FALL DETECTION IN ELDERLY USING GAIT ANALYSIS

Richa Pandya
Student# 1009959779
richa.pandya@mail.utoronto.ca

Anya Pedersen
Student# 1007637504
anya.pedersen@mail.utoronto.ca

Krish Chhajer
Student# 1009877705
krish.chhajer@mail.utoronto.ca

Nicholas Eckhert
Student# 1010037568
nick.eckhert@mail.utoronto.ca

—-Total Pages: 8

1 INTRODUCTION

Falls are the leading cause of injury death in adults aged 65 and over, causing about 38,000 deaths and almost 3 million hospital visits in 2021 (for Disease Control & Prevention, 2024). Through the synthesis of current deep learning techniques and gait (walking patterns) based data, our team proposes a new and viable solution to predicting and preventing elderly falls. Gait data surrounding elderly people can be recorded using gyroscopes and accelerometers located in cell phones or other portable technologies. While mobile devices make this data easily accessible, it is often complex, containing time series data and multiple sensors. Neural networks can easily extract and recognize patterns, making it a good tool for analyzing the complex sensor data and returning an accurate reading. Additionally, a neural network's ability to learn over time allows it to scale well with large amounts of data that other models may struggle with.

To accomplish our goal, the team will combine the efficiency of a CNN with the accuracy of an LSTM to create our own hybrid model. The model will receive input from sensor readings (off of the user's phone) and output information that can be used as data for future research as well as issue a notification to nearby emergency responders. The use of a neural network in this way should significantly alleviate the pressure on both the patient and the emergency responders and help fall victims to receive immediate and effective care.

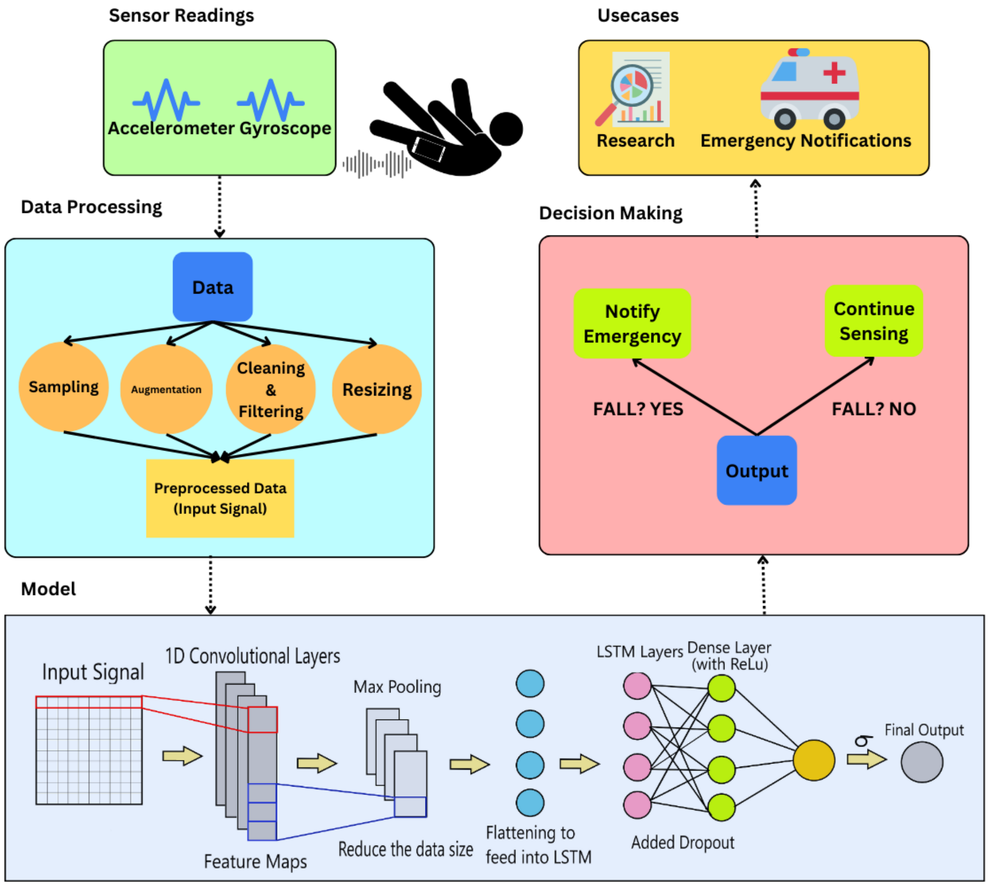


Figure 1: Project Illustration

2 Background and Related Work

Fall detection technology has evolved using deep learning techniques for predicting and preventing falls in the elderly. This section focuses on key advancements in the field, showcasing diverse approaches from different neural network models.

2.1 Gait-based person fall prediction using deep learning approach:

Murthy et al. (2021) proposed a deep convolutional neural network (DCNN) model that analyzes gait analysis images to predict falls in persons with walking disabilities. Their DCNN model achieves 99.1% classification accuracy and a 98.64% prediction ratio, outperforming the ResNet50 and CNN methods. The model detected subtle precursors to falls with significant sensitivity

2.2 Computer Vision and Machine Learning-Based Gait Pattern Recognition for Flat Fall Prediction

Chen et al. (2022) used a motion capture system to detect spatiotemporal gait data from 7 healthy subjects walking with normal gait, pelvic obliquity gait, and knee hyperextension gait. They evaluated CNNs, SVMs, KNNs, and LSTM to classify the three gait patterns. The SVM had the highest accuracy of 94.9%.

2.3 A SMARTPHONE-BASED ONLINE SYSTEM FOR FALL DETECTION WITH ALERT NOTIFICATIONS AND CONTEXTUAL INFORMATION OF REAL-LIFE FALLS

Harari et al. (2021) developed a smartphone based online system that detects falls with 98.6% precision using accelerometer and gyroscope data. They captured and analyzed 16 unscripted falls from 9 senior adults over two years, providing insights into real life fall dynamics.

2.4 DETECTION OF GAIT ABNORMALITIES FOR FALL RISK ASSESSMENT USING WRIST-WORN INERTIAL SENSORS AND DEEP LEARNING

Kiprijanovska et al. (2020) proposed a deep learning approach to detect gait abnormalities using data collected from inertial sensors on both wrists. Their LSTM-based model analyzed the inertial data to identify abnormal gait, and achieved 98.1% accuracy.

2.5 ENIGHTTRACK: RESTRAINT-FREE DEPTH-CAMERA-BASED SURVEILLANCE AND ALARM SYSTEM FOR FALL PREVENTION USING DEEP LEARNING TRACKING

Mao et al. (2023) introduced eNightTrack, a system that uses depth cameras and deep learning for nighttime fall prevention monitoring. Their convolutional pose machine model accurately tracks 14 key body joints from depth images and an SVM then assesses fall risk based on joint angles and velocities. Their model achieved 95.6% accurately without privacy concerns from RGB video.

2.6 FALL DETECTION WITH CNN-CASUAL LSTM NETWORK

Wu et al. (2021) proposed a CNN-Casual LSTM network for fall detection using wearable sensor data from the SisFall Dataset (Sucerquia et al., 2017). Their model uses a CNN to extract spatial features from the sensor, and a Casual LSTM to capture the temporal dependencies. Their model achieved 99.79% accuracy, 100% sensitivity, and 99.73% specificity.

3 DATA PROCESSING

MobiAct Dataset (BMI) contains readings from accelerometer and gyroscope sensors collected from simulated Activities of Daily Living (ADL) and Falls. To simulate these activities, a Samsung S3 mobile phone was placed in the subjects' trouser pocket in a random orientation and realistic falls were performed under guided supervision. The collected sensor readings are stored in a SQLite Database, which includes three txt files per subject trial for sensor readings and orientation. The following preprocessing methods will be used by the team as and when required:

- **Forming the Feature Tensor:** Use a python script to convert the txt files into a single feature tensor for each subject trial. Features will include the timestamp, sensor readings and different orientations.
- **Resampling (Frequency Conversion):** In case of different frequencies, convert samples into the same frequency or if the frequency is too large, downsample (using pandas) (Kaggle's Resampling Notebook) the collected data's frequency for better computation time.
- **Data Cleaning:**
  - **Missing Data:** Check for any NAN Values (using torch) and remove them if it does not affect the window sample. The team may also use Interpolation Techniques (Juewang) to fill in the NAN values in case they affect the window sample.
  - **Noise:** Motion Artifacts are when movement can affect placement of the sensors and introduce noise into the collected data. If the dataset is found to contain substantial noise, the team will analyze the trend of the data and choose an appropriate noise removal technique if needed. Common noise removal techniques (LinkedIn) include Moving Averages, Kalman Filters (commonly used for smoothing motion sensor data), and Frequency domain filters such as Low pass and High Pass.
- **Feature Extraction:** The team will explore the dataset to look for any patterns or correlation between sensor readings using Feature Extraction and Processing. Specific Time-

Domain, Frequency-Domain or Sensor Input features will be explored and based on any observations, may be utilised to expand the feature matrix.

- **Data Augmentation:** To improve model robustness and prevent overfitting, the team will combine other datasets to expand the above dataset:
  - **Human Activity Recognition Dataset** (UCI) contains accelerometer and gyroscope sensor readings recording Activities of Daily Living. While it does not contain any Fall labels, it contains other classes such as Walking and Sitting which can be used to create an equal data distribution across all classes.
  - **Fall vs Normal Activities Dataset** (Kaggle) contains accelerometer and gyroscope sensor data collected by arduino. While this dataset does not contain any timestamp or orientation data, it could be used as a test dataset or used to expand the data in case a different model is pursued.

  In addition to combining the above datasets, Data Augmentation (Nikitin) and Synthetic Data Generation techniques such as Slicing, Shuffling or adding Gaussian Noise could also be useful if needed.
- **Normalization or Standardization:** While combining multiple datasets, different datasets have different ranges and standard deviation. Thus, in order to deal with this, Standardization or Normalization may need to be applied to the datasets. The team may also implement it to improve model accuracy.
- **Segmentation or Window Sampling:** The team will need to choose a fixed size time window for data sampling and create labels accordingly as this is needed for LSTMs.
- Convert the tensor into required shape for input into the model.
- **Splitting the Dataset:** Split the processed dataset into test, train and validation datasets based on common ratios such as 80:10:10 and 70:10:20. The team must try to ensure uniformity in sample size across all datasets for model robustness.

In case of disruption or inability to acquire the desired datasets, the team has also explored the FARSEEING Consortium Dataset (FARSEEING). This dataset contains real-world accelerometer and gyroscope data recorded by body-worn sensors and similar preprocessing techniques can be implemented on this dataset for training.

4 Architecture and Baseline Model

After thorough data preprocessing (segmenting raw inertial sensor data into fixed-length sequences and applying normalization and noise reduction techniques), the data is processed through machine learning models to perform gait analysis for fall detection. Past research has showcased varying models with varying success rates applied for similar tasks. Simpler models such as support vector machines (SVMs) and artificial neural networks (ANNs) have traditionally been applied, but sometimes face challenges with overfitting and capturing long-term dependencies within gait sequences (Murthy et al., 2021).

More advanced models such as convolutional neural networks (CNNs) and long short term memory (LSTM) networks have also shown promise when working with temporal data: a study by Chen et al. (2022) achieved an accuracy of 87.6% for flat fall prediction using CNNs, and an accuracy of 83.6% using LSTMs. Due to their sequential nature, LSTM networks are able to learn complex temporal patterns within gait data and allow for improved fall detection accuracy. On the other hand, CNNs excel at extracting spatial features from data, which allows for efficient recognition of movement patterns.

Given these insights, the proposed architecture for this project is a hybrid CNN-LSTM model. CNNs process the raw inertial data and exact critical features through one-dimensional convolutional layers and max pooling, which reduces data dimensionality without losing critical information (Lilhore et al., 2023) (Abdallah et al., 2021). The flattened output from the CNN feeds into the LSTM layers, which capture the temporal dependencies by maintaining state information over time. This combination allows the model to leverage the strengths of both the CNNs' feature extraction capabilities and LSTMs' strength in understanding sequential patterns. (Abdallah et al., 2021) (Portal-Porras et al., 2023) The final stage requires dense layers that process the output of the LSTM layers (high-level features) using an activation function (such as ReLU) to introduce non-linearity. The output layer provides the probability of a fall using a singular neuron with a sigmoid activation function for binary classification. This hybrid architecture combines the strengths of both architectures and effectively handles the complexities of temporal sensor data to predict falls accurately.

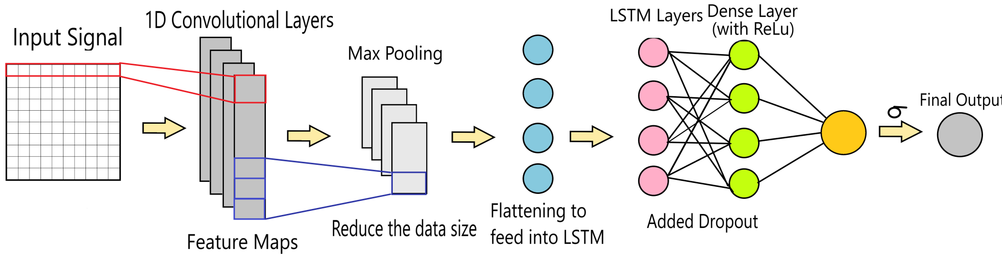


Figure 2: Proposed Model Architecture

To establish a baseline for the model, an SVM model with a linear kernel will also be implemented. The preprocessing steps for the SVM will be consistent with the main model and will include extracting statistical features for each segment which include mean, standard deviation, minimum, maximum, skewness, and kurtosis. Using a linear kernel, the SVM aims to find a hyperplane that separates the two classes - fall and no fall. The linear approach simplifies the model while also offering a robust benchmark for evaluating the performance of the primary model.

5 Ethical Considerations and Limitations

While the proposed fall detection system offers significant potential in enhancing elderly care, it is crucial to critically examine the ethical implications and limitations. The primary concern revolves around privacy as the system continuously monitors individuals' gait patterns - and is essentially tracking their every move. The MobiAct and UCI Human Activity Recognition datasets were collected via smartphones and could raise questions around data anonymity. Moreover, the Fall vs Normal Activities dataset from Kaggle lacks clear information on the data collection methods.

Data bias is another concern to consider. The MobiAct dataset's age range is predominantly between the ages of 20-47, and could poorly represent the target elderly population. (BMI) Typically, older individuals have distinct gait patterns due to age-related changes such as reduced stride length and increased double support time (Hausdorff et al., 2001). Training on data from younger adults creates the risk of algorithmic bias, and potentially leads to higher false-positive rates in the elderly, causing unnecessary interventions and stress.

The datasets also lack diversity. Most participants in the UCI and MobiAct datasets are from European and North American regions and don't account for specific gait variations that could be influenced by factors such as lifestyle and terrain (BMI) (Piwek et al., 2016) - leading to a lower accuracy in underrepresented populations. Additionally, the datasets don't include specifications for individuals with disabilities or using assistive devices, who would have differing gait patterns as well. (Godfrey et al.)

Aside from the data, the model itself also has many limitations. Although the hybrid CNN-LSTM architecture is strong, it may struggle with out-of-distribution data. For example, real world falls often occur in unpredictable ways (slips, trips, health-related falls) that may not have strong correlations with the individuals' gait patterns, could lead to missed detections in actual emergencies. The model could also falsely classify non-fall events that resemble falls (quickly sitting down) due to the datasets' limited range of "normal" activities.

There are also practical constraints for the system. The model runs assuming that a smartphone maintains a consistent placement (either waist or pocket) for accurate readings. However, real life usage of a smartphone varies - and the smartphone could be placed inside a bag or even in hands which could affect the quality of sensor readings.

While this fall detection system shows promise, its deployment demands careful ethical consideration and navigation. Addressing these concerns requires diversifying datasets, refining models for real world variability and conducting extensive testing with a diverse set of individuals.

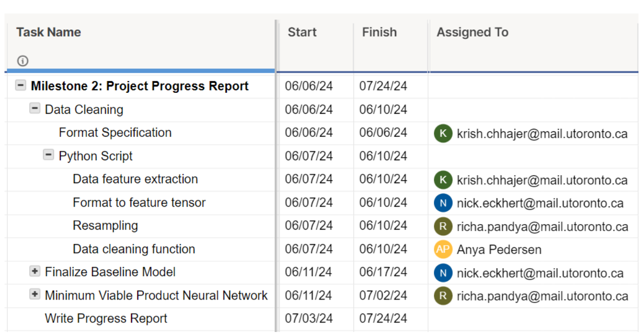
6 PROJECT PLAN

This team will work together by having weekly meetings that discuss progress, any challenges found completing tasks, setting internal deadlines. Distributing tasks is done by the team member in charge of the Major Task during that phase of the project. Those who are incharge of the selected Major Tasks (see table 1) are responsible for breaking down the assigned task to ensure successful execution by the team for the assigned due date.

Table 1: Project break down into Major Tasks and associated task managers.

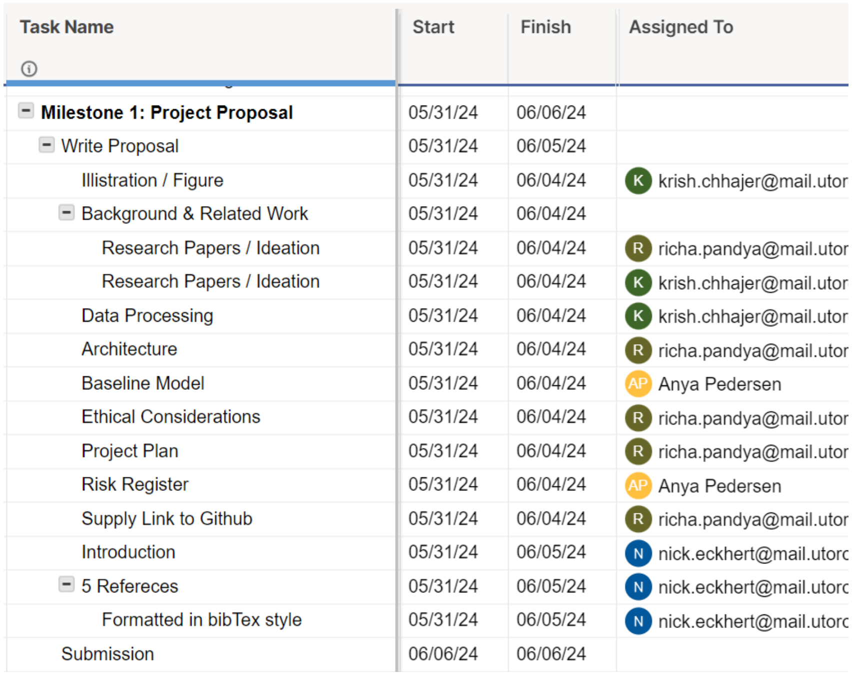
Major Tasks	**Data Collection and Testing**	**Baseline Model**	**Model Integration**	**Written Assignment**
Task Manager	Krish	Nick	Richa	Anya
Project Progress Report Objective (July 4, 2024)	Obtained all or most of the data.	Final Baseline Model	Produce one qualitative and quantitative result	Feasibility of completing project with time remaining

An example of this structure is visible in figure 3, where data processing, as planned by Krish, has been broken down and distributed amongst the team as immediate action items. This structure is used because throughout the project timeline, there will be emphasis on each of these different tasks at different points. The person in charge of the Major Task has already done prior research through the background research, giving them a stronger stance at what needs to be achieved in order for that corresponding task to meet all requirements laid out by this course. This also gives everyone a chance to plan out the project while keeping tasks flexible to individual's timelines.
The purpose of the monday meetings is to break down the large tasks into week-long assignments, distributing the work, and updating the team on tasks completed. Project progress and task break-down is tracked in a SmartSheet Gantt Chart. Previous work completed is displayed from SmartSheet in figure 4. Immediate future tasks are designated in figure 3. Platforms for information sharing are specified in table 2.



Task Name	Start	Finish	Assigned To
Milestone 2: Project Progress Report	06/06/24	07/24/24	
Data Cleaning	06/06/24	06/10/24	
Format Specification	06/06/24	06/06/24	krish.chhajer@mail.utoronto.ca
Python Script	06/07/24	06/10/24	
Data feature extraction	06/07/24	06/10/24	krish.chhajer@mail.utoronto.ca
Format to feature tensor	06/07/24	06/10/24	nick.eckhert@mail.utoronto.ca
Resampling	06/07/24	06/10/24	richa.pandya@mail.utoronto.ca
Data cleaning function	06/07/24	06/10/24	Anya Pedersen
Finalize Baseline Model	06/11/24	06/17/24	nick.eckhert@mail.utoronto.ca
Minimum Viable Product Neural Network	06/11/24	07/02/24	richa.pandya@mail.utoronto.ca
Write Progress Report	07/03/24	07/24/24	

Figure 3: Immediate Action Items and Due Dates



Task Name	Start	Finish	Assigned To
Milestone 1: Project Proposal	05/31/24	06/06/24	
Write Proposal	05/31/24	06/05/24	
Illistration / Figure	05/31/24	06/04/24	K krish.chhajer@mail.utor
Background & Related Work	05/31/24	06/04/24	
Research Papers / Ideation	05/31/24	06/04/24	R richa.pandya@mail.utor
Research Papers / Ideation	05/31/24	06/04/24	K krish.chhajer@mail.utor
Data Processing	05/31/24	06/04/24	K krish.chhajer@mail.utor
Architecture	05/31/24	06/04/24	R richa.pandya@mail.utor
Baseline Model	05/31/24	06/04/24	AP Anya Pedersen
Ethical Considerations	05/31/24	06/04/24	R richa.pandya@mail.utor
Project Plan	05/31/24	06/04/24	R richa.pandya@mail.utor
Risk Register	05/31/24	06/04/24	AP Anya Pedersen
Supply Link to Github	05/31/24	06/04/24	R richa.pandya@mail.utor
Introduction	05/31/24	06/05/24	N nick.eckhert@mail.utorc
5 Refereces	05/31/24	06/05/24	N nick.eckhert@mail.utorc
Formatted in bibTex style	05/31/24	06/05/24	N nick.eckhert@mail.utorc
Submission	06/06/24	06/06/24	

Figure 4: Complete Tasks and Associated Dates

Table 2: Selected platforms for project development.

PLATFORM	PURPOSE	DESCRIPTION
Google Drive	Written Repository	Research repository, version control of written assignments.
GitHub	Code Repository	Python Script Version Control
Whatsapp	Communication	Accessible platform for quick communication and meeting planning.
SmartSheet	Project Tracker	Gantt chart software that allows us to visualize the project timeline and tasks left to be completed.
Zoom	Meetings	Face-to-face and accessible meeting platform for Monday meetings.

7 Risk Register

Table 3 elaborates on the various types of likely risks associated with the project and potential solutions for them.

Table 3: Risk register

PROJECT STAGE / RISK	LIKELIHOOD	SOLUTION
Planning		
Lack of quality data / Noisy data	80% Collecting data with a phone allows for multiple other perturbations to be recorded, including shifting in a pocket, using it, or dropping it.	(1) Collection of multiple data sources, (2) Investigation into different types of data collection including accelerometer data or image of stride data.(3) Develop a filter to clean data and get appropriate frequencies from the test subject's stride.
Development		
Computational resource limitations	20% This is less likely to happen as methods such as sampling can be implemented to reduce function input.	(1)Segmenting data into epochs.(2) Testing function processing time individually before implementing. (3) Sampling data.
Improper transformation of data to abstraction	80% This is likely to happen given human error when writing code	(1) Create simplified data test cases with known output. (2) Have a modular data processing code that can be tested in stages.
Teammate drops course	5% This is a summer course and critical for many degrees.	(1) Assign internal deadlines with tolerance to course due date. (2) Prioritize tasks as a group that have sooner deadlines. (3) Weekly check-ins to monitor when work is falling behind internal deadlines.
Application		
Bias towards particular demographic or device collection method	20% This could happen due to selection of one data set to start with when training, however that likelihood is reduced as we incorporate other data.	(1) Incorporate multiple datasets. (2) Taking our own data.

8 Link to Group Github Repository:

The following is the link to the group's GitHub repository: `https://github.com/rpandya5/gaitanalysis`

References

The mobifall and mobiact datasets — bmi. URL `https://bmi.hmu.gr/the-mobifall-and-mobiact-datasets-2/`.

Uci machine learning repository. URL `https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones`.

M. Abdallah, N. An Le Khac, H. Jahromi, and A. Delia Jurcut. A hybrid cnn-lstm based approach for anomaly detection systems in sdns. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, Aug 2021. doi: 10.1145/3465481.3469190.

B. Chen et al. Computer vision and machine learning-based gait pattern recognition for flat fall prediction. *Sensors*, 22(20):7960, Oct 2022. doi: 10.3390/s22207960.

FARSEEING. The farseeing real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls — farseeing. URL `https://farseeingresearch.eu/the-farseeing-real-world-fall-repository-a-large-scale-collaborative-database-to-c`

Centers for Disease Control and Prevention. Older adult falls, 2024. URL `https://www.cdc.gov/falls/about/index.html#:~:text=Key%20points,injury%20death%20for%20that%20group.&text=In%202021%2C%20emergency%20departments%20recorded,visits%20for%20older%20adult%20falls`.

A. Godfrey et al. Icap: Instrumented assessment of physical capability. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4551273/`.

Y. Harari et al. A smartphone-based online system for fall detection with alert notifications and contextual information of real-life falls. *Journal of NeuroEngineering and Rehabilitation*, 18(1), Aug 2021. doi: 10.1186/s12984-021-00918-z.

J. M. Hausdorff, D. A. Rios, and H. K. Edelberg. Gait variability and fall risk in community-living older adults: A 1-year prospective study. *Archives of Physical Medicine and Rehabilitation*, 82 (8):1050–1056, Aug 2001. doi: 10.1053/apmr.2001.24893.

Jue Juewang. Handle missing values in time series for beginners. URL `https://www.kaggle.com/code/juejuewang/handle-missing-values-in-time-series-for-beginners#time-series-imputation`.

Kaggle. Falls vs normal activities. URL `https://www.kaggle.com/datasets/enricogrimaldi/falls-vs-normal-activities`.

Kaggle's Resampling Notebook. Resampling and frequency conversion. URL `https://www.kaggle.com/code/suruthi41/resampling-and-frequency-conversion`.

I. Kiprijanovska, H. Gjoreski, and M. Gams. Detection of gait abnormalities for fall risk assessment using wrist-worn inertial sensors and deep learning. *Sensors*, 20(18):5373, Sep 2020. doi: 10.3390/s20185373.

U. K. Lilhore et al. Hybrid cnn-lstm model with efficient hyperparameter tuning for prediction of parkinson's disease. *Scientific Reports*, 13(1), Sep 2023. doi: 10.1038/s41598-023-41314-y.

LinkedIn. What are the best techniques for filtering noisy sensor data in control systems? URL `https://www.linkedin.com/advice/0/what-best-techniques-filtering-noisy-sensor#:~:text=One%20of%20the%20most%20common`.

Y.-J. Mao, A. Y.-C. Tam, Q. T.-K. Shea, Y.-P. Zheng, and J. C.-W. Cheung. Enighttrack: Restraint-free depth-camera-based surveillance and alarm system for fall prevention using deep learning tracking. *Algorithms*, 16(10):477, Oct 2023. doi: 10.3390/a16100477.

A. Sampath Dakshina Murthy, T. Karthikeyan, and R. Vinoth Kanna. Gait-based person fall prediction using deep learning approach. *Soft Computing*, 26(23):12933–12941, Sep 2021. doi: 10.1007/s00500-021-06125-1.

A. Nikitin. Time series augmentation. URL `https://towardsdatascience.com/time-series-augmentations-16237134b29b`.

L. Piwek, D. A. Ellis, S. Andrews, and A. Joinson. The rise of consumer health wearables: Promises and barriers. *PLOS Medicine*, 13(2), Feb 2016. doi: 10.1371/journal.pmed.1001953.

K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, O. Irigaray, and R. Garcia-Fernandez. Hybrid lstm+cnn architecture for unsteady flow prediction. *Materials Today Communications*, 35: 106281, Jun 2023. doi: 10.1016/j.mtcomm.2023.106281.

A. Sucerquia, J. López, and J. Vargas-Bonilla. Sisfall: A fall and movement dataset. *Sensors*, 17 (12):198, Jan 2017. doi: 10.3390/s17010198.

J. Wu, J. Wang, A. Zhan, and C. Wu. Fall detection with cnn-casual lstm network. *Information*, 12 (10):403, 2021.