparia, A., Gupta, D., Khanna, A., Rodrigues, J. J. P. C., & De Albuquerque, V. H. C. (2021). A comparative performance analysis of data resampling methods on imbalance medical data. IEEE Access, 9, 109960–109975. https://doi.org/10.1109/ACCESS.2021.3102399

[31] Son, H., Park, J., Ahn, J., Seo, H., & Kim, H. (2022). A machine learning approach for the classification of falls and activities of daily living in agricultural workers. IEEE Access, 10, 77418–77431. https://doi.org/10.1109/ACCESS.2022.3190618

[32] MathWorks. (2024). Choose a classifier. MATLAB Documentation. Retrieved from https://in.mathworks.com/help/stats/choose-a-classifier.html

[33] Liu, K.-C., Hung, K.-H., Hsieh, C.-Y., Huang, H.-Y., Chan, C.-T., & Tsao, Y. (2022). Deep-learning-based signal enhancement of low-resolution accelerometer for fall detection systems. IEEE Transactions on Cognitive and Developmental Systems, 14(3), 1270–1281. https://doi.org/10.1109/TCDS.2021.3116228

[34] Winston, D. P., et al. (2021). Solar PV's micro crack and hotspots detection technique using NN and SVM. IEEE Access, 9, 127259–127269. https://doi.org/10.1109/ACCESS.2021.3111904

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

[36] Sangapu, S. C., Prasad, K. S. N., Kannan, R. J., et al. (2023). Impact of class imbalance in VeReMi dataset for misbehavior detection in autonomous vehicles. Soft Computing. https://doi.org/10.1007/s00500-023-08003-4

[37] Mamidi, S. S. R., Munaganuri, C. A., Gollapalli, T., Aditya, A. T. V. S., & B, R. C. (2022). Implementation of machine learning algorithms to identify freshness of fruits. In 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT) (pp. 1395–1399). https://doi.org/10.1109/ICICICT54557.2022.9917989