FALL DETECTION USING WEARABLE SENSOR DATA

Inderpreet Singh
Faculty of Applied Science
Simon Fraser University
isa21@sfu.ca

Amandeep Singh Kapoor
Faculty of Applied Science
Simon Fraser University
askapoor@sfu.ca

Prabhjot Singh
Faculty of Applied Science
Simon Fraser University
psa83@sfu.ca

1 Introduction

Falls are the leading cause of injury-related hospitalization among older adults. Over 90% of hip and wrist fractures and 60% of traumatic brain injuries in older adults are due to falls. This project is about detecting fall accurately in real-time by utilizing machine learning classifiers which would be trained using the simulated data collected through trials in the lab.

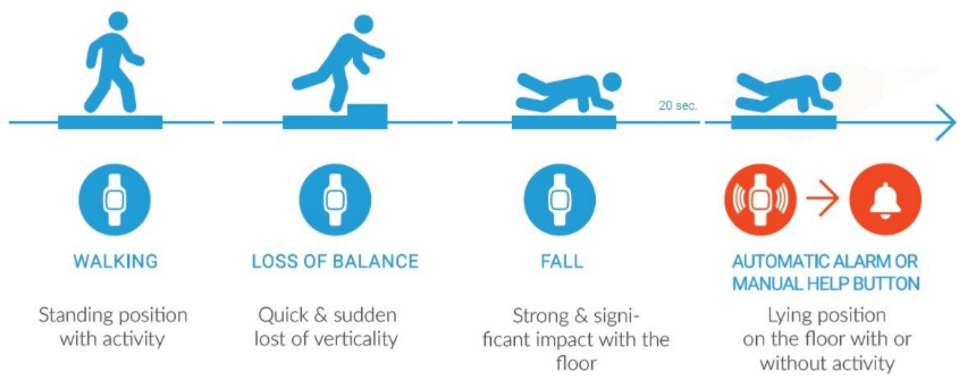

Fig 1: Real Time Phases

Dataset was prepared by simulating 7 types of falls, 5 near-falls, and 8 activities of daily living(ADLs) by each participant. In total, 10 participants performed 3 repeated trials for each category, resulting in 210 falls, 150 near-falls, and 240 ADLs. For fall classification, near-falls and ADLs were combined in the same 'non-falls' category.

2 Background and Motivation

2.1 Background

Research to date has focused primarily on proposing a upper fall threshold (UFT) and lower fall threshold (LFT) in an attempt to optimize the balance of false positives and false negatives. The UFT showed 100 % sensitivity and 100 % specificity, while the LFT provided 100 % sensitivity and 91 % specificity. Similarly, Kangas (experimentation) attached a tri-axial accelerometer at the waist, wrist, and head of volunteers who performed simulated falls and ADLs in the laboratory. Their algorithms considered the pre-impact, impact, and post-impact phases of the fall, separately and in combination, and achieved up to 100 % specificity and 95 % sensitivity, based on a single sensor mounted at the waist.

Limitation: Despite exhibiting high classification accuracy in laboratory experiments, inertial sensor-based fall detection systems have yet to achieve high market penetration. One barrier to their acceptance is the lack of evidence of their effectiveness in real-world falling scenarios in older adults.

2.2 Motivation

The motivations behind this project are-

- Accurate "Fall Detection" is a very important problem which has a great significance especially in the old age people.
- It is the major cause of injury-related hospitalization along with trauma of "long lie", which is due to the inability of getting back up without any assistance.
- Half of the elderly people who experienced a long lie (for an hour or more) passed away within 6 months, even if no direct injury occurred due to fall.
- Given the current demographic dividend of the developed countries, this becomes even crucial to have a system of this kind which can detect and alert the care providers if there is a fall.

3 Problem Statement

- To utilize all the recorded body kinematics from multiple sensors mounted on different body parts.
- Apply advanced machine learning algorithms in order to achieve better sensitivity/specificity.
- Optimizing two requirements which are Sensitivity (ability to detect actual fall) and Specificity (ability to avoid False Positive which could desensitize the receiver) i.e. how to identify Optimal window size or data points corresponding to each trial.
- We want to achieve no false positive and still have acceptable true positive rate (>90%).
- To simulate real life scenario in which data is generated continuously and we need to find a way to handle the data streaming, then apply trained machine learning classifiers to detect fall.

4 Data Science Pipeline

This project data flow is broadly divided in to 2 categories: **Offline Model Training** using Batch Processing and **Real-Time prediction** utilizing trained Model.

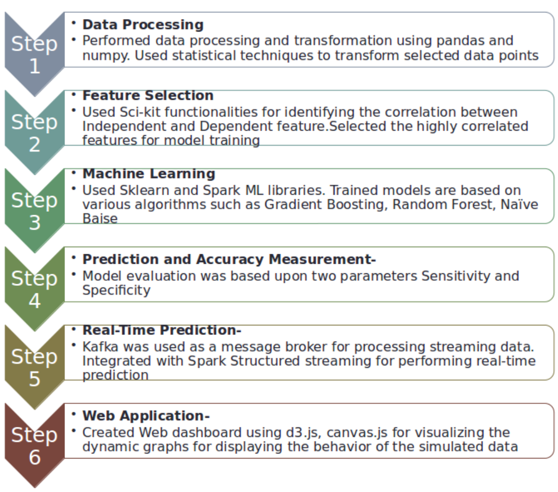

Fig 2: Data Pipeline

In the offline model training, regular batch processing concept is utilized. That is loading the batches/generated data in to Cassandra/HDFS and utilizing Spark ML for training the classifier based on different Machine Learning Models like Random Forest, Gradient Boost.

For real-time prediction, we have used Kafka message broker for storing the stream of data getting generated in real-time

Spark Structured Streaming is used to consume the data in real-time from the message broker and load the already trained classifiers for making prediction in real-time.

Once prediction is made, spark streaming would act as a producer and store the data in to a different topic in Kafka where it can be used for web application. We have also considered a model for **Active Learning**, where the real-time data sitting in Kafka would be loaded in to Cassandra for Batch Processing and Offline training.

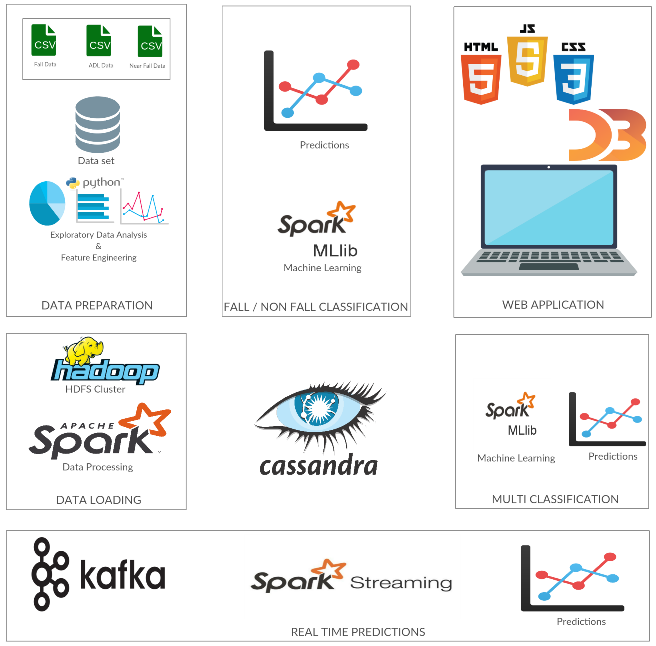

Fig 3: Data Pipeline

5 Methodology

5.1 Understanding Data

- The current trial data is taken from 10 subjects across 3 categories of Fall, Near Fall and Activity of Data Living (ADL).
- Each of these categories are divided into subcategories and there are 3 trials corresponding to each of them.

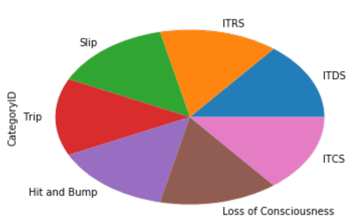

Fig 4: Fall Subcategories

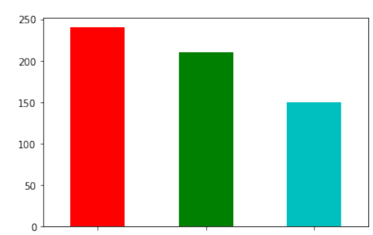

Fig 5: Multi-class Data Distribution

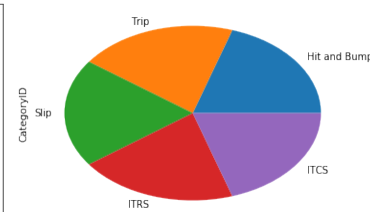

Fig 6: ADL Subcategories

5.2 Exploratory Data Analysis (EDA)

- The data for each trial is collected for duration of 15 s. This was taken from the tri-axial accelerometer mounted at 5 different body part, recording body kinematics (acceleration, angular velocity, magnetic field).
- The line graphs representing the flow of the collected body kinematics – Acceleration, Magnetic Field and Angular Velocity are given below-

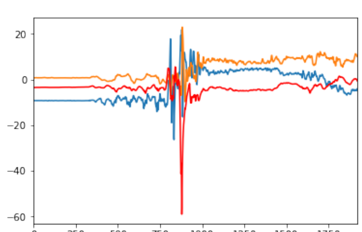

Fig 7:Acceleration

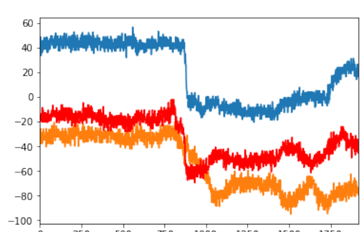

Fig 8: Magnetic Field

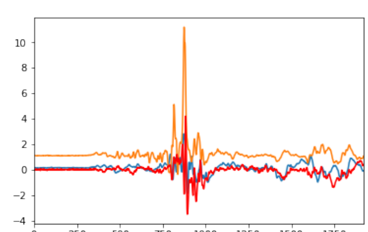

Fig 9:Angular Velocity

5.3 Feature Engineering

For training, only those data points are considered which are consistent with the type of the trial i.e. Fall, Near-Fall and ADL. These data points are then transformed using various statistical techniques like mean, variance etc. The feature selection was done using two methods-

- **Variance Threshold-** baseline method used for feature selection. It removes all features whose variance doesn't meet some threshold.
- **Corr**- to find the correlation between independent and dependent variables. Top-20 features were selected for Fall/Non-Fall Classification (fig 10) and Top-10 features for Multi Classification (fig 11).

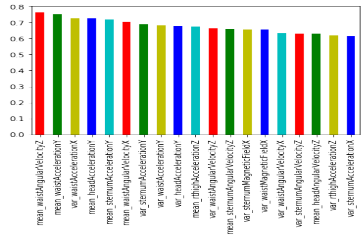

Fig 10: Top20 Features

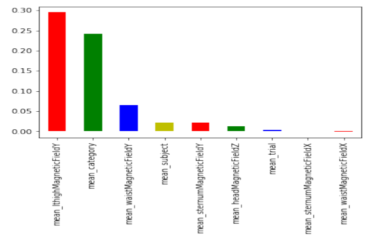

Fig11:Top10 Features

5.4 Machine Learning

Advanced machine learning algorithms were applied in order to achieve better sensitivity/specificity. These techniques include Logistic, Decision, Gaussian Naïve Bayes, Ada Boost, Gradient Boost, Random Forest. Then the trained model was saved to simulate real-life scenario in which sensor data is continuously generated in streams.

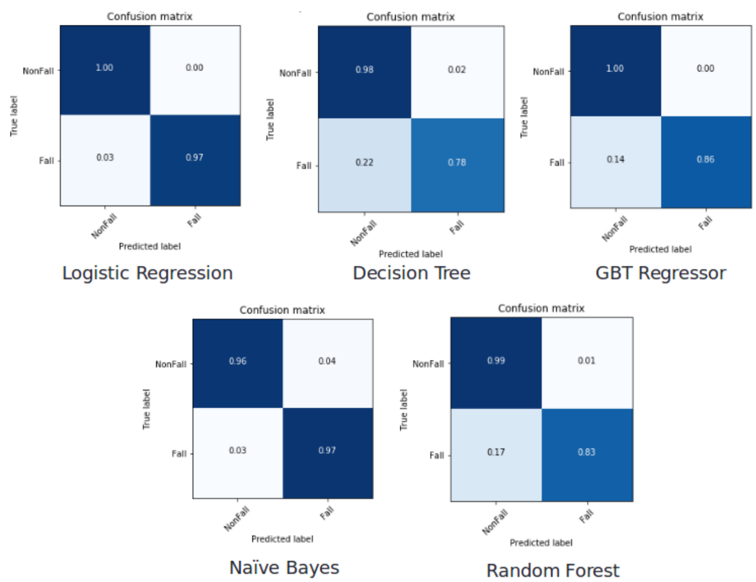

Fig 12: Confusion Matrices corresponding to different classifiers

6 Evaluation

All of the machine learning classifiers were trained and evaluated for various window sizes in-order to get the maximum specificity and sensitivity corresponding to the optimal window size. Based upon the evaluation on different window sizes and models, the best model is found out to be Majority Voting and optimal window size as 1.5 sec. The best sensitivity and specificity corresponding to the major voting is 96.29% and 95.72%.

Sensitivity Analysis corresponding to different window sizes							
Window Size(in sec)	Logistic	DecisionTree	GNB	Adaboost	GradBoost	RForest	MajorityVoting
1	91.176	97.05	91.176	88.23	94.17	91.17	91.176
1.5	98.4	96.26	96.29	97.5	96.29	96.29	96.29
2	93.75	87.5	97	96.8	93.75	96	96
2.5	97	77	97	94	86	83	91

Specificity Analysis corresponding to different window sizes							
Window Size(in sec)	Logistic	DecisionTree	GNB	Adaboost	GradBoost	RForest	MajorityVoting
1	97.27	97.27	95.45	96.36	95.45	96.36	96.36
1.5	94.87	98.29	88.88	99.145	95.72	96.58	95.72
2	98.2	97.3	92.85	99.1	99.1	98.214	98.2
2.5	100	98	96	99	100	99	99

Fig 13: Sensitivity and Specificity of different Machine Learning Classifier

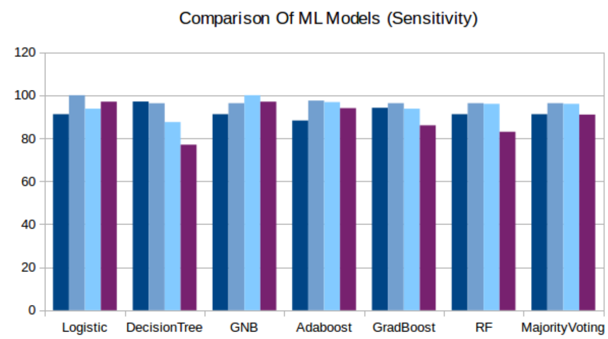

Fig 14: Comparisons of Machine Learning Classifiers (Sensitivity)

7 DATA PRODUCT

This project can be transformed in to a fully functional application where the data is recorded from the sensors and streamed(offloaded) to the Cloud. The trained machine learning classifier would be applied on the streaming data for real-time detection and subsequent scripts would activated for taking the required action in case of a classified Fall. We were able to train machine learning classifiers with a very high accuracy, applied trained model on streaming data and developed a website for sharing the analysis.

8 LESSONS LEARNT

- In this project we had to identify the optimal time window in which the event was occurring. This time window was identified using EDA and domain knowledge.
- The data was inserted in batches by using the concept of checkpoints as aggregated data could not be handled in a single dataframe.
- Publishing the data at the similar rate to the one collected from the actual trials in order to keep the stream rate close to real-time using Kafka.
- Running Machine Learning classifiers on different window sizes to identify the optimal window size and getting comparable results to the research done so far.
- D3.js can be integrated with Canvas.js for generating dynamic graphs.

9 SUMMARY

This project aims at detecting fall in real time so that in the event of fall, caretakers can be informed and the impact of fall on older adults can be minimized. Data collected from trials conducted on 10 different specimen in the lab was used to train various Machine Learning classifiers. The project covers the Exploratory Data Analysis of the collected data to find out the optimal time window which has to be fed into the Machine Learning classifiers to get the best classification results. We have also explored the usage of these classifiers on the streaming data which has more practical significance. For future scope, we have considered applying the concept of Active Learning for improving the model in real-time.