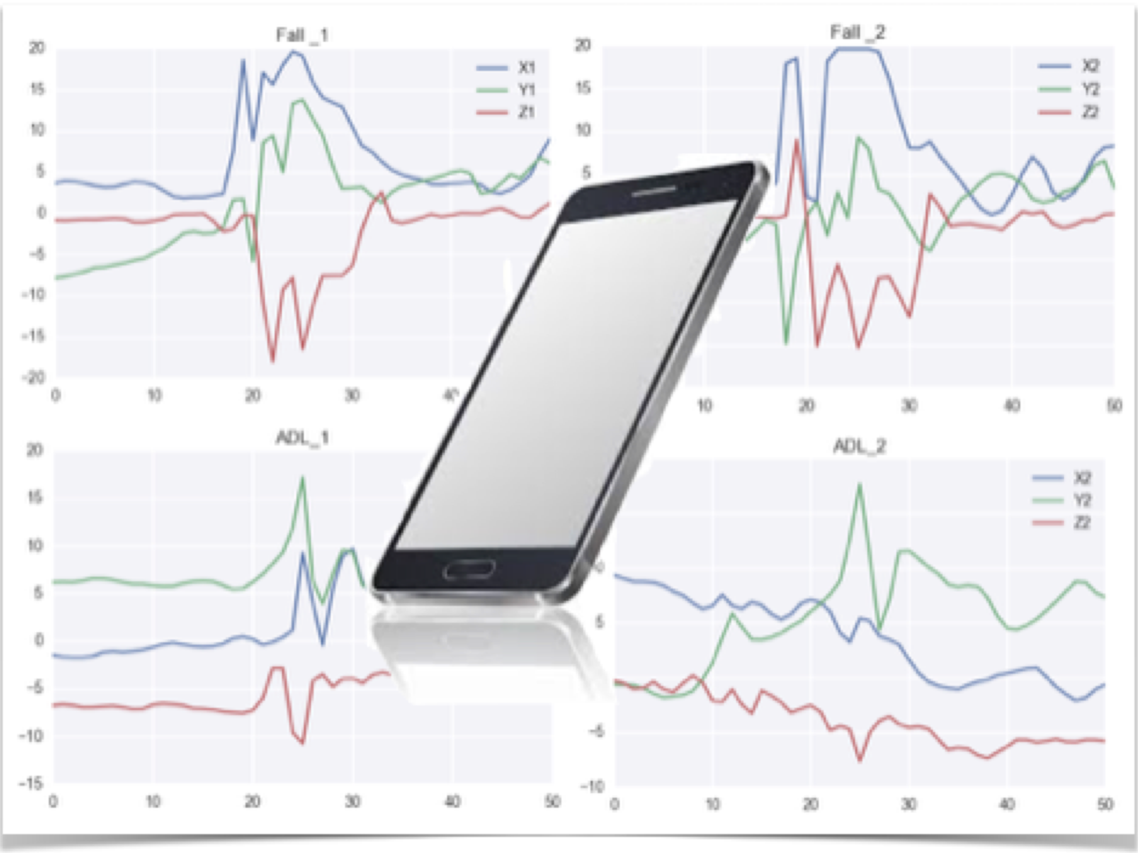
ASSISTIVE TECHNOLOGIES

Smartphone-based Fall Detection System



Project Report

Prepared by: Anna Zamaraeva
June 2017

Table of Contents

1 OVERVIEW

1.1 Introduction

Bluebird Care (http://www.bluebirdcare.co.uk) is a homecare provider, who specialises in working with customers requiring social care in their own home. Typically, Bluebird customers are elderly and less-able people living independently. Bluebird provides support services to keep their customers safe and comfortable in their own homes.

Many Bluebird customers are subject to a higher risk of accidental falls. When a fall occurs, a prompt notification helps to provide quick help and reduce potential injuries.

Bluebird Care uses an *indoor* telecare system, consisting of a alert button device (a pendant or a bracelet) and a central unit plugged into the phone mainline. In case of a fall, a customer can press a device button to send a signal to the central unit which will dial into the care service. The device works inside the house (if walls are not too thick for a signal to reach the central receiver), but does not work in a garden and does not work outdoors.

Bluebird Care is interested in a fall notification system that can be used ***outdoors.*** They would like to have a system that will support their customers when they are spending time in a garden, taking a dog for a walk or going to visit friends and family.

1.2 Proposed solution

With modern technologies, a smartphone can serve as a detection system for the falls. Smartphones are wide spread, do not introduce any additional costs, can be used in any place and are accepted by people as a part of everyday life. Smartphones have inbuilt sensors: an accelerometer and a gyroscope, data from which can be used for falls detection.

An effective smart-phone based system can provide multiple benefits: (1) automatic notification of occurred falls; (2) promptness in order to provide quick assistance and (3) communication capabilities in order to alert the caregivers.

1.3 Project goal and results

The goal of this project is to serve as a proof of concept. We train a machine-learning algorithm to classify human activities and detect falls using patterns recorded by smartphone accelerometers. We use a publicly available dataset of accelerometer data, extract various feature sets and then train several classifiers.

The model demonstrated a respectable performance with the probability of fall detection above 90% , and false alarm rate of 3%. We conclude that using smartphones is a feasible solution.

In what follows, we present the technical analysis in chapters 2 – 4, schematically outline a proposed smartphone-based fall detection system in chapter 5, and provide our recommendations in chapter 6.

2 PROJECT DATA

2.1 Accelerometer Datasets for Human Activities

Recently several accelerometer datasets for human activities have been collected by researchers worldwide and made publicly available. The datasets can be broadly divided by two criteria: a type of sensor used in data acquisition (a smartphone or a specialised device) and inclusion of falls into a dataset. A comprehensive review of the accelerometer datasets is available in (Igual, Medrano, & Plaza, 2015)

For the purpose of the project, we evaluated available datasets by the following requirements:

a) data captured by smartphones in the pockets of study participants, rather than by specialised device;
b) a large number of simulated falls in a dataset;
c) a wide range of human activities and a rich variety of study participants.

Based on these criteria, we selected the UniMiB SHAR dataset created by researchers of University of Milano-Bicohcca, available at http://www.sal.disco.unimib.it/technologies/unimib-shar/ .

The UniMiB SHAR dataset is a labelled, rich and complete collection of acceleration patterns and can serves as a good base for conducting data experiments. A comprehensive description of the dataset can be found in (Micucci, Mobilio, & Napoletano, 2017)

2.2 UniMiB SHAR Dataset

The UniMiB SHAR dataset contains triaxial acceleration data captured by smartphones located in the pockets of participants performing a wide range of activities. Recorded daily activities and simulated falls are performed by a large number of subjects varying in age, gender and physical characteristics. The dataset contains over 7,000 activities samples performed by 30 subjects, mostly females, of ages ranging from 18 to 60 years.

The dataset is composed of

- 5,314 samples of daily activities: walking, jumping, running , sitting down, standing up, lying down, getting up, going up/down;
- 1,699 simulated falls of 8 different types.

The dataset is imbalanced with falls being a minority class. So classification algorithms can likely benefit from rebalancing techniques applied to the data.

2.3 Data Acquisition and Preprocessing

The data was acquired using a Samsung Galaxy smartphone equipped with a BMA220 acceleration sensor. This sensor is a triaxial low-g acceleration sensor with digital output which allows measurements of acceleration in three perpendicular axes. The sampling rate for data recording was about 50 Hz. The accelerometer signal for each time instant is made of a triplet of numbers (x, y, z) that represents the accelerations along each of the 3 Cartesian axes.

From the labelled recorded accelerometer data, the UniMiB researchers extracted a 1 sec signal window for each time a peak was found, based on the following conditions:

- the magnitude of the signal $m_t$at time t was higher than 1.5 g;
- the magnitude m_{t-1} at the previous time instant (t-1) was lower than 0.

Each signal window of 1 sec was then cantered around each peak. Since the acceleration was recorded with a 50 Hz sampling frequency, the data for each sample is made of 3 vectors (one for each acceleration direction) of 51 time points. Data samples for various daily activities and fall types are shown in Figure 1.

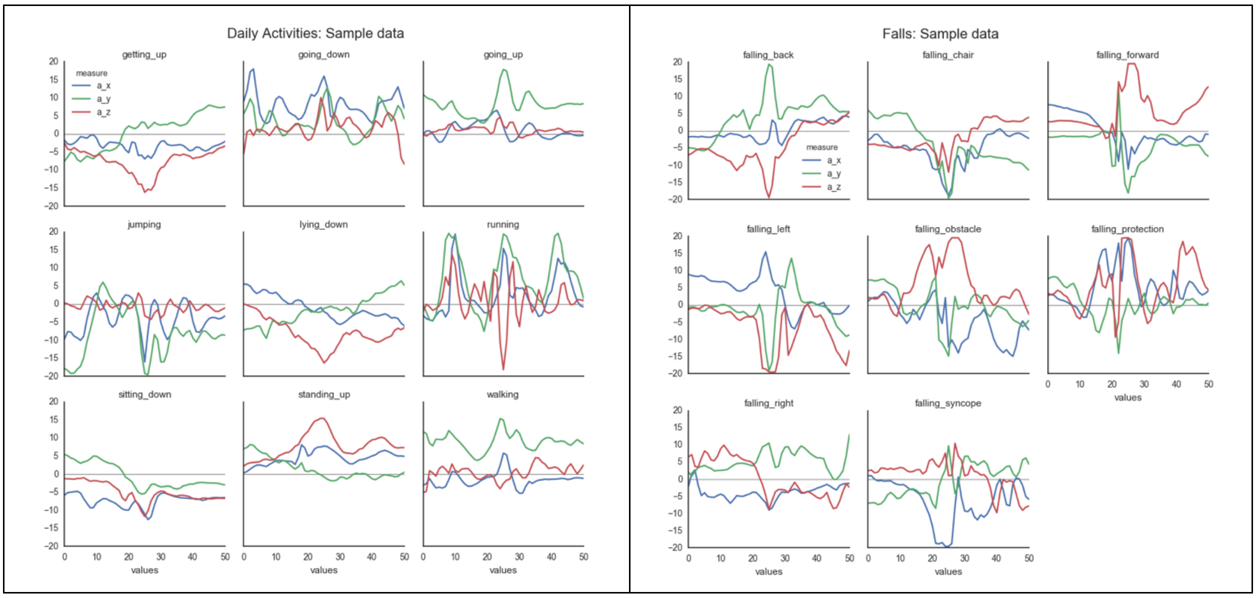


Figure 1. Sample Data for Daily Activities and Falls

3 EXPLORATORY DATA ANALYSIS

3.1 Extracted Features

Given that the nature of the problem lies in the physics domain and is related to detecting certain movements, it makes sense to turn to physics measures for feature extraction. So we came up with four different feature sets based on various physics measures and their summary statistics

Recall that the original data is a timeseries of acceleration projected along 3 Cartesian axes:

$$Triax\_Accel(t_i) = \{a_x(t_i)\,, a_y(t_i)\,, a_z(t_i)\},\ i = (0,..,50).$$

Given that the data is recorded by a smartphone and it can easily move and turn in the pocket, one can think that acceleration directionality is perhaps less important, while the magnitude can be a significant feature. So we calculate the acceleration magnitude, as

$$Magn(t_i) = \sqrt{a_x(t_i)^2 + a_y(t_i)^2 + a_z(t_i)^2}, \quad i = (0,..,50).$$

Additionally, falls are likely to be associated with sharp changes in the movement. In physics, the rate of change of acceleration is defined as jerk ; that is, the derivative of acceleration with respect to time. So we calculate a triaxial jerk by differencing the acceleration timeseries:

$$Triax\_Jerk_k(t_i) = a_k(t_i) - a_k(t_{i-1}), \qquad k = (x, y, z).$$

We also calculate a jerk from acceleration magnitude, as

$$Magn\_Jerk(t_i) = Magn(t_i) - Magn(t_{i-1}) \ .$$

To reduce dimensionality, we aggregate acceleration and jerk timeseries across time by summary statistics: mean, standard deviation, minimum and maximum of timeseries.

As a result, we arrive at four feature sets summarised in Table 1, where they are ordered by number of features.

*Table 1: **Feature Sets Summary***

	Feature Set	Description	N feat
1	Magnitude_Stats	Summary statistics for magnitudes of acceleration and jerk	8
2	TriaxAccel_Stats	Summary statistics for triaxial acceleration and jerk	24
3	Magnitude_Timeseries	Time series of acceleration magnitude	51
4	TriaxAccel_Timeseries	Time series of triaxial acceleration (raw data)	153

We will use these four feature sets as a base of analysis throughout the rest of the paper.

3.2 Data Clustering

To get an insight into possible groupings and distances between classes we use hierarchical clustering. For each activity, we calculated its mean sample to use as a class representative and then built the hierarchy of activities. The resulting dendrograms for each of the four feature sets are shown in Figure 2.

The dendrograms tell an interesting and compelling story. In the first three sets, running and jumping stand apart from other activities, which makes an intuitive sense. As intense sport-related movements, jumping and running can indeed differ in nature from more relaxed activities of daily living.

Also, a fall group and a daily activity group look to be apart. If we remove jumping and running from consideration, we can cut the dendrograms across dashed green lines and daily activities and falls get nicely separated, which is a promising picture for a classification task.

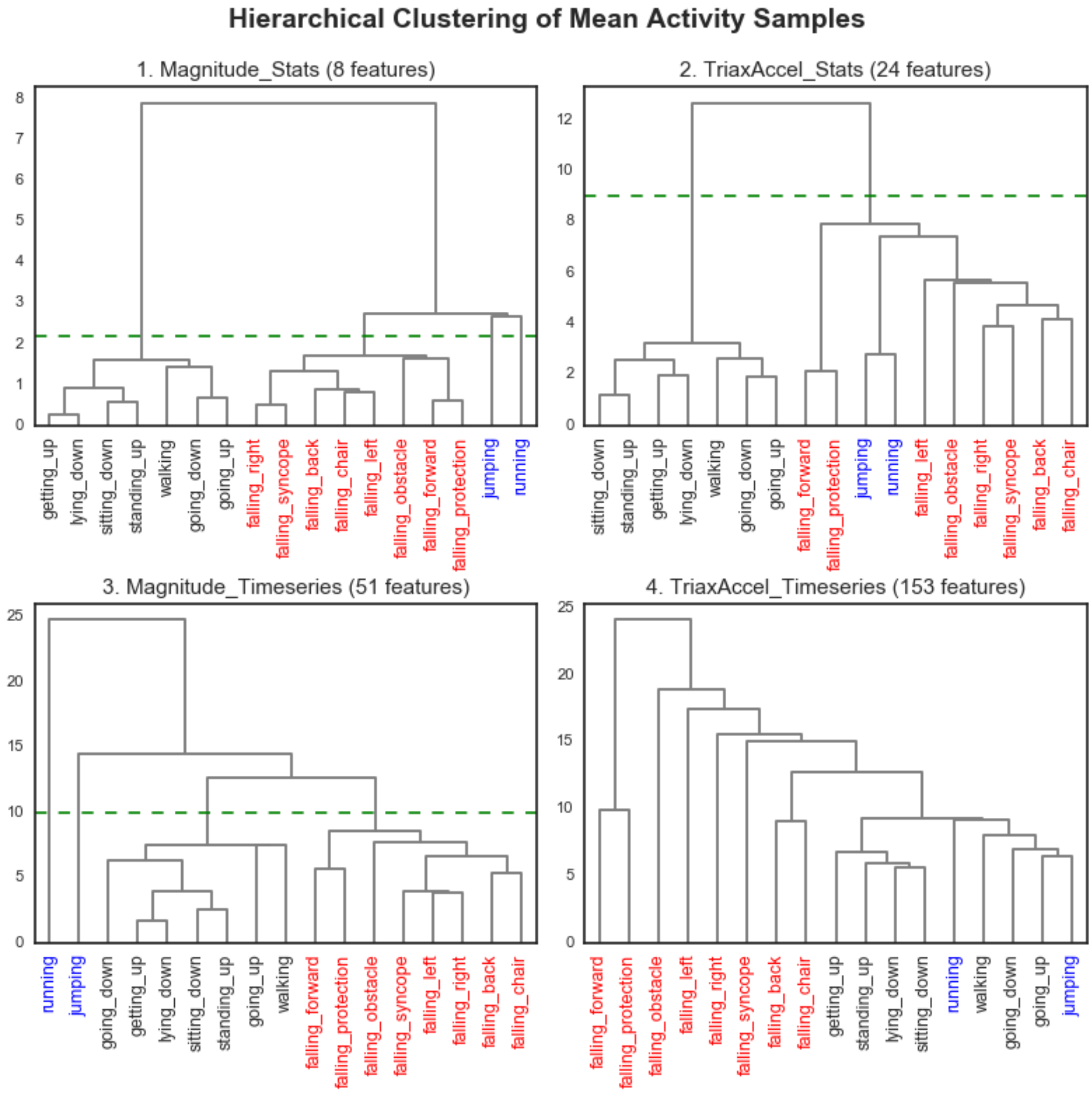


Figure 2. Hierarchical Clustering of Mean Activity Sample

The fourth dendrogram, based on triaxial timeseries, shows a less clear picture. The daily activities still form a definite group. However, different types of falls seem look disparate and get recombined only at the top. This can be an effect of a high dimensionality of the space (153 dimensions) where points are generally far apart from each other.

3.3 Separability in PCA Space

Motivated by the activities dendrograms, we now look into separating data into 4 categories: Running, Jumping, Daily Activities and Falls.

For each feature set we perform a PCA decomposition and then project the data into first two principal components. The resulting scatter plots are shown in FIGURE 3, where data points are plotted with different colors corresponding to the data groups.

Once again, the clustering look promising and the groups seem largely separable. Falls (a red group) are generally shifted from daily activities (a green group). Running (blue) in many cases also look separate, jumping (yellow) is less so.

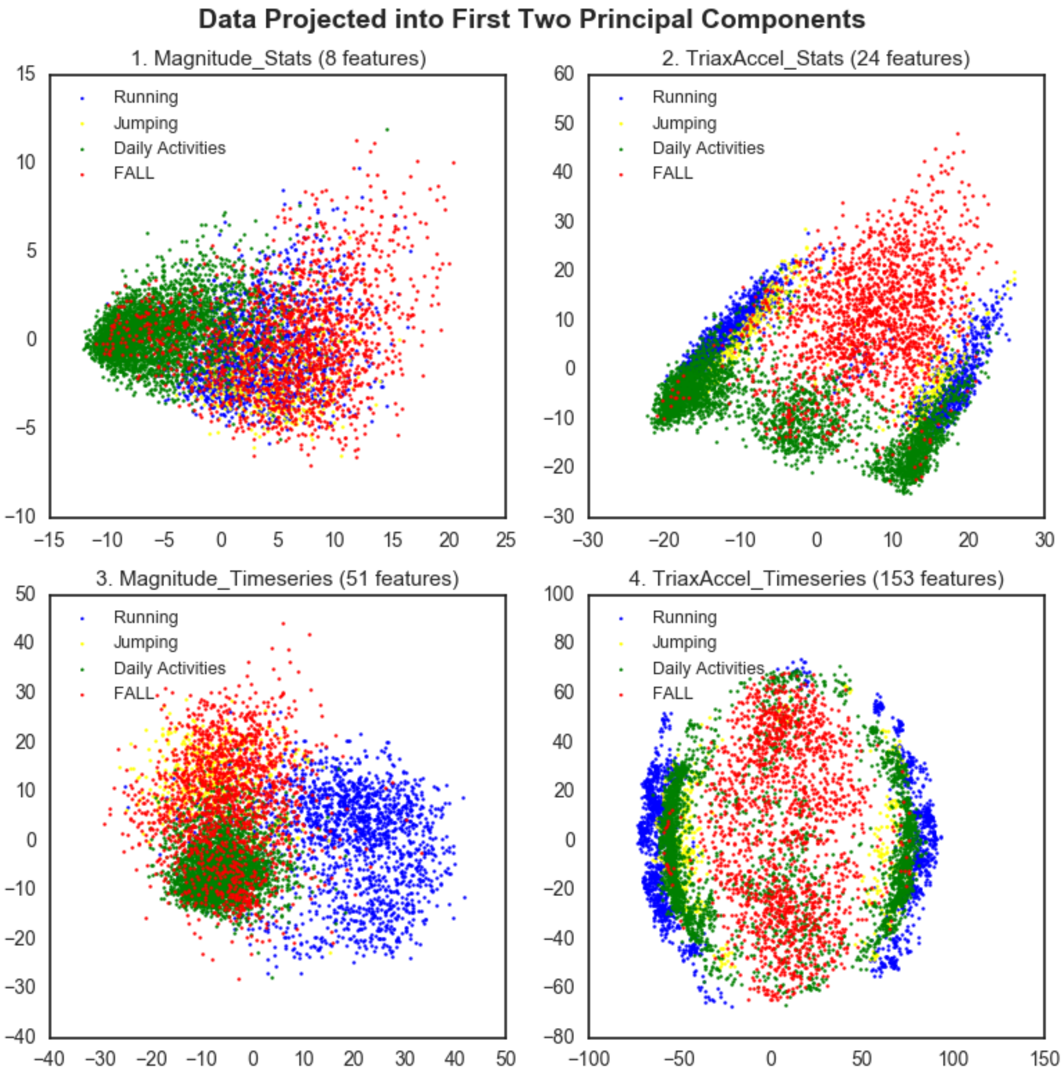


Figure 3. Separability of Classes in PCA space

One can also notice an interesting symmetrical pattern in the triaxial sets 2 and 4, where the parts of groups look to be reflected around zero with one sub-cluster on the negative side and another sub-cluster on the positive side. This can be an indication that a sign does not really matter, and if the data made sign-less (e.g. squared) then the sub-clusters will recombine into a single group.

An additional observation is that the clusters in Set 1 and Set 3 look to be separable by straight lines, so these sets can probably work well with a linear classifier, such as a logistic regression. However, for Set 4 the boundary between classes is clearly non-linear, which likely calls for the use of a non-linear model, such as a Support Vector Machine.

4 CLASSIFIERS FOR FALL DETECTION

4.1 Effect of Sport-related Activities in the Dataset

As seen from dendrograms in 3.2, sport-related activities of running and jumping appear to form distinct groups of their own. Also clustering graphs in 3.3 show that these sport activities, especially jumping, can potentially create a confusion with falls (which is reasonable given that jumping is an intense movement in a vertical direction, i.e. somewhat similar to falling).

Given that the target audience for fall detection is older and less mobile people, we can assume that sport-related activities are likely to be less relevant in our setting. So we look into splitting off sport-related activities to see if excluding running and jumping from the data can give a boost to the classifier performance.

We do this test by using a Logistic Regression classifier for fall detection in each feature set with (a) including and (b) excluding jumping and running sample. FIGURE 4 shows resulting Receiving Operator Characteristic (ROC) curves and corresponding areas-under-curves (AUC).

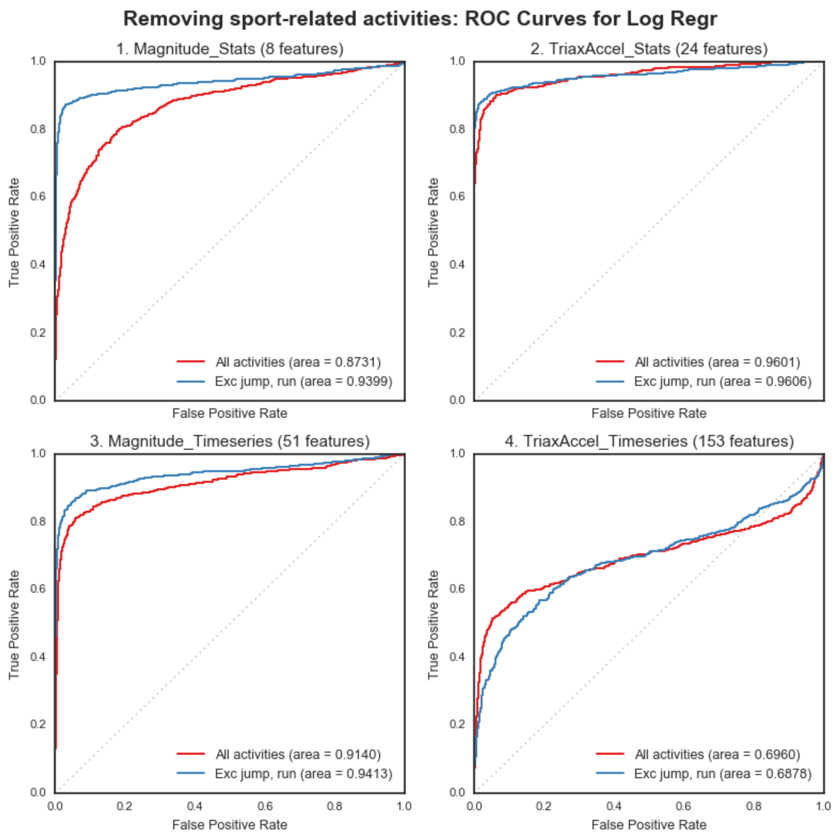


Figure 4: ROC Curves for Excluding jumping and running samples

As expected, removing running and jumping gives a boost to the classifier performance. The ROC curves for fall detection in general become higher and steeper when running and jumping samples are removed from consideration.

We also note that even a simple linear classifier, such as Logistic Regression, performs reasonably well for the task at hand. A notable exception is classifying on the feature set 4, where Logistic regression performance is strikingly poor. This is likely an effect of the high dimensionality and non-linearity of the boundaries in this feature space, as we have seen before in FIGURE 3 .

4.2 Two-class classifiers for Fall Detection

Armed with a better data understanding, we can now proceed to building classifiers for fall detection. We remove running and jumping records and also rebalance the data by oversampling the falls by duplicating the existing fall records.

Using the same feature sets as before, we build four different classifiers based on: Logistic Regression, k-Nearest Neighbour (k-NN), Random Forest and Support Vector Machine (SVM). We use machine learning methods from Python-based 'SciKit Learn' package and employ 3-fold cross validation to search for optimal model parameters, as summarised in Table 2 . Resulting ROC curves and performance metrics are shown in Figure 5 and Table 3.

*Table 2. **Classifiers and their parameters***

Classifier	Sklean Method	Parameters
a. **Logistic Regression**	LogisticRegression	Cross-validation for regularization parameter C and for penalty type { L1/ L2}
b. **k-NN**	KNeighborsClassifie	Cross-validation for n_neighbors
c. **Random Forest**	RandomForestClassifier	Log2 maximum features; cross-validation for n_estimators
d. **SVM**	svm.SVC	Radial Basis Funtion kernel; cross-validation for regularization parameter C

The classifiers have performed quite well. The best models are the Random Forest classifier applied to Magnitude Timeseries and SVM applied to Triaxial Acceleration Timeseries. The classifiers have a 90-93% probability of fall detection with a 2-4% probability of false alarm on a given dataset. We note that the reported probability of false alarm was calculated using dataset samples as a base. The data in the dataset are high-intensity samples just by the dataset construction (samples are pre-selected by the peak acceleration exceeding a certain threshold). An actual false alarm rate, i.e. the rate at which the false alarm can be triggered in the normal mostly-calm daily activities is unknown here, as we don't have the necessary information. This can be estimated later either from finding more about how the data was recorded or by determining it from other sources.

The classification results have successfully proved the concept: the falls indeed can be detected from the accelerometer data and smartphones indeed can indeed serve as a foundation for building a fall-detection system.

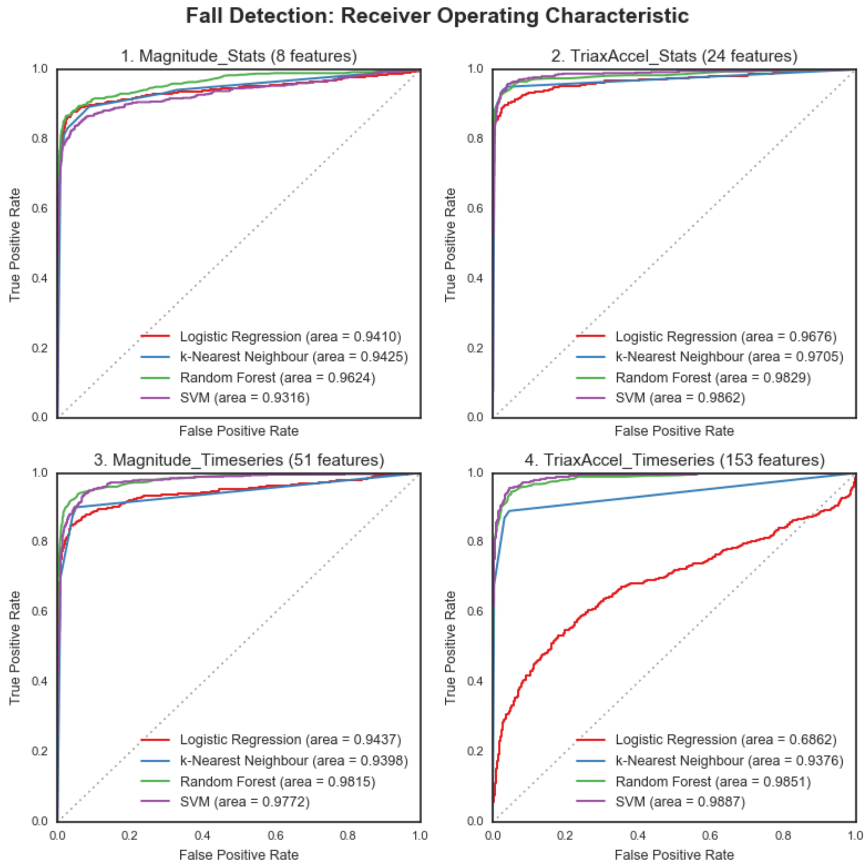


Figure 5. Fall Detection ROC curves

Table 3. ***Model Performance***

	Feature Set	Classifier	Prob of Detection	Prob of False Alarm	Classifier Accuracy	Area Under Curve
1	**Magnitude_Stats** *(8 Features)*	Logistic Regression	89.4%	7.6%	91.4%	0.9410
		k-Nearest Neighbour	88.1%	7.6%	91.0%	0.9425
		Random Forest	87.6%	4.0%	93.2%	0.9624
		SVM	84.1%	5.7%	90.9%	0.9316
2	**TriaxAccel_Stats** *(24 Features)*	Logistic Regression	90.8%	5.6%	93.2%	0.9676
		k-Nearest Neighbour	94.1%	2.8%	96.2%	0.9705
		Random Forest	92.0%	2.2%	95.9%	0.9829
		SVM	93.5%	2.5%	96.1%	0.9862
3	**Magnitude_Timeseries** *(51 Features)*	Logistic Regression	87.2%	6.8%	91.2%	0.9437
		k-Nearest Neighbour	87.6%	4.2%	93.1%	0.9398
		Random Forest	90.3%	2.5%	95.0%	0.9815
		SVM	90.3%	4.9%	93.5%	0.9772
4	**TriaxAccel_Timeseries** *(153 Features)*	Logistic Regression	64.7%	32.3%	66.7%	0.6862
		k-Nearest Neighbour	87.2%	3.2%	93.5%	0.9376
		Random Forest	93.4%	4.4%	94.8%	0.9851
		SVM	91.1%	2.7%	95.2%	0.9887

5 SMARTPHONE-BASED FALL DETECTION SYSTEM

We propose to implement a fall detection system as an app running on users smartphones. The app will be processing the accelerometer data in a real time, and if a fall is detected and confirmed, the app will alert care services. A high-level schematic design of the system is shown in Figure 6.

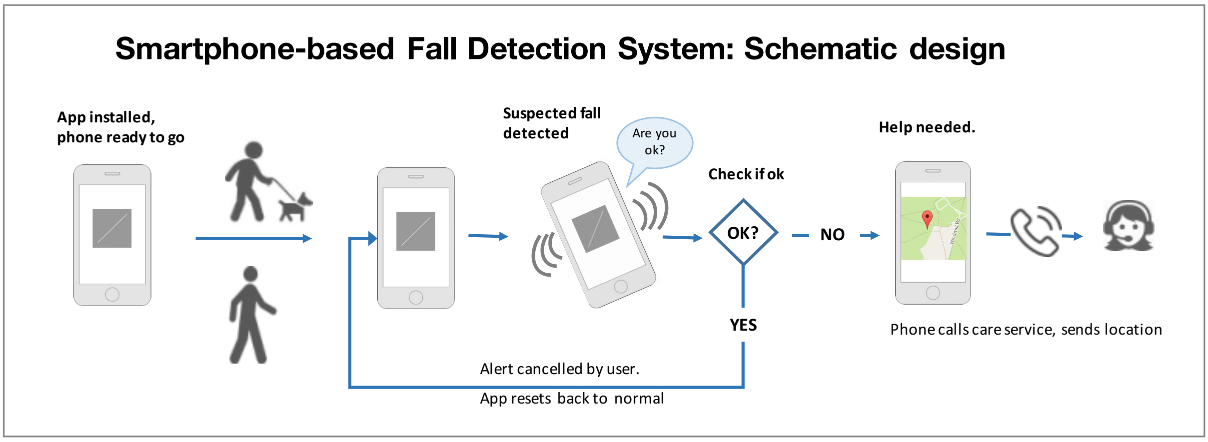


Figure 6. Smartphone-based Fall Detection System

The key features of the system are as follows.

A. **The system is particularly suitable for an outdoor use**:
   - Smartphones have an in-built location positioning service. If an alert is triggered, the app will send a current phone location to the care service, so that a person can be quickly found and necessary help promptly provided.

B. **Interaction with a user**:
   - If a suspected fall detected, the phone will display a pop-up message with a vibration alert and will also use a voice message asking if help is needed. In case of a false alarm, the user can cancel the alert;
   - If the help is required, the app will automatically call the caregivers and they will communicate to the user over the phone to asses the situation and provide the necessary help.

C. **Continuous iterative machine learning**:
   - the app will come with a preloaded standard fall-detection model trained on the previously collected data.
   - Once a user starts using the app, new accelerometer data will get captured and used for training a personalised model, adapted to the user; Data collected for multiple users then can be used to enrich the standard model.

Recommendations and practical steps towards implementation of the system are outlined in the next section.

6 RECOMMENDATIONS

Given a successful proof of concept, we recommend the following action and steps:

1. **Collect more data.** The data can be captured using available smartphone apps, such as Sensor Kinetics, or Accelerometer app. A particular data of interest is
   - walking outdoors;
   - dropping a phone when no actual fall occurs;
   - data recorded by a phone carried in a bag.
2. **Train a classification model based on acquired data.** We recommend Random Forest and SVM classifiers on triaxial and magnitude acceleration timeseries
3. **Build a prototype system**. *A minimal viable product* should consist of (a) an app capturing the accelerometer data and (b) pre-loaded standard classification model. No connectivity to the central service required.
4. **Run test studies with healthy participants** carrying out daily activities and simulating falls while using the app.

After test results are analysed and necessary tweaks incorporated, a fully functional system can be implemented to start a pilot program.

Acknowledgements

I would like to thank my Springboard mentors Ankit Jain and Guy Maskall for their input, discussions and support. Ankit's advice at early stages and his enthusiastic support of the idea were invaluable and helped the project to start off. And Guy's immense knowledge and direction helped to get from an idea to the actual implementation. Ankit, Guy – thank you.

Bibliography

Igual, R., Medrano, C., & Plaza, I. (2015). A comparison of public datasets for acceleration-based fall detection. *Medical Engineering and Physics*(37), 870-878.

Micucci, D., Mobilio, M., & Napoletano, P. (2017). UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *MDPI.*

UniMiB SHAR, http://www.sal.disco.unimib.it/technologies/unimib-shar/. (2017). Retrieved from A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones: http://www.sal.disco.unimib.it/technologies/unimib-shar/