

Article

Not peer-reviewed version

Regressive Machine Learning for Real-Time Monitoring of Bed-Based Patients

[Paul Joseph](#) , Husnain Ali , Daniel Matthew , [Anvin Thomas](#) , [Rejath Jose](#) ^{*} , [Jonathan Mayer](#) , [Molly Bekbolatova](#) , [Timothy Devine](#) , [Milan Toma](#) ^{*}

Posted Date: 17 October 2024

doi: 10.20944/preprints202409.1279.v2

Keywords: Machine Learning; Model; Regressor; Ensemble; Patient Safety; Automated Detection



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Regressive Machine Learning for Real-Time Monitoring of Bed-Based Patients

Paul Joseph ¹, Husnain Ali ², Daniel Matthew ¹, Anvin Thomas ¹, Rejath Jose ¹, Jonathan Mayer ¹, Molly Bekbolatova ¹, Timothy Devine ³ and Milan Toma ^{1,*} 

¹ Department of Osteopathic Manipulative Medicine, College of Osteopathic Medicine, New York Institute of Technology, Old Westbury, New York 11568, United States; pjosep07@nyit.edu; dmatth06@nyit.edu; athoma47@nyit.edu; rjose02@nyit.edu; jmayer03@nyit.edu; mbekbola@nyit.edu

² Department of Clinical Medicine, American University of the Caribbean School of Medicine, 1 University Drive at Jordan Road, Cupecoy, Sint Maarten, Sint Maarten (Dutch Part); ha13ty@outlook.com

³ The Ferrara Center for Patient Safety and Clinical Simulation, New York Institute of Technology, College of Osteopathic Medicine, Department of Osteopathic Manipulative Medicine, Old Westbury, New York 11568, United States; tdevine@nyit.edu

* Correspondence: tomamil@tomamil.com

Abstract: This study introduces an ensemble model designed for real-time monitoring of bedridden patients. The model was developed using a unique dataset, specifically acquired for this study, that captures six typical movements. The dataset was balanced using the Synthetic Minority Over-sampling Technique, resulting in a diverse distribution of movement types. Three models were evaluated: a Decision Tree Regressor, a Gradient Boosting Regressor, and a Bagging Regressor. The Decision Tree Regressor achieved an accuracy of 0.892 and an R2 score of 1.0 on the training dataset, and 0.939 on the test dataset. The Boosting Regressor achieved an accuracy of 0.908 and an R2 score of 0.99 on the training dataset, and 0.943 on the test dataset. The Bagging Regressor was selected due to its superior performance and trade-offs such as computational cost and scalability. It achieved an accuracy of 0.950, an R2 score of 0.996 for the training data, and an R2 score of 0.959 for the test data. The study also employs K-Fold cross-validation and learning curves to validate the robustness of the Bagging Regressor model. The proposed system addresses practical implementation challenges in real-time monitoring, such as data latency and false positives/negatives, and is designed for seamless integration with hospital IT infrastructure. This research demonstrates the potential of machine learning to enhance patient safety in healthcare settings.

Keywords: machine learning; model; regressor; ensemble; patient safety; automated detection

1. Introduction

Falls among bed-bound patients, particularly older adults, remain a persistent issue. The incidence of falls is high, with about one-third of adults aged 65 and older experiencing falls annually, leading to injuries and increased healthcare needs [1]. About 10% of fatal falls in the elderly occur in the hospital. Fatal falls in the elderly, particularly in hospital settings, often occur due to a combination of factors such as gait and balance disorders, cognitive impairment, frailty, deconditioning, and the use of certain medications. For instance, an elderly patient with cognitive impairment might not recognize the risk of getting out of bed unassisted, leading to a fall that could result in serious injury or even death. Falls can be costly, leading to increased treatment and extended hospital stays, usually resulting in 60% higher medical expenses. This issue is especially relevant in both hospital and home care settings, where achieving effective fall prevention strategies is difficult due to the lack of resources and staff education [2].

Preventing falls in bed-bound patients requires systematic approaches, such as identifying individual risk factors and providing tailored interventions. These may include strength and balance training for patients who can perform limited movements, proper use of bed alarms, and positioning techniques to reduce fall risk. Employee involvement and scenario-based training, such as workshops, are also key to successful fall prevention measures [3].

Falls are the leading cause of injury among adults aged ≥ 65 years (i.e., older adults) in the United States [4]. Falls are preventable, and health-care providers can help their older patients reduce

their risk for falls. Screening older patients for fall risk, assessing modifiable risk factors (e.g., use of psychoactive medications or poor gait and balance), and recommending interventions to reduce this risk (e.g., medication management or referral to physical therapy) can prevent older adult falls [4]. Falls can lead to serious consequences such as fractures, head injuries, reduced mobility, premature long-term care admissions, and even death, especially considering that 30% of fallers will fall again [5].

Gait and balance disorders are multifactorial, common causes of falls, especially in older adults with other comorbidities and underlying medical conditions [6]. Cognitive impairment is also associated with an increased risk of falls [6], including impaired visuospatial function, psychomotor speed, executive function, and attention [7]. This can be attributed to neurologic diseases such as Parkinson disease or neurocognitive disorders such as dementia [8]. The current literature does not document sufficient evidence for effective fall prevention intervention in this population [9]. Frailty and deconditioning, due to a mix of factors such as unintentional weight loss, muscle weakness, and low physical activity are associated with falls [8]. Polypharmacy and the use of certain medications, such as opioids, vasodilators, antihistamines, antidepressants, and B-blockers, are associated with increased fall risk [6]. With older populations, medication reconciliation becomes increasingly important as managing multiple comorbidities and medical problems leads to conflicting side effect profiles.

These factors are often exacerbated by each other - the effect of psychotropic medications on gait and balance is known, along with wandering behavior and poor situational awareness associated with impaired cognition, impulsivity, and poor balance control [7]. Nonmedical reasons, such as unsafe home environments and poor footwear, can also contribute [4,5]. Adherence to fall prevention recommendations and referrals is a factor that should be explored further [10].

Today, inpatient and long-term care use multiple observation modalities to prevent falls. These methods include the use of human resources, technological resources, or even both. The most traditional and well-known method is one-to-one observation, wherein hospital staff are assigned to a patient to stay nearby and monitor them at all times [11]. Although a well-used modality, the efficacy of one-to-one depends on the abilities of the staff to both keep a constant watch on the patient and to make clinical predictions regarding what actions or movements may lead to falls. Additionally, one-to-one as a monitoring method is costly and resource consuming [12]. By assigning a member of the hospital staff to constantly monitor a single patient, there are fewer staff on the floor available to do other tasks for other patients. Reducing the number of staff that can serve other patients increases the workload on other staff members and decreases the care said staff can deliver to each individual patient. As the cost of one-to-one is that which pays the hospital staff, it is a costly modality.

As technology has advanced, inpatient settings have been able to implement cameras into patient monitoring [11]. The accessibility of the camera has offered an efficient and less costly method of patient monitoring [13]. With it, a single hospital staff is no longer required to dedicate their entire day to a single patient. As such, there are more staff available to work on the floors and the work can be distributed more effectively. Additionally, a single - or just a few - staff member(s) can be utilized to monitor multiple patients at once. Although they may need to be trained to utilize the technology as best as possible, staff can integrate their own creative and clinical decision-making skills to optimize monitoring. The largest issue with camera monitoring, however, is the inability to take rapid action, making it overall a more passive modality [14]. As staff are monitoring a patient online, if a patient does experience a fall, the monitoring staff is unable to take action themselves and must instead inform other staff on the respective floor to help the patient. Additional costs for camera monitoring are attributed mostly to the initial investment as well as maintenance of camera equipment.

Hence, the proposed model overcomes specific limitations of previous fall detection methods, thereby enhancing its applicability in real-world healthcare settings. Unlike video monitoring methods, which pose privacy issues due to constant surveillance, our model employs sensor technology, ensuring patient privacy. The sensor-based approach not only facilitates automated detection, a significant advancement over video methods, but also reduces the need for healthcare professionals to continuously monitor a screen. Most importantly, while other studies have concentrated solely on fall

prevention, our model broadens the scope by detecting six different types of movements commonly performed by bed-based patients. This detection capability not only improves patient safety but also contributes to more efficient and effective patient care in healthcare settings.

1.1. Related Works

Bed alarms are another prominent modality used to reduce fall risk [11]. Functioning by alerting staff of when a patient is off the bed yet requiring fewer costs and less personnel allocation, the bed alarm is an intuitive tool in fall risk management [15]. By directly alerting staff of an adverse event, it expedites the spread of information and the time to action. However, bed alarms are only effective after an event has already occurred, making alarms another passive modality [16]. Although effective in making patients feel more secure and hastening a call to action, bed alarms are ineffective in actively reducing falls.

Using machine learning (ML) to monitor bed bound patients can mitigate the current issues associated with monitoring bed-bound patients. Current literature on ML for fall detection and prevention indicates that many studies use wearable inertial measurement units (IMU), which generate data that will be analyzed by an ML algorithm. Some of the ML algorithms include, Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF), k-Nearest Neighbors (kNN), k-means, Linear Discriminant Analysis (LDA) [10]. Each algorithm has its own benefits and shortcomings depending on the specific data that is fed. Overall, the accuracy, sensitivity and specificity of fall detection and prevention were comparable among the different ML algorithms, with SVM having an accuracy of 98%, kNN having an accuracy of 99%, ANN having an accuracy of 95.25% when using an IMU sensor at the waist [10]. Sensor location is another important aspect as different studies place the sensor in different locations in the body to maximize the data that can be captured. Most studies seem to place the sensors at the waist.

In the field of ML for patient care [17], a significant focus has been on fall detection and prevention, as evidenced by a systematic review by Usmani et al. [16]. This review underscores the increasing adoption of wearable sensors, such as IMUs, to monitor patient movements. It also highlights the effectiveness of ML models like SVM, RF, and ANN, which have achieved high accuracy rates (up to 99%) in fall detection. However, our approach extends beyond fall detection. We aim to detect six different movements typically performed by bed-based patients, thereby providing a more comprehensive understanding of patient mobility. By integrating Electronic Medical Record (EMR) data, including mobility restrictions, history of falls, and cognitive impairments, with sensor data, we aim to further optimize the predictive accuracy of our regression models [18]. This approach allows for the generation of personalized alerts tailored to the patient's specific condition and treatment regimen, thereby creating a more dynamic and proactive system for patient care [19].

2. Materials and Methods

This section begins with the 'Data Collection' subsection, which explains the use of a mannequin model and sensors to capture and transmit data related to patient movements. This is followed by the 'Machine Learning' subsection, where the preparation of the dataset for ML is discussed, including the normalization of data and the handling of class imbalance. The 'Models' subsection introduces the three ML models used in the study, while the 'Evaluation' subsection describes the evaluation of these models using the R-squared score and mean squared error. The 'Cross-Validation' subsection further elaborates on the use of KFold Cross-Validation in the ML pipeline. Finally, the 'Calculating Confusion Matrix' subsection details the use of a confusion matrix to evaluate the performance of the models, including a strategy to categorize the continuous output of regression models into discrete classes.

2.1. Data Collection

In this study, we utilized a high-fidelity mannequin, a tool commonly used in medical education and simulation centers, to simulate the various movements typically experienced by inpatient

individuals in a bed setting. This mannequin, often employed by medical students for practicing diverse patient scenarios, was adapted to replicate the common movements and positions of bedridden patients.

In the data collection process, a single high-fidelity mannequin was utilized. This mannequin was equipped with Movella DOT sensors to simulate and capture the various movements typically experienced by inpatient individuals in a bed setting. The use of a single mannequin ensured consistency in the data collected and controlled for any potential variability that might arise from using multiple mannequins. This approach allowed the focus to be on the effectiveness of the ML algorithms in detecting these movements. The Movella DOT sensors are wearable sensor development platforms known for their signal processing and sensor fusion framework, making them particularly optimized for applications involving human movement. The sensor was positioned on the mannequin's torso, specifically at the midpoint of the sternum, i.e., a critical anatomical landmark. This placement was done to ensure optimal data capture and reproducibility of the measurements. To secure the sensor in place and prevent any displacement during the experiments, a cross-shaped fixation method was employed using a high-strength, adhesive-backed material. The Movella DOT sensors, which include an accelerometer, a gyroscope, and a magnetometer, were strategically placed on the mannequin, as shown in Figure 1. These sensors recorded velocities, accelerations, and Euler angles at specific time intervals, providing a comprehensive dataset of the mannequin's movements [20]. The data was transmitted wirelessly via Bluetooth to eliminate any potential interference from cables.

The accelerometer measured the acceleration of the mannequin's movements, the gyroscope captured the rotational speed, and the magnetometer recorded the magnetic field. The data from these individual components were then combined using sensor fusion algorithms to accurately calculate the mannequin's orientation.

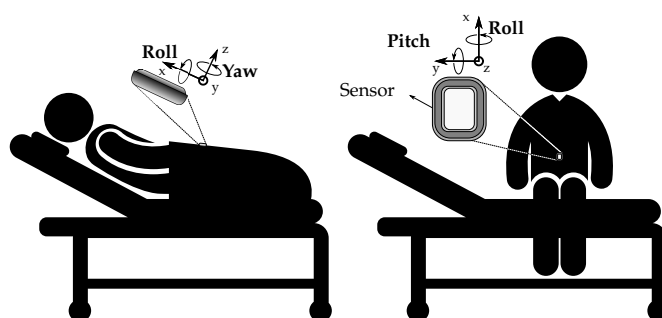


Figure 1. Illustration of pitch, roll, and yaw relative to a patient in both supine and sitting positions.

The sensor was placed on both adult-sized and infant-sized mannequins to simulate a range of patient demographics. Each movement was repeated approximately 100 times to gather a substantial amount of data for each movement type. For breathing, the mannequin performed one full cycle of tidal volume inhalations and exhalations for 3 minutes straight. For seizures, the mannequin underwent a seizure for 10 minutes straight. For the rolling and falling off the bed movements, the infant mannequin was used. The mannequin was started in the supine position and rolled approximately 90 degrees to its left or right side and back to the original supine position. This was repeated approximately 100 times for each side. The data for dropping off the bed from the left was collected by having the infant mannequin start in the supine position and rolling it beyond its left side to the point where it falls off the bed. This was also repeated approximately 100 times.

This approach of using a high-fidelity mannequin and sensors like Movella DOT has enabled us to gather precise and reliable data on the movements typically experienced by inpatients. The movements were then categorized into six distinct labels: "Roll right" (0), "Roll left" (1), "Drop right" (2), "Drop left" (3), "Breathing" (4), and "Seizure" (5). These labels represent the most common movements and positions experienced by inpatients, providing a comprehensive understanding of patient behavior in a bed setting.

2.2. Machine Learning

The methodology involves the application of ensemble ML models on a given dataset [21]. The dataset is first imported into the Python environment using the pandas library. The dataset is divided into two parts: the input data (X), which consists of all columns except the first one, and the target variable (y), which is the first column. The input data is then normalized to be between 0 and 1 using the MinMaxScaler from the sklearn.preprocessing module. This is done to ensure that all input data is on the same scale, preventing the model from being biased towards data with larger scales.

In our dataset, we observe an imbalance in the distribution of classes within the target variable. This imbalance can potentially lead to a model bias, where the model might overfit to the majority class and underperform when predicting the minority class. To mitigate this issue, we employ the Synthetic Minority Over-sampling Technique (SMOT), which is a technique that generates synthetic data for the minority class, thereby balancing the class distribution. The process of synthetic data generation involves calculating the k nearest neighbors for each instance in the minority class, randomly choosing one of these neighbors, and creating a synthetic instance at a random point on the line segment connecting the two instances. This can be represented by the following formula:

$$x_{\text{synthetic}} = x_{\text{minority}} + \lambda \cdot (x_{\text{neighbor}} - x_{\text{minority}}), \quad (1)$$

where $x_{\text{synthetic}}$ is the synthetic instance, x_{minority} is the instance from the minority class, x_{neighbor} is the randomly chosen neighbor, and λ is a random number between 0 and 1.

The sampling strategy, to mitigate the imbalance in the dataset, is set such that each class in the target variable has 200 instances. The random state is set to 42 for reproducibility. This is a common practice in ML where a specific seed is set for the random number generator. This allows the results to be reproduced exactly, as the same “random” numbers will be generated each time the code is run.

The data is then split into training and test sets, with 80% of the data used for training and 20% used for testing. The random state is again set to 42 for reproducibility. Several models are then fitted to the training data, including the Decision Tree Regressor, Gradient Boosting Regressor, and Bagging Regressor.

The methodology diverges from traditional feature selection processes due to the regression-based nature of the model. In contrast to the conventional feature selection methods often employed in classification problems, this study assesses the model's effectiveness based on its precise prediction of movements, using the entire dataset. This dataset is compiled from measurements captured by sensors as a mannequin executed six unique movements along three specific axes: east-west, north-south, and up-down. Consequently, the typical notion of feature selection, frequently observed in classification problems, is not directly applicable in this regression-based scenario. The model's performance is instead evaluated on its capacity to accurately forecast the movements, utilizing the entire dataset.

2.3. Models

Following a review of various models, taking into account their performance, complexity, and susceptibility to overfitting, the decision was made to employ the Decision Tree Regressor, the Gradient Boosting Regressor, and the Bagging Regressor. The Bagging Regressor is an ensemble model that primarily uses the Decision Tree Regressor as its base regressor. Detailed descriptions of all three models are provided below.

The Decision Tree Regressor (DTR) is a simple model that splits the data at certain thresholds to make predictions. The formula for a decision tree can be represented as:

$$Y = f(X) = \sum_{m=1}^M c_m I(x \in R_m), \quad (2)$$

where Y is the output, X is the input, R_m are the regions of the data space that result from the splits, c_m are the constants (mean of the target variable in the region R_m), and I is the indicator function.

The Gradient Boosting Regressor (GBR) is a more complex model that fits new predictors to the residual errors of the previous predictor. The formula for gradient boosting can be represented as:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^J \gamma_{jm} I(x \in R_{jm}), \quad (3)$$

where $F_m(x)$ is the boosted model at step m , $F_{m-1}(x)$ is the boosted model at the previous step, R_{jm} are the regions of the data space that result from the splits, γ_{jm} are the optimal coefficients, and I is the indicator function.

The Bagging Regressor (BR) is an ensemble model that fits base regressors each on random subsets of the original dataset and then aggregates their individual predictions to form a final prediction. The formula for bagging can be represented as:

$$F(x) = \frac{1}{B} \sum_{b=1}^B f_b(x), \quad (4)$$

where $F(x)$ is the final prediction, B is the number of base regressors, and $f_b(x)$ is the base regressor. In the context of a BR, base regressors are the individual models that are trained on different subsets of the original dataset. The final prediction of the BR is typically an average of the predictions made by each of these base regressors. In this study, the number of base regressors (B) is 50. It means that the BR is training 50 separate instances of its base regressor on different subsets of the data. These 50 models collectively make up the BR. In Python, BR is implemented using the DTR as the base regressor. BR enhances the performance of DTR by reducing overfitting and improving prediction accuracy.

The default hyperparameters, as provided by the Python libraries for the DTR, GBR, and BR methods, were employed in this study. This selection was based on a series of experiments involving various parameters, none of which demonstrated a significant improvement over the default settings. These default hyperparameters, having been well-optimized and extensively tested by the ML community, were deemed appropriate. By adhering to these default settings, a simpler methodology was maintained without sacrificing the results, which is considered beneficial at this stage for facilitating easier reproducibility and replicability of the study.

2.4. Evaluation

The models are evaluated using the R-squared (R^2) score, and mean squared error (MSE) on both the training and test sets. In a regression problem, the goal is to predict a continuous outcome variable from one or more predictor variables. The performance of regression models is typically assessed using error metrics that quantify the difference between the predicted and actual values. Two of the most common metrics are MSE and R^2 . MSE is the average of the squared differences between the predicted and actual values. It is a measure of the model's prediction error. A lower MSE indicates a better fit of the model to the data, as it means the model's predictions are closer to the actual values. In contrast, accuracy and Area Under the Curve (AUC) are metrics typically used for classification problems, where the goal is to predict a categorical outcome. Accuracy measures the proportion of correct predictions, while AUC measures the ability of the model to distinguish between classes. These metrics are not suitable for regression problems because regression predictions are not categorical, but continuous. Hence, MSE and R^2 are used in regression problems because they provide meaningful measures of the model's ability to predict continuous outcomes.

MSE is the average of the squared differences between the predicted and actual values. It is a measure of the model's prediction error. A lower MSE indicates a better fit of the model to the data, as it means the model's predictions are closer to the actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (5)$$

where Y_i are the actual values, \hat{Y}_i are the predicted values, and n is the number of observations.

R^2 , also known as the coefficient of determination, measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with 1 indicating that the model perfectly predicts the actual values and 0 indicating that the model does not explain any of the variability of the response data around its mean. A higher R^2 indicates a better fit of the model to the data. The formula for R^2 is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}, \quad (6)$$

where SS_{res} is the sum of squares of residuals and SS_{tot} is the total sum of squares.

In the context of this research, it remains beneficial to compute accuracy, particularly because the continuous predictions are classified into bins corresponding to a target categorical variable expressed numerically. In this case, a continuous variable representing different movements of bedridden patients is binned, and the predictions are subsequently categorized into labels such as “Roll right” (0), “Roll left” (1), “Drop right” (2), “Drop left” (3), “Breathing” (4), and “Seizure” (5).

2.5. Cross-Validation

The ML pipeline utilized uses cross-validation to evaluate the performance of the resulting model. A technique called KFold Cross-Validation is utilized. This method involves breaking down the training data into several sections, specifically 10 in this case. These sections are often referred to as “folds”. To make sure the results can be replicated, a specific “random state” is set. Think of it as setting a seed for a random number generator. In this context, the seed is 42. Moreover, the data is “shuffled” before it’s divided into folds. This is to ensure that the sequence of the data doesn’t influence the outcomes of the model training.

Once the data is divided into folds, the ensemble model undergoes training and evaluation several times. In each cycle, a different fold of the data is set aside for evaluation while the model is trained on the remaining data. The model’s performance in each cycle is assessed and a score is given. This process is repeated for each fold of the data, resulting in a collection of scores. In mathematical terms, if we consider n as the total number of data points and k as the number of folds, the model will be trained k times. Each time, it will be trained on $n - (n/k)$ data points and evaluated on (n/k) data points. The final outcome is an array of k scores, one for each cycle.

2.6. Calculating Confusion Matrix

The confusion matrix is a table that is used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows visualization of the performance of an algorithm. The model makes a prediction for each sample in the test set. A confusion matrix is traditionally used for evaluating classification models; however, by employing a binning strategy to categorize the continuous output of regression models into discrete classes, it becomes possible to calculate a confusion matrix for regression models as well.

Since the BR’s output is continuous, to make these predictions meaningful in the context of categorical labels, they need to be categorized into bins. Binning is used as a method to convert continuous predictions from a BR into categorical labels. It is used to segment and sort data values into bins. Bin edges [-0.5, 0.5, 1.5, 2.5, 3.5, 4.5, 5.5] are used to categorize the continuous predictions from the BR into one of the six categories represented by the labels “Roll right” (0), “Roll left” (1), “Drop right” (2), “Drop left” (3), “Breathing” (4), and “Seizure” (5). Each bin edge represents the boundary between two categories. For example, a prediction value of 0.3 would fall into the bin between -0.5 and 0.5, and would be labeled as ‘Roll right’. A prediction value of 1.7 would fall into the bin between 1.5 and 2.5, and would be labeled as ‘Roll left’, and so on.

Finally, the confusion matrix is calculated using the true test labels and the predicted test labels, which have been transformed into numerical form. The resulting confusion matrix provides an

additional evaluation of the BR model’s performance. To reiterate, this approach allows us to effectively evaluate a model that produces continuous output by using a binning strategy.

In ML studies, the dataset is divided into training and testing sets to ensure a robust evaluation of the model’s performance. Furthermore, in this study, the SMOT was employed to generate synthetic data for the minority class, thereby balancing the class distribution in the training set. Note that this explains the discrepancy between the number of instances mentioned above and the number of instances shown in the confusion matrix below. Similarly, the learning curves were generated by incrementally increasing the size of the training set, which is why the horizontal axis shows different training sizes. Hence, the size of the training set used to train the model does not correlate with the number of instances in the raw dataset.

3. Results

In the pursuit of identifying the most appropriate ML regressors for the given dataset, an initial analysis was conducted on a selection of commonly used regressors. The findings of this analysis are presented in the Table 1, with the algorithms arranged in descending order of performance.

Table 1. Performance Comparison of Different Regressors.

Regressor:	MSE:	R2:
Gradient Boosting	0.093	0.970
Random Forest	0.120	0.962
Decision Tree	0.138	0.956
Support Vector	0.220	0.930
K-Nearest Neighbors	0.322	0.897
Linear	19.419	-5.209

The GBR emerged as the superior performer. This ML model is adept at detecting intricate patterns within the data. However, it is susceptible to overfitting, as it constructs trees in a sequential manner, with each new tree designed to rectify errors made by its predecessors. As the model incorporates more trees, it becomes increasingly expressive, which, if not properly managed, can lead to overfitting.

The Random Forest, the second-best performer, shares GBR’s ability to discern complex patterns in the data. However, it too is prone to overfitting, particularly when dealing with noisy data. This is attributed to the Random Forest’s methodology of constructing numerous deep trees, each trained on a different data subset. While this can result in a model that fits the training data exceptionally well, it may perform poorly on unseen data.

In contrast, the DTR is less likely to overfit due to its simplicity. Decision trees are generally more interpretable and can handle both numerical and categorical data. Despite this, they can still overfit if allowed to grow excessively deep, although this is typically easier to control than in more complex models.

Given the circumstances, we opted for the BR, an ensemble model that incorporates the DTR as its base regressor. This method takes advantage of DTR’s inherent ability to counteract overfitting. While the GBR or Random Forest may offer marginally better MSE and R2 values, their use heightens the overfitting problem with our dataset. The slight improvement in these metrics does not justify their use if they contribute to overfitting issues.

Table 2. Performance Comparison of Decision Tree Regressor, Gradient Boosting Regressor, and their Ensemble, without SMOT.

Model	Accuracy	R2 (Train)	R2 (Test)	MSE (Train)	MSE (Test)
DTR	0.8362	1.000	0.9035	0.000	0.3017
GBR	0.8621	0.9925	0.9702	0.0219	0.0932
BR	0.8621	0.9915	0.9592	0.0247	0.1276

The Table 3 presents a comparative evaluation of three distinct models: DTR, GBR, and BR. The performance of these models is assessed using metrics such as Accuracy, R2 (for both training and testing datasets), and MSE (for both training and testing datasets).

Table 3. Performance Comparison of Decision Tree Regressor, Gradient Boosting Regressor, and their Ensemble, with SMOT.

Model	Accuracy	R2 (Train)	R2 (Test)	MSE (Train)	MSE (Test)
DTR	0.892	1.000	0.939	0.000	0.167
GBR	0.908	0.990	0.943	0.031	0.156
BR	0.950	0.996	0.959	0.012	0.112

The DTR model demonstrates an accuracy of 0.892, an R2 score of 1.000 for the training data, and an R2 score of 0.939 for the test data. The MSE for the training data is 0.000, suggesting no error, while the MSE for the test data is 0.167. The GBR model exhibits a slightly higher accuracy of 0.908, an R2 score of 0.990 for the training data, and an R2 score of 0.943 for the test data. The MSE for the training data is 0.031, and for the test data, it's 0.156.

The ensemble model, i.e., BR, outperforms both individual models with an accuracy of 0.950, an R2 score of 0.996 for the training data, and an R2 score of 0.959 for the test data. The MSE for the training data is 0.012, and for the test data, it is 0.112. Additionally, BR models are known for their robustness to overfitting, which is a significant advantage in predictive modeling. While the BR may present a higher computational cost compared to simpler models, its superior performance justifies this trade-off. Although the interpretability of the BR may not be as straightforward as that of simpler models, the primary focus of this study was on achieving high predictive accuracy. Furthermore, the BR is highly scalable, making it suitable for future expansions of the study.

3.1. Cross-Validation

The BR, our chosen model, was evaluated using a KFold Cross-Validation technique. This involved dividing the dataset into 'k' subsets. Each unique subset served as a test dataset, with the remaining subsets forming the training dataset. The model was fitted on the training set and evaluated on the test set, a process repeated 'k' times. Therefore, the number of runs for the BR method equates to the number of folds in the cross-validation. The resulting mean and standard deviations are detailed in Table 4. The DTR and GBR were not subjected to these calculations before the decision to proceed with BR was made. The choice of BR over DTR and GBR was based on specific considerations, which are outlined and defended above. The mean and standard deviations of DTR and GBR, which were not used further, did not influence this decision.

The Table 4 further compares the performance of a model with and without the SMOT. The metrics used for comparison are the mean and standard deviation. The mean represents the average performance of the model, while the standard deviation measures the variability of the model's performance.

Table 4. Performance Comparison of Ensemble Model with and without SMOT

Technique:	Mean:	Standard Deviation:
Without SMOT	0.9374	0.0258
With SMOT	0.9705	0.0146

It can be observed that the model performs better when SMOT is used, as indicated by the higher mean score. The standard deviation is also lower with SMOT, suggesting that the model's performance is more consistent when this technique is used.

3.2. Learning Curves

Learning curves are a diagnostic tool in ML that provides insights into how well a model is learning from the training data and generalizing to unseen data. They plot the model's performance on both the training and cross-validation datasets over a series of training iterations. The training score reflects how well the model fits the training data. The cross-validation score, on the other hand, indicates how well the model generalizes to unseen data. If the training score is significantly higher than the validation score, it suggests that the model might be overfitting, meaning it is too complex and is fitting the noise in the training data rather than the underlying pattern (as seen in Figure 2a). The point where the training score and cross-validation score converge is considered a good indication of the optimal model complexity (as seen in Figure 2b).

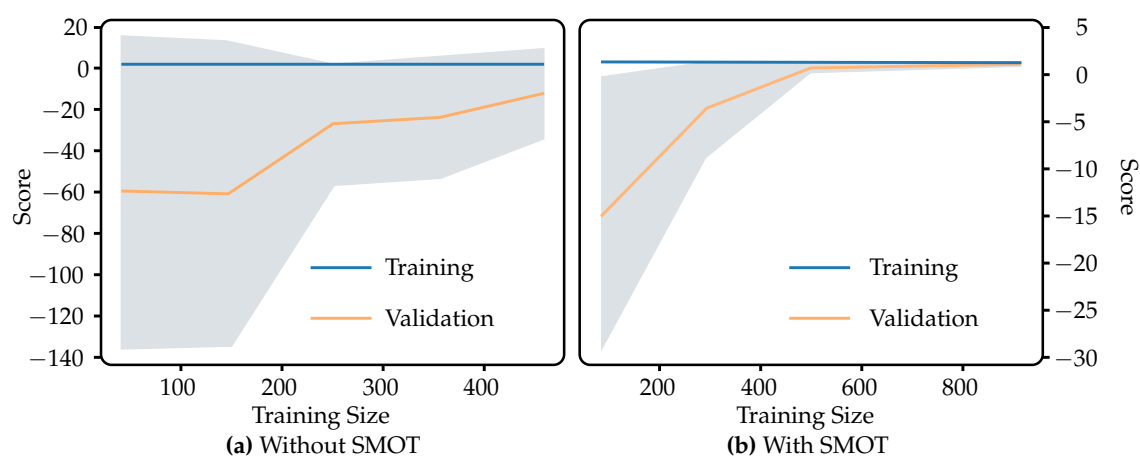


Figure 2. The performance of the ensemble model on the training and validation sets over a series of training iterations. Part (a) shows the learning curve without the application of SMOT, while part (b) demonstrates the learning curve with SMOT. The use of SMOT appears to enhance the model's ability to handle imbalanced datasets, as evidenced by the improved performance in part (b).

Note, that the maximum training sizes (in Figure 2) are different with and without SMOT because SMOT creates synthetic examples of the minority class, effectively increasing the size of the training data. This allows the model to be trained with more data when SMOT is used.

The learning curves in Figure 2a start with a significant gap of 60 score points between the training and cross-validation scores. This large gap indicates that the model is initially overfitting the training data, meaning it performs well on the training data but poorly on the unseen cross-validation data. As the model is trained with more data, the cross-validation score rises, suggesting that the model is learning and generalizing better. However, by the maximum training size, the curves are still far apart, indicating that the model is still overfitting. This suggests that the model has learned the training data too well and is not generalizing well to unseen data.

The learning curves in Figure 2b start with a smaller gap of 15 score points between the training and cross-validation scores. This smaller gap suggests that the model is not overfitting as much as in the case without SMOT. As the model is trained with more data, the cross-validation score rises and converges with the training score by approximately halfway between the smallest and largest training size. Once converged, the two scores remain close for the rest of the training size, from the middle to the maximum. This sustained convergence indicates that the model is generalizing well to unseen data and is not overfitting or underfitting. The use of SMOT, which creates synthetic examples of the minority class, seems to help the model learn better and generalize better to unseen data.

The SMOT was employed to mitigate the issue of overfitting, as evidenced by the non-convergence of the training and validation curves in the initial model (Figure 2a). The application of SMOT led to the convergence of these curves (Figure 2b), indicating an improvement in the model's performance. Although the specific changes in the distribution of movement classes after applying SMOT were

not directly evaluated, the overall enhancement in the model’s performance suggests that SMOT effectively bolstered the predictive power for minority classes.

3.3. Confusion Matrix

The confusion matrix, shown in Figure 3, represents a 6-class classification problem. The classes are: Roll right (0), Roll left (1), Drop right (2), Drop left (3), Breathing (4), and Seizure (5). The rows of the matrix represent the actual classes, while the columns represent the predicted classes by the ML model. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions.

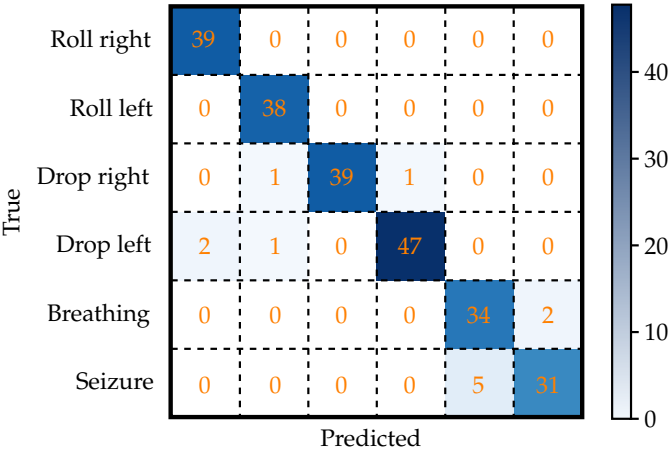


Figure 3. Confusion matrix heatmap for the model. The x-axis represents the predicted labels and the y-axis represents the actual labels. The color intensity in each cell corresponds to the number of instances, with darker colors indicating higher numbers. The actual numbers are also included in each cell of the heatmap.

In this study, the SMOT was employed to balance the class distribution in the dataset. Note that SMOT was applied before the data splitting process, and therefore, the synthetic instances generated by SMOT are included in both the training and testing sets. The data was then split into training and testing sets. However, the distribution of classes in the testing set is subject to the randomness of the data splitting process, and therefore, it does not necessarily reflect the exact ratio of the original class distribution. The confusion matrix in Figure 3 represents the performance of the model on the testing set, which includes both real and synthetic instances.

The model correctly predicted all 39 instances of Roll right (0), as there are no off-diagonal elements in the first row. Similarly, the model correctly predicted all 38 instances of Roll left (1), as there are no off-diagonal elements in the second row. For Drop right (2), the model correctly predicted 39 instances, but misclassified 1 instance as Roll left (1) and 1 instance as Drop left (3). For Drop left (3), the model correctly predicted 47 instances, but misclassified 2 instances as Roll right (0) and 1 instance as Roll left (1). For Breathing (4), the model correctly predicted 34 instances, but misclassified 2 instances as Seizure (5). For Seizure (5), the model correctly predicted 31 instances, but misclassified 5 instances as Breathing (4).

Overall, the model seems to be doing well in predicting the classes, as the majority of the predictions fall on the diagonal (correct predictions). However, there are some misclassifications, particularly between Breathing (4) and Seizure (5), and to a lesser extent between Drop right (2) and Drop left (3), and Drop left (3) and Roll right (0) and Roll left (1).

4. Discussion

The main utility of utilizing sensor-based surveillance is the effectiveness in continuously monitoring patients while avoiding privacy concerns of other technologies, such as AI-based video surveillance.

In one study, the authors utilized a security framework that had successfully maintained both privacy and security in the healthcare setting [13]. The recording system was activated only as a response to health or security concerns. Additionally, only medical professionals (nurses, doctors) may access the encrypted video data. The ability to balance privacy and security in video-based patient monitoring systems. The regression model in the current study did not rely on video surveillance which ensures absolute privacy by utilizing the Movella DOT sensors [22]. In another study which used Internet-of-Things Sensors and AI to monitor patient activities and alert healthcare professionals to recent changes [23]. It was found that with the use of the CNN-UUGRU deep learning model, which outperforms existing models in terms of accuracy (97.7%) and precision (96.8%) in identifying human activities based on sensor data [23]. Additionally, this sensor-based system can predict possible falls based on behavioral and physiological data, as discussed in fall prevention research by El-Bendary et al. [24]. These systems not only detect immediate threats but also help predict falls before they occur, utilizing patient history and sensor fusion technology.

Incorporating EHR data not only enhances the predictive power of these models but also allows for the creation of personalized care plans based on a combination of sensor-based movement data and detailed patient histories. For example, patients with a history of mobility issues or chronic diseases that impact their movement (such as arthritis or Parkinson's disease) could have higher-risk scores calculated by the model, which can trigger early intervention measures. In line with this, Lee et al.'s systematic review (2020) emphasizes that integrating predictive models into EHR systems significantly improves clinical outcomes, particularly in critical care settings like sepsis management and thrombotic disorder prevention [25].

Usmani et al. conducted a systematic review of fall detection and prevention using ML, emphasizing the growing use of wearable sensors, such as IMUs, to capture patient movements. Their review highlights the effectiveness of ML models like SVM, RF, and ANN, all achieving high accuracy rates (up to 99%) in fall detection. By including EMR data, such as mobility restrictions, history of falls, and cognitive impairments, alongside sensor data, the predictive accuracy of the regression models could be optimized. This integration would allow the system to generate alerts that are personalized to the patient's specific condition and treatment regimen, creating a more dynamic and proactive fall prevention system [16].

The selection of the six specific movement categories in this study was based on their relevance to the safety and well-being of bed-ridden patients. These movements include breathing, seizures, rolling to the right side, rolling to the left side, rolling off the bed from the left, and rolling off the bed from the right. These categories were chosen to provide a comprehensive overview of potential scenarios that these patients might encounter. While there are other movements that patients might perform in bed, such as shifting positions or reaching for objects, these were not included in the current study. Future research could consider these additional movements to further enhance the model's practical applicability.

The data collection process was designed to minimize the potential for noise and outliers. A single mannequin was employed to perform six distinct types of movements, and the Euler angles were measured and collected during these movements. This controlled setup significantly reduced the likelihood of extraneous variables introducing noise or causing outliers. Therefore, additional preprocessing steps such as data augmentation or noise filtering were not deemed necessary. However, it is acknowledged that in real-life scenarios, additional preprocessing methods may be required to account for potential artifacts or outliers. This is an area of focus for future research.

Sensor placement, noise levels, and sampling rates are pivotal factors that can influence the accuracy and precision of ML models. The positioning of the sensor can affect the quality of the data collected, as different placements may capture varying aspects of the movements. Noise levels can introduce an element of variability in the data, potentially leading to misclassification of movements. The sampling rates can impact the granularity of the data, with higher rates providing more detailed data that could enhance the model's performance. However, in the context of this study, the movements

were simulated by a mannequin in a controlled environment, which mitigated the influence of these factors on the model's performance.

The constraints imposed by the limited size of the dataset are recognized in this study. Data collection in this domain can be challenging due to privacy considerations and the sensitive nature of the information. Although the dataset is of limited size, it is regarded as representative and provides useful insights. It has been utilized to train an ML model capable of detecting and classifying different types of movements in bed-based patients. To address the issue of imbalanced data, the SMOT was utilized. While the quality of synthetic data may raise concerns, it is important to highlight that SMOT is a widely accepted approach for handling imbalanced datasets in ML. Its effectiveness in enhancing the performance of ML models has been demonstrated in numerous studies [26,27]. Furthermore, patient variability was not specifically addressed in this study, indicating an area for future research.

This study primarily relies on internal validation techniques, such as K-fold cross-validation and learning curves, to assess the model's performance. The learning curves of the BR model illustrate the convergence of the training and validation curves, with the validation curve consistently staying slightly below the training curve. This pattern indicates that the model, in spite of its complexity, does not appear to be overfitting. Nonetheless, it is acknowledged that the generalizability of the model is a subject that requires further exploration. Future research directions are set to include testing the model on human subjects and in real-life environments to verify its applicability in a variety of healthcare settings.

The computational efficiency of the proposed model is characterized by its ability to be trained, validated, and tested within a short time frame on a single PC unit. This suggests that the model is computationally efficient and could potentially be implemented in a real-time hospital setting. However, the specific hardware and software requirements for such an application have not been evaluated in this study. Future research could focus on determining these requirements to facilitate the implementation of the model in a real-time hospital setting. It is important to note that the primary focus of this study was the development and testing of the model, and as such, the specific computational requirements for real-time application were not evaluated. Regarding cost-effectiveness, while it was not directly evaluated in this study, the sensors used are inexpensive and reusable, suggesting that the model could potentially be more cost-effective than traditional video monitoring systems. Moreover, while the sensors utilized are effective in capturing patient movements, they may not be able to capture all types of patient behavior or conditions. For instance, they might not be able to detect subtle changes in a patient's condition that could be picked up by a human observer. Additionally, the sensors' effectiveness could potentially be influenced by factors such as their placement on the patient or the patient's position in bed. These limitations could affect the reliability and accuracy of movement detection, potentially leading to misclassifications or missed detections. However, it is important to note that these limitations are inherent in the use of sensor-based systems and do not necessarily undermine the overall utility of the model.

The sensor-based monitoring approach was selected primarily for its potential to offer enhanced patient privacy, as many individuals may feel uncomfortable being under video surveillance while sleeping in a hospital room. This study, however, did not explicitly compare sensor-based monitoring to video monitoring in terms of detection latency, false positives, and real-time responsiveness. This is primarily because our study was designed as a proof of concept rather than a comprehensive assessment with a fully operational clinical application in mind. While we can hypothesize about the potential advantages and limitations of sensor-based monitoring compared to video monitoring, these are not grounded in the empirical data collected in this study.

5. Conclusions

The primary objective of this study was to address the challenge of continuous patient monitoring while ensuring privacy. The proposed methodology involved the use of Movella DOT sensors, which provided a privacy-preserving alternative to AI-based video surveillance. This sensor-based

surveillance system proved to be relevatively effective in monitoring patients' movements in bed. The experiments conducted in this study involved the evaluation of three distinct models: DTR, GBR, and BR. The performance of these models was assessed using metrics such as Accuracy, R2 score, and MSE. Among these, the ensemble model, BR, outperformed both DTR and GBR in terms of accuracy, R2 score, and MSE. The confusion matrix heatmap reveals some misclassifications, particularly between the labels "Breathing" and "Seizure". These misclassifications can be attributed to the inherent similarities between these two movements, which involve rhythmic patterns that differ primarily in frequency rather than magnitude. This is a known limitation of machine learning models, which can struggle to differentiate between classes with similar features. Despite these misclassifications, the overall performance of the model is reasonable, as evidenced by the majority of predictions falling on the diagonal, indicating correct predictions.

The results demonstrated the effectiveness of the proposed methodology. The BR model achieved an accuracy of 0.950, an R2 score of 0.996 for the training data, and an R2 score of 0.959 for the test data. The MSE for the training data was 0.012, and for the test data, it was 0.112. The discussion highlighted the utility of sensor-based surveillance and the potential of integrating AI systems with EHR systems. The integration of EHR data can enhance the predictive power of the models and allow for the creation of personalized care plans.

However, the study recognized the limitations imposed by the limited size of the dataset. Data collection in this domain can be challenging due to privacy considerations and the sensitive nature of the information. To address the issue of imbalanced data, the SMOT was utilized. Future work will focus on expanding the dataset and exploring other ML models. The quality of synthetic data generated by SMOT could also be further investigated. Despite these limitations, the study provides useful insights into the potential of sensor-based surveillance in healthcare settings.

As with any ML model, a certain degree of misclassification is anticipated, particularly in the healthcare domain where the complexity and variability of human physiological patterns can lead to overlaps between different conditions. However, these misclassifications do not detract from the clinical applicability of ML models. Instead, they underscore the necessity of using ML as a supportive tool, not a replacement, for human judgement. In real-time monitoring environments, the model's results should not be relied upon absolutely, but rather used to assist healthcare personnel in making more informed decisions. Strategies such as setting thresholds for alarm activation, incorporating feedback mechanisms, and providing training to healthcare personnel on the interpretation of the model's outputs could help mitigate such errors.

Author Contributions: All authors (P.J., H.A., D.M., A.T., R.J., J.M., M.B., T.D., and M.T.) contributed to all aspects of this work. This includes but is not limited to conceptualization, methodology, software management, validation, formal analysis, investigation, resources management, data curation, writing—original draft preparation, writing—review and editing, and visualization. The supervision, and project administration were handled by M.T.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

ML	Machine Learning
IMU	Inertial Measurement Units
SVM	Support Vector Machine
ANN	Artificial Neural Network
RF	Random Forest

kNN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
EMR	Electronic Medical Record
SMOT	Synthetic Minority Over-sampling Technique
DTR	Decision Tree Regressor
GBR	Gradient Boosting Regressor
BR	Bagging Regressor
MSE	Mean Squared Error
R2	R-Squared
AUC	Area Under the Curve

References

1. Linnerud, S.; Kvael, L.A.H.; Graverholt, B.; Idland, G.; Taraldsen, K.; Brovold, T. Stakeholder development of an implementation strategy for fall prevention in Norwegian home care – a qualitative co-creation approach. *BMC Health Services Research* **2023**, *23*. doi:10.1186/s12913-023-10394-x.
2. Karen B. Pearson, MLIS, M.; Andrew F. Coburn, P. Evidence-based Falls Prevention in Critical Access Hospitals, 2011.
3. Jennifer Van Pelt, M. Fall Prevention: Fall Prevention in Hospitals. *Today's Geriatric Medicine* **2023**, *16*, 28.
4. Morello, R.T.; Soh, S.E.; Behm, K.; Egan, A.; Ayton, D.; Hill, K.; Flicker, L.; Etherton-Beer, C.D.; Arendts, G.; Waldron, N.; Redfern, J.; Haines, T.; Lowthian, J.; Nyman, S.R.; Cameron, P.; Fairhall, N.; Barker, A.L. Multifactorial falls prevention programmes for older adults presenting to the emergency department with a fall: systematic review and meta-analysis. *Injury Prevention* **2019**, *25*, 557–564. doi:10.1136/injuryprev-2019-043214.
5. Ang, G.; Low, S.; How, C. Approach to falls among the elderly in the community. *Singapore Medical Journal* **2020**, *61*, 116–121. doi:10.11622/smedj.2020029.
6. Cuevas-Trisan, R. Balance Problems and Fall Risks in the Elderly. *Physical Medicine and Rehabilitation Clinics of North America* **2017**, *28*, 727–737. doi:10.1016/j.pmr.2017.06.006.
7. Whitney, J.; Close, J.C.; Jackson, S.H.; Lord, S.R. Understanding Risk of Falls in People With Cognitive Impairment Living in Residential Care. *Journal of the American Medical Directors Association* **2012**, *13*, 535–540. doi:10.1016/j.jamda.2012.03.009.
8. Colón-Emeric, C.S.; McDermott, C.L.; Lee, D.S.; Berry, S.D. Risk Assessment and Prevention of Falls in Older Community-Dwelling Adults: A Review. *JAMA* **2024**, *331*, 1397. doi:10.1001/jama.2024.1416.
9. NICE. Clinical Guideline 161. Falls: assessment and prevention of falls in older people. <https://www.nice.org.uk/guidance/cg161>, 2014. Accessed December 19, 2014.
10. LeLaurin, J.H.; Shorr, R.I. Preventing Falls in Hospitalized Patients. *Clinics in Geriatric Medicine* **2019**, *35*, 273–283. doi:10.1016/j.cger.2019.01.007.
11. Rausch, D.L.; Bjorklund, P. Decreasing the costs of constant observation. *The Journal of nursing administration* **2010**, *40*, 75–81. doi:NNA.0b013e3181cb9f56.
12. Cournan, M.; Fusco-Gessick, B.; Wright, L. Improving Patient Safety Through Video Monitoring. *Rehabilitation nursing : the official journal of the Association of Rehabilitation Nurses* **2018**, *43*, 111–115. doi:10.1097/RNJ.000000000000089.
13. Woltsche, R.; Mullan, L.; Wynter, K.; Rasmussen, B. Preventing Patient Falls Overnight Using Video Monitoring: A Clinical Evaluation. *International Journal of Environmental Research and Public Health* **2022**, *19*, 13735. doi:10.3390/ijerph192113735.
14. Seow, J.P.; Chua, T.L.; Aloweni, F.; Lim, S.H.; Ang, S.Y. Effectiveness of an integrated three-mode bed exit alarm system in reducing inpatient falls within an acute care setting. *Japan Journal of Nursing Science* **2021**, *19*. doi:10.1111/jjns.12446.
15. Mileski, M.; Brooks, M.; Topinka, J.B.; Hamilton, G.; Land, C.; Mitchell, T.; Mosley, B.; McClay, R. Alarming and/or Alerting Device Effectiveness in Reducing Falls in Long-Term Care (LTC) Facilities? A Systematic Review. *Healthcare* **2019**, *7*, 51. doi:10.3390/healthcare7010051.
16. Usmani, S.; Saboor, A.; Haris, M.; Khan, M.A.; Park, H. Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors* **2021**, *21*, 5134. doi:10.3390/s21155134.

17. Bekbolatova, M.; Mayer, J.; Ong, C.W.; Toma, M. Transformative Potential of AI in Healthcare: Definitions, Applications, and Navigating the Ethical Landscape and Public Perspectives. *Healthcare* **2024**, *12*, 125. doi:10.3390/healthcare12020125.
18. Gozalo-Brizuela, R.; Merchan, E.E.G. A Survey of Generative AI Applications. *Journal of Computer Science* **2024**, *20*, 801–818. doi:10.3844/jcssp.2024.801.818.
19. Jamshidi, M.B.; Lalbakhsh, A.; Talla, J.; Peroutka, Z.; Roshani, S.; Matousek, V.; Roshani, S.; Mirmozafari, M.; Malek, Z.; La Spada, L.; Sabet, A.; Dehghani, M.; Jamshidi, M.; Honari, M.M.; Hadjilooei, F.; Jamshidi, A.; Lalbakhsh, P.; Hashemi-Dezaki, H.; Ahmadi, S.; Lotfi, S., Deep Learning Techniques and COVID-19 Drug Discovery: Fundamentals, State-of-the-Art and Future Directions. In *Emerging Technologies During the Era of COVID-19 Pandemic*; Springer International Publishing, 2021; p. 9–31. doi:10.1007/978-3-030-67716-9_2.
20. Mayer, J.; Jose, R.; Bekbolatova, M.; Coletti, C.; Devine, T.; Toma, M. Enhancing patient safety through integrated sensor technology and machine learning for bed-based patient movement detection in inpatient care. *Artificial Intelligence in Health* **2024**, *1*, 132. doi:10.36922/aih.2790.
21. Toma, M.; Wei, O.C. Predictive Modeling in Medicine. *Encyclopedia* **2023**, *3*, 590–601. doi:10.3390/encyclopedia3020042.
22. Braeken, A.; Porambage, P.; Gurtov, A.; Ylianttila, M. Secure and Efficient Reactive Video Surveillance for Patient Monitoring. *Sensors* **2016**, *16*, 32. doi:10.3390/s16010032.
23. Palanisamy, P.; Padmanabhan, A.; Ramasamy, A.; Subramaniam, S. Remote Patient Activity Monitoring System by Integrating IoT Sensors and Artificial Intelligence Techniques. *Sensors* **2023**, *23*, 5869. doi:10.3390/s23135869.
24. El-Bendary, N.; Tan, Q.; Pivot, F.C.; Lam, A. Fall Detection and Prevention for the Elderly: A Review of Trends and Challenges. *International Journal on Smart Sensing and Intelligent Systems* **2013**, *6*, 1230–1266. doi:10.21307/ijssis-2017-588.
25. Lee, T.C.; Shah, N.U.; Haack, A.; Baxter, S.L. Clinical Implementation of Predictive Models Embedded within Electronic Health Record Systems: A Systematic Review. *Informatics* **2020**, *7*, 25. doi:10.3390/informatics7030025.
26. Fernandez, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research* **2018**, *61*, 863–905. doi:10.1613/jair.1.11192.
27. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357. doi:10.1613/jair.953.

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