CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

UNOBTRUSIVE SENSOR FUSION FALL DETECTION MAT

DESIGNED FOR ELDERLY INDIVIDUALS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

By

Ramona Khorrami Yaghoubi

December 2021

The thesis of Ramona Khorrami Yaghoubi is approved:

Adam Kaplan, Ph.D.	Date
Jeff Wiegley, Ph.D.	Date
Li Liu, Ph.D., Chair	Date

California State University, Northridge

Dedication

This thesis is dedicated to my entire extended family. I will be forever grateful for their encouragement, support, and unwavering confidence in me throughout my academic journey.

Acknowledgements

My thesis chair, Professor Liu, has extensive knowledge and insights within the field of assistive technology. He allowed me the creative freedom to work on a project that applications truly excite me.

This thesis would have not been possible without the continual support of Adam Kaplan. Kaplan's undergraduate and graduate courses not only reaffirmed my decision to get my masters in computer science, but also made me confident that I would greatly benefit and thrive under his supervision. Professor Wiegley was an integral part of finishing this thesis. His class file, `CSUNthesis.cls` [22] notably helped format this thesis in LaTeX.

The committee's expertise in this field and constant creativity propelled this thesis forward. A big thank you to my entire committee for accepting my proposal and supervising this thesis, as well as to CSUN for allowing me to work along side academics that I regard as encyclopedias of knowledge

Table of Contents

List of Figures

ABSTRACT

UNOBTRUSIVE SENSOR FUSION FALL DETECTION MAT DESIGNED FOR ELDERLY INDIVIDUALS

By

Ramona Khorrami Yaghoubi

Master of Science in Computer Science

Falls are the most significant cause of injury in people aged 65 and over. Injury caused by falls such as traumatic brain injuries and hip fractures could be life-threatening. Falls within the older population can have the most devastating results. Most of these falls occur in the home. The largest subgroup of falls at home happens in the bathroom when the user is the most vulnerable. The severity and frequency of these falls are a result of moisture and slip conducive floors. Assistive technology has greatly helped older adults live safer autonomous lives. Fall detection has been an extensively studied and funded topic. Unfortunately, most fall detectors on the market are invasive and often require the user to carry or attach a device to their body. Furthermore, the accuracy of these wearable fall detectors is dependent on the devices' positioning.

The proposed system, Catch Me If You Bath Mat (CMBM), a non-intrusive bath mat that detects falls, was created to address this issue. The intention of this system is to detect falling, as well as classify walking and sitting. Additionally, the system can distinguish between a human fall and an object fall. A non-slip bath mat in itself is an effective preventative method, absorbing water that causes slippery falls when exiting a shower or bath. The bath mat developed implements sensor fusion consisting of an accelerometer, gyro-

scope, ceramic piezo sensor, vibration sensor, and an unweighted and weighted piezo film sensor. Multiple machine learning methods were used to find which time series classification algorithm would lead to the most accurate result. CMBM sensors are mounted beneath a standard bath mat, 17 by 24 inches, and detect falling from at least 2 feet from the microcontroller.

Using sensor fusion and an Elegoo Nano microcontroller, we collected data from our four activity classes: human falls, object falls, walking, and sitting. After preprocessing the data, we generated and compared multiple models using R's caret package. Human falls can be detected with an accuracy of 96.15%, a sensitivity of 98.3%, and a specificity of 99.28% using the proposed system. Furthermore, it is capable of accurately predicting all classes with an accuracy of 97.54%.

Chapter 1

Introduction

1.1 Problem Domain

The evolution of modern medicine has increased longevity, and with that significantly increased age related health issues and injuries. As we age, our bones become brittle and our muscle function declines, making us more susceptible to life-altering injuries. The most consequential is falling, the second chief cause of unintentional injury deaths worldwide [21]. This is caused by cognitive and physical changes associated with aging. This vulnerable subset of the population is not only more susceptible to injuries and falls, but are less likely to be able to get up following a fall.

If an older person is not able to get off the floor following a fall the complications can lead to pressure sores, carpet burns, dehydration, hypothermia, pneumonia, and perhaps death [10]. According to the CDC, Center of Disease Control [4], people over 65 make up the majority of falls. There are severe and costly consequences associated with these falls, including head injuries, hip fractures, and broken bones. As a result of these falls, over 3 million older adults are treated in emergency departments every year [11].

The number of fall deaths among older adults in the United States has increased by 30% from 2007 to 2016, forecasting 7 falls each hour by 2030 [4]. Most falls occur at home, and the majority of home falls occur in the bathroom, when a person is getting in and out of the tub or shower. Catch Me If You Bath Mat (CMBM) was created in response to the prevalence, severity and consequences of such accidents.

1.2 Approach to Problem

We explored a variety of sensor configurations and placements. The CMBM system includes an accelerometer, gyroscope, three piezo sensors, and a vibration sensor. Each sensor is connected to a Nano microcontroller, which resides on tile flooring. In order to test the range of our fall detector, we marked a radius of two feet around the microcontroller and sensors.

The sensors were tested for sensitivity by inspecting the impact of an object falling. Figure 1.1 displays the 16 sensor output when a 10 lb dumbbell is dropped from a height of three feet within two feet of the sensor system. A height of three feet was chosen as the average height of a table or human hip. Sensors that exhibit a discernible change when the dumbbell hits the floor are considered useful for detecting nearby floor vibrations. All sensors, including the accelerometer and gyroscope, showed a noticeable change when the object fell. According to Figure 1.1, when an object falls it produces a nearly vertical slope, or a dramatic peak or valley.

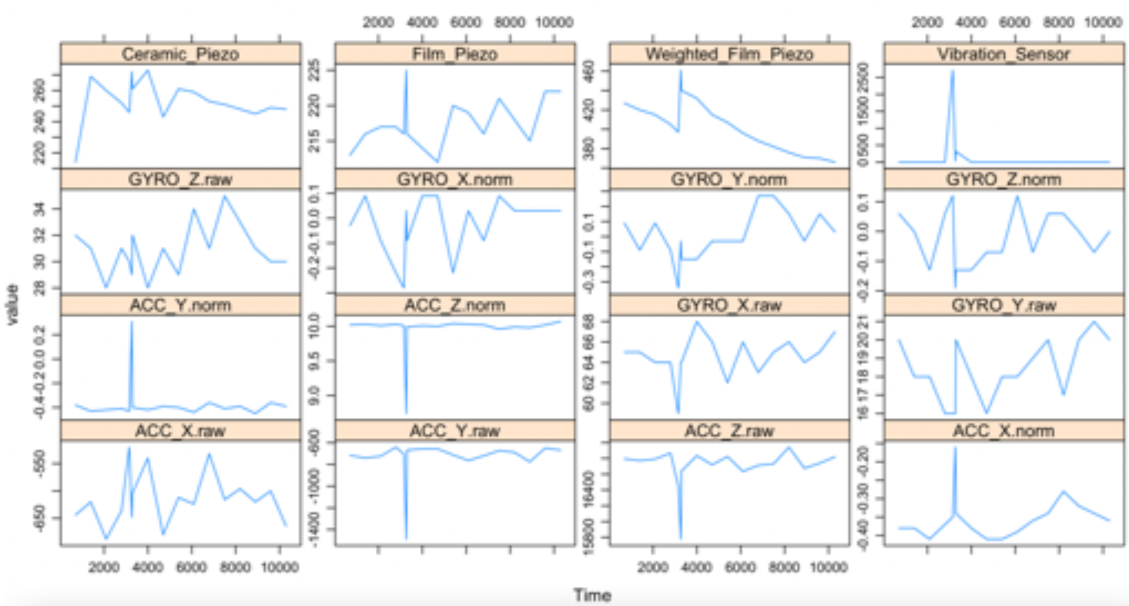


Figure 1.1: Sensor Output for Object Falling

1.3 Pre-Statement of Results

There are 16 sensor outputs that are used as part of the data collection system. Based on 2 layers, using visual judgment, and examining the models with selected covariates, the resulting system is composed of 4 sensor inputs and 11 outputs. Comparatively to the system we start out with, which has 5 sensor inputs and 16 sensor outputs. The functionality of the sensor system was tested on 75% of the training data, and the remaining 25% was used to check the accuracy of our predictive model. Based on data from the selected outputs and surveying multiple classification and prediction algorithms the resulting system provides an accuracy of 96.15% at correctly predicting falls.

Chapter 2

Related Work

2.1 Fall Detection Frameworks

Having access to assistive technology has made it possible for many people to live independently. It is now possible to monitor the well being and safety of individuals living alone, creating a more secure living environment. Among many individuals, there is a generalized fear of falling and its potential consequences. As a result, fall detection has been a topic of extensive research and a number of devices have been developed to detect falls.

They come in the form of wearable, visual dependent, or ambient based devices. Each approach has varying limitations, levels of practicality, privacy, accessibility, and reliability. Lords and Colvin's fall detection badge created in 1991 is a pioneering example. This wearable device is made up of an accelerometer and an analog-input microcomputer chip embedded in a dosimeter badge [18].

Since, there have been many innovative fall detection devices. Including those that are embedded in shoes, under floor boards, or make use of image processing technology. They can be worn on the wrist, neck, and even the side length of your leg. This field has vastly improved with the rapid development of sensor networks, Internet of Things (IoT), machine learning, and classification algorithms.

2.2 Wearable Fall Detectors

The most common fall detectors are wearable. These devices can come in the form of smart watches, pendants, and are even embedded within insoles. They generally take advantage of accelerometers and the occasional gyroscope. Fall detection systems are readily available on the market and sold by countless companies. Many smart watches like the Apple Watch have built-in fall detection capabilities. Starting with the Series 4, released in 2018, the Apple Watch with the help of its embedded accelerometer and gyroscope was able to detect when a user falls.

Aside from watches, most body-contact smart devices are calibrated and tested at a fixed orientation and location, allowing the user to wrongly position the device. According to Yu's work on existing fall detection methods this leads to a high rate of false positives [23]. Doctors report [23] that most of their patients are not likely to actually wear devices that detect falls. Requiring constant periodic charging and wearing at all times creates an expectation of engagement, making it an impractical solution for elderly individuals.

Further, the technological gap between older and younger generations discourages the use of smart gadgets. There is only a small percentage of elderly users that will correctly position, remember to keep their device charged and are technologically savvy enough to use a wearable fall detector.

Devices such as these are intrusive and can easily fail if the user is not compliant with the rules. Despite their widespread popularity, these devices have a blind spot. Our problem's domain is restricted to users falling in the bathroom. This is usually triggered by going into and out of the shower or bath, or by going to the bathroom in the middle of the night while disoriented. It is unlikely, sometimes impossible for the user to be wearing an electronic fall detector in these cases, thus this class of fall detectors would not be applicable.

2.3 Non-Wearable Fall Detectors

Alternatives to wearable fall detectors include vision and ambient-based devices. These solutions do not require constant user participation, and are a great alternative for users who do not find it practical to wear a monitoring device at all times.

2.3.1 Vision Based Solutions

A common approach is using image processing to detect falls. These solutions can be inexpensive and easy to install. Many reliable image-based falls detection systems have been researched and deployed. They range from heat maps to full resolution cameras. Vision-based fall detection takes advantage of RGB, infrared, and depth cameras. Angal and Jagtap from Rajarshi Shahu College of Engineering have put forward a fall detection device using the Microsoft Kinetic Sensor, a depth imaging sensor [2].

The Kinect SDK queries it's sensors 30 frames per second. It is made up of a 640 x 340 pixel resolution RGB Camera, a 3D depth sensor, a motorized tilt, and multi-array microphone. The foreground images are chosen utilizing a dynamic background subtraction algorithm. Velocity, acceleration, width and height ratio of the user's body are extracted from the system. If a fall is detected a message is sent to the previously chosen emergency contact via GSM, Global System for Mobile.

Despite their advantages, camera-based sensors have a number of disadvantages, including cost, intrusion into privacy, high computing power requirements, and susceptibility to occlusion. For a camera-based system to be accurate, it must be positioned correctly, and have a clear line of sight. It is not easy or realistic to guarantee an unobstructed view and location that optimizes accuracy. Therefore, the system is clearly inadequate.

Image-based detection systems with low pixel resolution are available, but users remain uneasy about the input data involved. In the wake of numerous data breaches, most users do not want their bathrooms to be monitored round the clock by any company. Taking into consideration the application of the problem, sensitive bathroom imaging data would not be a feasible possibility for even the most trusting user. Considering the application of this problem, allowing for sensitive bathroom imaging data is out of the question even for the most trusting users.

2.3.2 Ambient Solutions

Ambient fall detectors are positioned around the user. This solution is unobtrusive and does not need constant user participation. A floor sensor developed by the LARSEN Team monitoring and detecting falls, INRIA-Nancy, uses a combination of four force sensors and a three-axis accelerometer encased under each floor tile [8]. The pressure sensors measure load forces, the amount of pressure, and the duration of pressure. The sensors are queried 50 Hz, and built with wireless communication. Once the user's movement is classified, the classification is communicated via Zigbee. The data surveyed in the study used 104 tiles, 60 cm^2 each, and concluded an accuracy of 90% at detecting falls.

Despite the elegance of this system, it is very expensive, rigid, and requires a lengthy and laborious custom installation. These smart tiles need to be customized to fit a space that is not exactly 60n by 60n, where n is a positive integer. Requiring professional installation makes it an impractical solution for most users. The tiles under furniture are unequipped with sensors, which would not allow the user to move or change furniture in the future. Further, to confirm a fall the system needs to detect load forces from at least 3 linear tiles. For tight spaces, in our case a bathroom, the user might not have the 180 cm^2 space to install these tiles.

Clemente's work in 2019, not only detects falling at an accuracy of 93.75%, but is also able to identify a person with only a few steps [7]. The system can calculate fall down localization, with an error of 0.28 meters, achieved from multiple systems communicating. The proposed idea uses a three channel seismometer, which detects the velocity of ground movements. Multiple sensors are installed throughout various rooms. Each sensor system is programmed to know the exact location of the other sensor systems, requiring costly customization.

The study does not mention the power consumption, and high manufacturing cost necessary to have these devices communicate over long ranges and varying obstructions (i.e walls/floors). The experimental data collected and the success of the system was based on collecting data from four separate sensor systems in a 2.5 meter by 3.5 meter carpeted space. The high cost and large size of seismometer sensors, and the dependence of multiple systems required to make a prediction makes this an impractical system for most people.

An alternative solution detects sound vibrations using condenser microphones [20]. While cheaper, and then the previously mentioned system, this system also relies on four microphones, each being placed in a corner of the room. Using the training data the calculations were then customized by adjusting the parameter of the sensor with the highest average discrimination rate. Despite ambient solutions providing a level of privacy and comfort, none of the aforementioned systems are realistic solutions to the problem domain.

Chapter 3

Background

3.1 Primary Objective

Most falls occur as a result of tripping, slipping or losing consciousness. A study of 340 elderly people in Taipei and Yilan County reported that 28.6% of accidental falls occur in the bathroom. The standard dangers associated with falling are exacerbated in this setting. Aside from the unavoidable dangers of slippery floors, hot showers dilate your peripheral blood vessels, lowering your blood pressure enough to cause syncope, fainting [5]. Thus a fall detection system that can discern a bathroom fall is vital.

Current solutions for fall detection are impractical for most older adults and do not work for our problem's domain. Commercial floor sensors capable of tracking steps have been widely available. However, these systems have been shown to interfere with user movement, and can be very rigid and costly [15]. An early example to consider is The PodoBoard, which uses 1 inch aluminum tiles. From this matrix it determines what coordinates the footsteps are coming from [13]. It detects the electrical circuit created between the tiles and the customized shoes. The required shoes are made with metal contacts at the heel and toe.

Six years after the Podoboard, The Magic Carpet, a carpeted smart floor that captures foot position and pressure, emerged from MIT [6]. The carpet is insulated with piezoelectric material. When squeezed or pressed it generates an electrical charge. Also considered was Pinkston's "A Touch Sensitive Dance Floor" [1], where force sensors (arranged in strips) detect footsteps. Both of these systems use contact, force or weight of the user to detect footsteps and "subsequently to generate sonic events".

Despite the capability to make positive predictions these systems require user contact to detect falls, and in some cases an additional required wearable device. Compared to current fall detectors our approach improves on the preceding work by using sensor fusion, granting privacy to the user, and being cost effective enough to make it practical for users from a range of socioeconomic backgrounds.

As seen in the comparison of smart phone based fall detectors between 2009 and 2013 in Habib's work [14] accelerometers and gyroscopes are solely used, with the exception of the use of magnometer in Built-In Kinematic Sensors of a Smartphone [15], to detect falling. The introduction of new types of sensors, with the combination of fusing multiple sensors in the CMBM inputs creates a robust system.

Our system has the added bonus of being a non-vision, non-audio solution, giving the user a nonnegotiable right to privacy. Additionally, the detection of falls is not dependent on physical contact with the sensors. This notable advancement makes the CMBM portable. The all purpose nature of this system, and the advantage of requiring no extra equipment, nor an expensive and customized professional installation allows the Catch Me If You Bath Mat to reach a wide audience.

A system like this promotes autonomy for older adults. For those that live alone, delayed assistance following a fall can have irreparable effects. Lying on the floor for an extended period of time is associated with a higher risk of both bodily and physiological harm. The system presented in this thesis addresses issues of privacy, and practicality while incorporating a convenient and cost effective solution.

3.2 Approach to Testing

The original system used for testing and data collection started with 5 sensor inputs, which resulted in 16 sensor outputs. The 5 data collection sensors are shown in Figure 3.1. This includes a vibration sensor, weighted ceramic piezo, weighted film piezo, unweighted film piezo, and MPU-6050 that has an accelerometer and gyroscope on board.

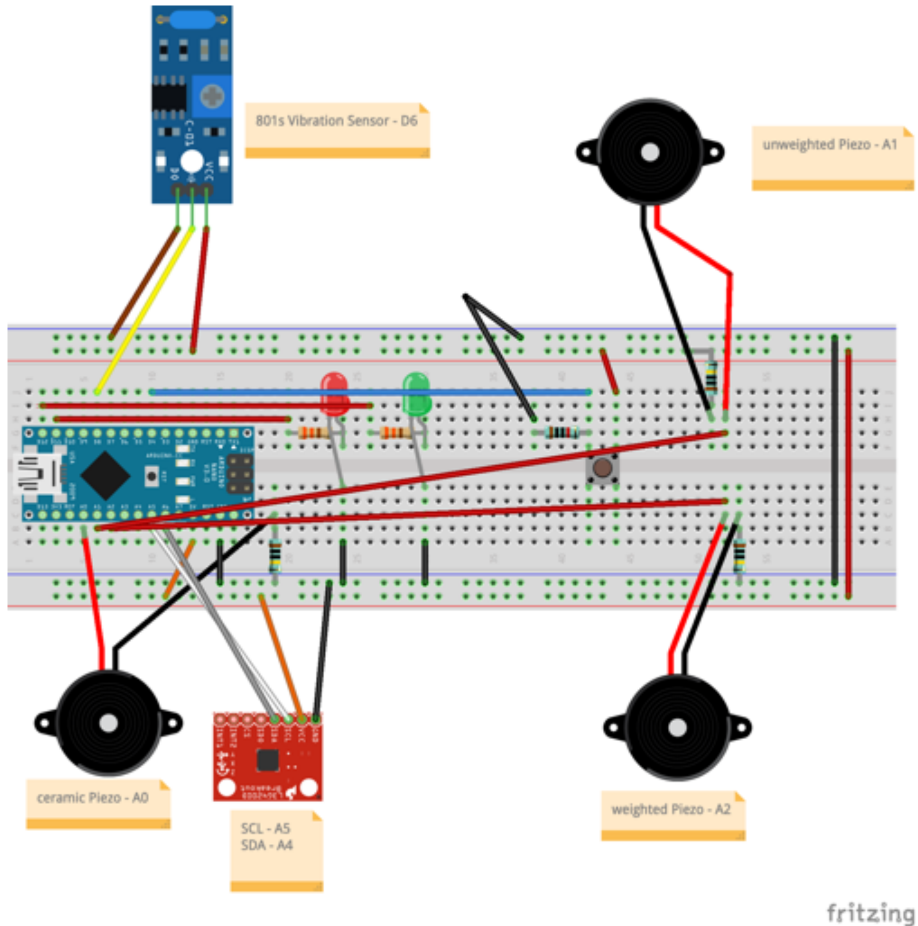


Figure 3.1: Setup: Arduino Nano Testing

The setup and data collection is done on tile flooring to account for the majority of bathrooms since carpet and hardwood are outdated in this setting. The aforementioned sensors are taped to the tile floors. Data is collected a total of 48 times per activity, 12 times in each quadrant, North-East, South-East, South-West, and North-West. The data is collected in the following manner:

The sensors in the system are all queried concurrently. Concurrent testing involves surveying all sensors for 10 seconds while performing the following tasks: human emulated falling, object falling, walking, and sitting. The initial round of testing includes a 75lb dummy intended to emulate a human fall. The second round involved a 10 lb neoprene dumbbell used for classification of an object falling. To collect data for sitting and walking

Algorithm 1 Data Collection Testing

```
Python script reads the serial port
Open new file to write data
while Program does not encounter escape key do
    if Button on breadboard is pushed then
        Red LED turns on
        Delay 5 seconds
        Red LED turns off
        Green LED turns on
        for 10 seconds do
            Collect data from all sensors concurrently
            Parse the data
            Print state to terminal
            Add data to CSV
        end for
        Green LED turns off
        Red LED turns on
    end if
end while
```

a volunteer was used.

When comparing frequency of sampling rate of different threshold-based algorithms Fudickar [13] concluded that a frequency of at least 50 Hz can be a good predictor. This conclusion is reinforced in Abbate's findings [1]. Where he concludes that 50 Hz is a good compromise for minimizing power consumption and precision detecting falls. Measurements and data are taken by the CMBM system at a rate of approximately 2Hz in order to maximize battery life while maintaining a high level of accuracy.

3.3 Criteria for Choosing Sensors

The first step in optimizing the system and reducing the computational load is to reduce the number of sensors, and thus the input vectors. After gathering the data, it is important to clean and visually access it in every stage of testing.The initial determination is done by exploring the sensor data visually using R. Testing revealed that the gyroscope z-axis output, accelerometer z-axis output, as well as the ceramic piezo output did not show adequate sensitivity, which raised concerns regarding their inclusion in the final model.

In a subsequent experiment, adding 0.3 ounces of weight to the ceramic piezo disc sensor demonstrated promising results. Consequently, the design will incorporate a weighted ceramic piezo disc sensor into the CMBM. In the second stage of testing, we looked at sensor correlation to eliminate redundant sensors from the final CMBM system. At the end of the process, after collecting 48 instances of each activity, and choosing a model, variable importance is examined, and backward feature elimination is done to determine which sensors can be removed without losing accuracy.

Chapter 4

Methodology

4.1 Technical Setup

Through the use of a microcontroller, data is collected from five sensors. The fall detection system is implemented using an ELEGOO Nano Board CH340, which is compatible with Arduino Nano V3.0. The CMBM consists of the following hardware components: a vibration module, 3 different piezo sensors, as well as an accelerometer and gyroscope breakout board.

The Vibration Sensor Module 801S, connected to D6, includes an LED indicator and potentiometer, which allows the user to adjust the sensitivity levels accordingly. Using a resistive element rather than a mechanical switching element, this micro shock sensor is sensitive to any type of movement, shock, or vibration. It draws approximately 2mA or 10mA at 5V, which is largely due to the LED mounted on the board.

The system is equipped with two laminated film piezo sensors connected to A1 and A2, as well as a ceramic disk piezo sensor connected to A0. Piezoelectric sensors are transducers that convert mechanical stress into electrical energy. It is recommended to place weights on these sensors to encourage vibrations. Piezoelectric sensors are unique in that they produce an alternating current (AC) voltage when stressed, converting mechanical energy into electrical energy.

The flexible PVDF Piezo Polymer Film can detect and measure flex, vibration, shock, and touch. These films are exceedingly sensitive, mechanically durable, and have a high level of stability. There is no difference between the second laminated piezo sensor and the first except that it is weighed. They both have a wide frequency range from 0.001 Hz to 10^9 Hz. Piezo Pickup 27mm Piezo Amplifier Disc, connected to A0, shows excellent results with

Figure 4.1: Setup: Data Collection and Power Source from Mac

its low power consumption and high sensitivity. As a result of its sensitivity to movement, it can be used to detect vibrations by monitoring its output voltage. Moreover, this brass and ceramic product is temperature resistant, stable, durable, and has a long service life.

The last hardware component is the MPU-6050, connected to A4 and A5, it combines a 3-axis gyroscope and a 3-axis accelerometer, including the hardware accelerator engine for devices connected to the second I2C port, like another accelerator of other brands, magnetometer, or Digital Motion Processor (DMP) of other sensors. For precision tracking of both fast and slow motions, the MPU-6050 features a user-programmable gyro full-scale range of ±250, ±500, ±1000, and ±2000°/sec (dps) and a user-programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g. With three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and another three ones for digitizing the accelerometer outputs. It can be used for gesture recognition, handset and portable gaming, motion-based game controllers, 3D remote controllers, and wearable sensors. Despite the common use of wearable sensors, this module is used to detect floor vibrations.

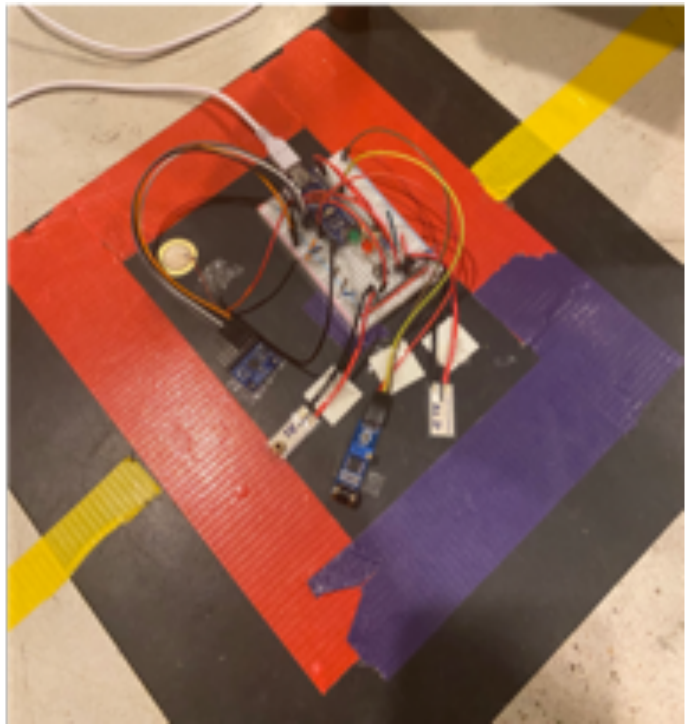


Figure 4.2: Close Up Of System Used For Testing

Originally, the microcontroller was connected to five sensors, resulting in 16 data points. The sensors were found to respond to floor vibrations caused by the 10 pound dumbbell as shown in Figure 1.1. It was determined that the sensors were sufficiently sensitive to pass the initial testing stage. This preparatory step is followed by another filtering step, ensuring that only sensor inputs and outputs that show a clear distinction between activities are included. According to the data, the vibration sensor is unresponsive while testing, and so it is omitted from the final model.

Structural vibration data analysis is only possible with sensors positioned flush on the floor. The arrangement of sensors should minimize any cushioning effect that might degrade response. In particular, piezo sensors perform best when they are firmly pressed against, taped, or glued to their sensing surfaces. In order to accomplish this, I used double-sided mounting tape. The wires are pressed into the foam mounting tape, which is placed right before the sensor. This allows the sensors to lie flush on the floor, while not obstructing its ability to detect floor vibrations.

From the center of the circle, as seen in Figure 4.2, the sensors collect data created by floor vibrations. It detects walking, sitting, and falling, either from an object or from a person weighing at least 75lbs. 48 trial cases were used to generate the data, a combination of 12 attempts for four different activities in each of the four quadrants, Northeast, Northwest, Southwest, and Southeast. Experiments to gather data are conducted on or within the yellow dashed line, Figure 4.1, which represents a 2 feet radius from the center radius.

4.1.1 System Specifications

The final design of the CMBM does not include the vibration sensor. Analyzing and graphing the data revealed that the sensor rarely produced non-zero results. In the human falling experiments, those thought to produce the greatest data changes, the sensor produced a non-zero value only 10 times out of 48. Worse still, only two out of ten of these cases contained more than one non-zero data point. The final components are shown in Figure 4.3. The following tables list the specifications.

Nano Specifications	
Operating voltage	5V
Flash memory	32KB (2KB is used by the bootloader)
SRAM	2KB
Clock speed	16 MHz
Analog IN pins	8
EEPROM	1 KB
DC current per I/O Pins	22 (6 of which are PWM)
PWM output	6
Power consumption	19 mA
PCB Size	18 x 45 mm
Weight	7g

Ceramic Piezo Disc Specifications	
Material	Brass and ceramic
Center Disc Size	Approx 27 mm diameter
Disc Thickness	0.4 mm
Cord Length	33mm
Type	External drive type element
Resonant Frequency	4.6 ± 0.5 KHz
Resonant Impedance	300 ohms max
Additional weight	CR2032, coin cell battery
Coin cell size	20 x 3.2 mm
Coin cell weight	3.0 grams

Weighted and Unweighted Film Piezo Specifications	
Wide frequency range	0.001 Hz to 109 Hz
Range	10^{-8} to 10^{6} psi
Sensitivity	50 mV/g to 800 mV/g
Operating Temperature	0ºC to 85ºC
Storage Temperature	-40ºC to 85 ºC
Connection	Solder Tab
Material	28 μm thick piezoelectric PVDF polymer film with screen-printed Ag-ink electrodes
Optional Mass Version	Additional 0.26g

GY-521 MPU-6050 Module 3 Axis Gyro + Accelerometer Specifications	
PCB size	1.65 x 2.0 cm
Main chip	MPU-6050
Built in	16 bit AD converter
Communication	Standard IIC communication protocol
Protocol	I2C
Technology	Micro Electro Mechanical System
Gyroscope range	±250, ±500, ±1000
Acceleration range	±2 g, ±4 g, ±8 g, and ±16 g
Voltage	3.3 -5 v (with voltage regulator)
Size	2 x 1.6 x 0.1 mm
Number of pins	8
Pin spacing	2.54 mm

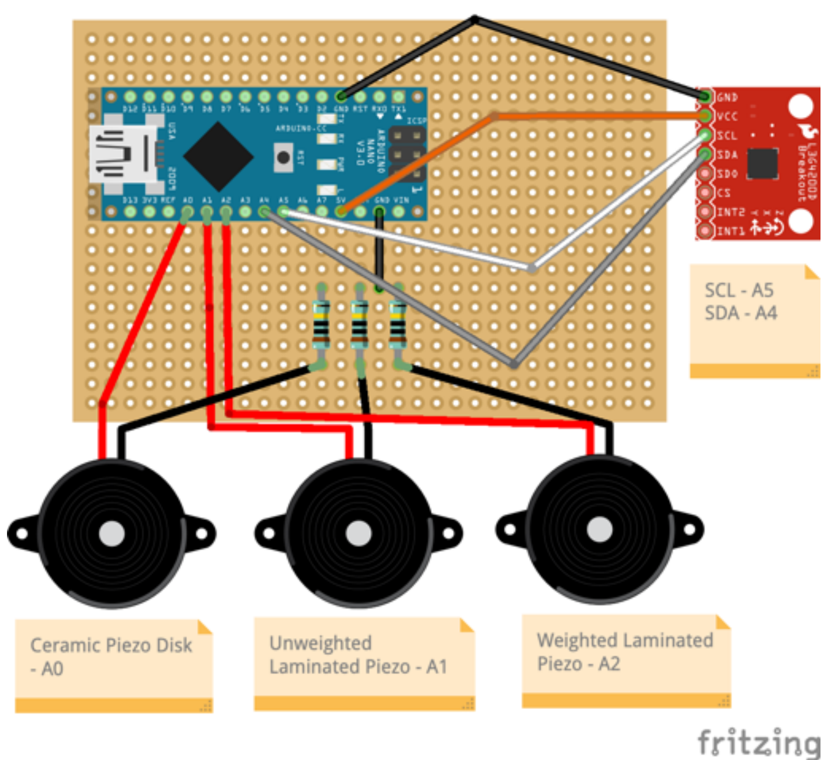


Figure 4.3: Final Configuration

4.2 Software Setup

Multi-stage data collection and processing is required with Human Activity Recognition, or HAR. The data stream is initiated by uploading nanoSensorData.ino, an Arduino program, to the Arduino Nano. We begin by getting offset values and calibrating the accelerometer and gyroscope on the MPU-6050 module. The .ino sketch is set to ±2 g, where the accelerometer and gyroscope are the most sensitive, in comparison with ±4 g, ±8 g, and ±16 g.

In the MPU-6050 module, gyroscope and accelerometer sensor data are represented as 16-bit raw data in 2's complement form. A normalization procedure is then performed, leaving the values of acceleration along the X, Y and Z axes, in gravity force units (g), and the gyroscope parameters along the X, Y and Z axes in dps. The g units of the accelerometer can be varied from 0 to the range chosen. In this calculation, the range setting of the accelerometer is taken into account, as well as the scale setting for the gyroscope, as well as their respective sensitivities.

The sensors are then queried for data. The Python script connects to the serial port and reads incoming data. This data is displayed on the terminal and a copy is written to a CSV file. The model was trained and tested by gathering data segmented by activities. Except for testing and training, data is not normally segmented by time. It is crucial to create segments that overlap when analyzing time series data. In the final product, a sliding window of 10 seconds with a 25% overlap is used. Following the data collection, normalization, and cleaning, the data should be visually examined.

The duration, slopes, and frequency of peaks and troughs of each event are analyzed. In order to extract features, we take into account event duration, standard deviation, maximum peak, five values before and after the maximum peak, as well as the number of peaks [7]. These features make it possible to differentiate common events with similar characteristics. As shown in Figure 1, both an object falling and a person falling can cause a significant trough or peak in the time domain. These peaks and troughs, however, will differ significantly in amplitude and duration. Walking data, on the other hand, produces a greater frequency of peaks at shorter intervals when compared to data from human falls.

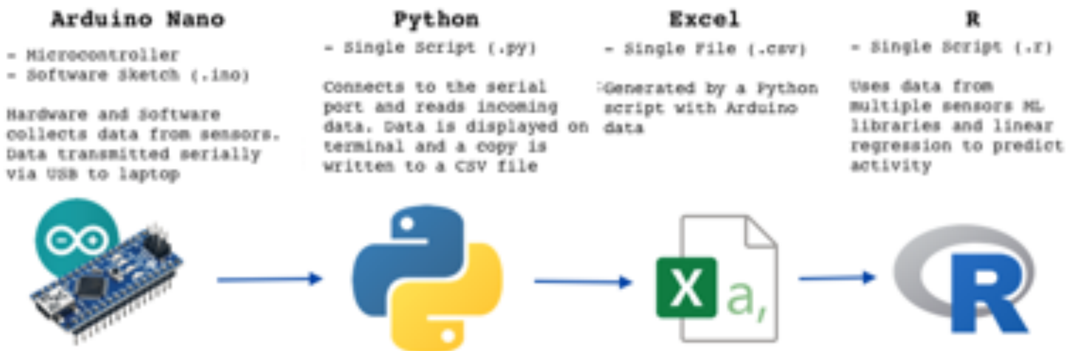


Figure 4.4: Data Flow

The R script, dataVV_Automated.R, collects activity data from each CSV and then cleans, graphs, and aggregates the data into one data frame. The data is then divided up for training and testing in createTestTrainData.R. Of the data, 75% is used to train the machine learning model, and 25% is saved for testing its accuracy. We conduct 12 experiments in each of our four quadrants of the circle per activity. For instance, 12 dummy falls are conducted in the northeast quadrant, then 12 dummy falls are conducted in the southeast quadrant, etc. When splitting data, for every 3 experiments added to the training data, 1 experiment is added to the testing data. For testing data, this would result in three instances per quadrant for each activity. Finally, classificationAlgo.R runs multiple machine learning algorithms in order to determine which has the greatest accuracy, sensitivity, and specificity. Figure /ref[fig:DataFlow] outlined the tools used for hardware programming, gathering data, preprocessing data, and running classification algorithms.

Listed below is the pseducode that details the HAR software steps used to create the CMBM.

Algorithm 2 HAR Process: Data Preprocessing and Classification

```
for Each activity do
    for Each quadrant do
        Gather experimental data 12 times
        Write data to file
    end for
end for
Load the data into memory
for Every cell that is blank, NA, NaN, and Inf do
    Set cell to 0
end for
Basic data exploration and visualization
Remove variables that have 30% or more non-integer or zero values
Remove timestamps
Split the data into training and testing data
Analyze principal components
Tune the cross-validation parameters
for Bagging with Trees,
  Logistic Regression with Boosting,
  SVM with Boosting,
  GBM,
  GBM with Alternate Tuning Grids,
  Random Forest do
      Train model
      Evaluate model based on prediction accuracy, sensitivity, and specificity
end for
```

4.3 Emulating Falls

The DEFY Leather Jiu Jitsu MMA Grappling Dummy Judo Martial Arts Punching Bag Karate Black is used to simulate falling. With regard to its weight capacity, it can support 70 kg, or around 154 pounds. The dummy is filled with sand, webbing and pool noodles. When empty, it measures 9.84 x 7.24 x 6.77 inches and weighs 3.75 lbs.

The noodles were sized down and positioned inside the dummy so that normal joint movement could be achieved. The noodles measured 58” long and 2.25” wide before they were cut. The final weight of the dummy is 75 lbs. This includes 71 lbs of sand, 3.75 lbs for the leather suit, 0.7 lbs for the pool noodles, and 0.3 lbs for the webbing. Dummy bulking is achieved by using webbing, which is lightweight and does not add excessive weight or toughness. It was added because the dummy’s density and hardness would not resemble normal human composition without it.

Figure 4.5: View 1: Dummy Used For Emulated Falls

Figure 4.6: View 2: Dummy Used For Emulated Falls

To ensure safety and to comply with IRB guidelines, a dummy was used to simulate falls in our makeshift laboratory. The average weight and height of a woman in North America is 170.8 pounds at 63.5 inches, while a man is 199.8 pounds at 69 inches [16]. Measurements for the dummy selected are slightly less than those for most adults, 91lb at 58 inches [16]. Figures 4.5 and 4.6 show the dummy lying on the ground following an simulated fall. Dummy specifications are as follows:

Dummy Specifications	
Material	Leather
Filling	Sand + Pool Noodles + Polyester Webbing
Weight	3.75 lbs (empty)
Dimensions	9.84 x 7.24 x 6.77 inches
Final Weight	75 lbs (with filling)

The CMBM system focuses on falls occurring in bathrooms, so we are interested in falls that start from a standing or walking position. According to Yu, different types of falls have their own characteristics. [23]. Our test data will be more accurate if we study the specific characteristics of our use case.

Falls where a person is initially lying or sitting are distinguished from those where they are initially standing or walking by the duration and the initial height of the fall. Falls are faster, lasting 1-2 seconds, and the starting height is at least twice as high, resulting in more impact. Yu describes upright falls as having the following characteristics: the head is in free fall, it drops to the ground, resulting in the head lying motionless or with slight movement on the floor for a long period of time [23]. Additionally, the user's head is within n inches of where their feet are, where n represents their height in inches. If a user falls on the floor, the system assumes they will not collide with nearby objects such as a wall or a toilet.

4.4 Classifier Models

Using R, we preprocessed, analyzed, and came up with an optimal Machine Learning algorithm. By utilizing packages and tools from this statistically focused language, we were able to pre-process the data, identify critical variables, as well as choose and tune various Machine Learning models. In this study, we used the caret package, Classification and Regression Training, to obtain the most effective classifier model for creating predictive models. To select a predictive model, pre-processed data is divided into training and testing sets. In order to accurately classify each event, 75% of data sets from human emulated falls, object falling, walking, and sitting will be used for model training and the remainder for evaluating model performance. As a result of the maximum dissimilarity sampling method, data is divided into training and test sets based on their predictor values.

It is a common challenge to create complex models when there are a large number of predictor variables. We aim to prevent this, but at the same time avoid oversimplifying.

We removed variables that have more than 50% NAs, which in our case was the vibration sensor. Timestamps and new window declarations were also ignored. We selected features based on the accuracy of multiple models that used different sets of predictor variables. Before we apply machine learning algorithms to train our model, we first need to tune the cross-validation parameters. We carried out a 5-fold Cross-Validation to prevent overfitting, resulting in a small out-of-sample error.

We compared the following supervised learning models for classifying events: Bagging with Trees, Logistic Regression with Boosting, Support Vector Machine with Boosting, Gradient Boosting Machine, and Random Forest. In the context of sensor data, these models have shown to perform well for time series classification and activity recognition. As noted in Figure 4.7, we considered 2 Decision Tree models, 2 Boosting models, and one model that incorporated both, Gradient Boosting. Caret's train() function uses these models along with our training data to create robust prediction models. In this function, the model is cross validated, the hyperparameters are tuned for optimal performance, the predictors are preprocessed, and the best model is selected based on a given evaluation metric.

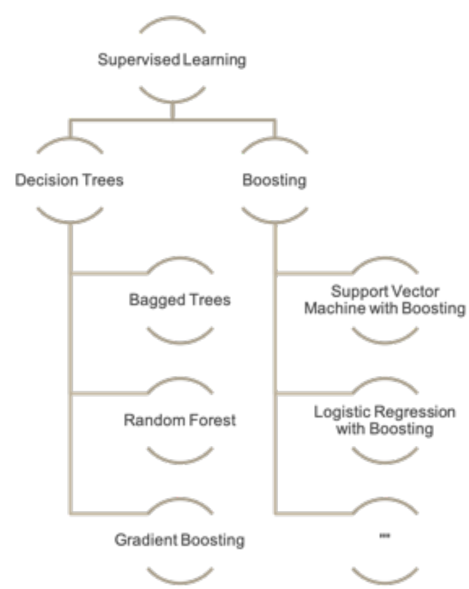


Figure 4.7: Models Considered

4.4.1 Boosted Support Vector Machines

Support vector machines (SVMs) are powerful, highly flexible techniques for general classification, regression, and outlier detection [19]. Whenever SVM classifies two classes, it maps observations in space as points, so that the gaps between the categories are as wide as possible. It performs complex data transformations based on the selected kernel function to determine the optimal separating boundary, also known as a hyperplane, between different classes. The hyperplane between the two classes aims to maximize the distance between the classes' closest points. We call the points located on the boundaries support vectors, and the middle of the margin is our optimal separating hyperplane. To make predictions, new observations are mapped into this space.

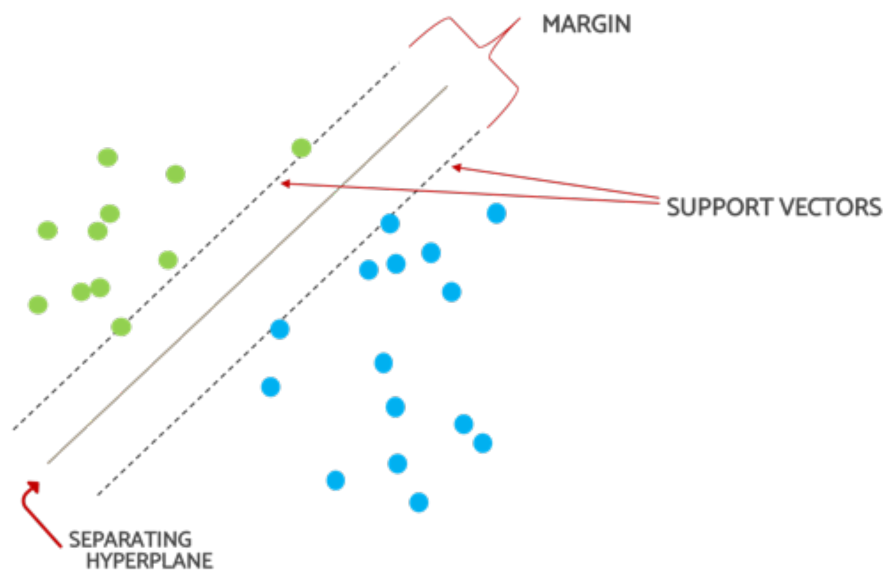


Figure 4.8: Binary Classification

SVMs were developed for binary classifiers, but can be extended to multiclass classification problems, by breaking down the multiple binary classification cases. This approach is also called one-against-one. Using a voting mechanism, it fits all binary subclassifiers and finds the correct class. We train a model using svmRadialCost, which tunes over cost (sigma) and uses a single value of sigma based on kernlab's sigest function, in this case sigest is run inside of each resampling loop. The tuning parameter for this method is C, cost also know as sigma.

4.4.2 Boosted Logistic Regression

Logistic regression, a supervised learning algorithm, is one of the most used Machine Learning techniques. For training a logistic regression, maximum likely hood estimation is used. For maximum likelihood there are two basic optimization methods: Gradient Descent and Newton's Method. In LogitBoost, the latter is used, which is the first and second derivatives of the loss function. You can get the maximum likelihood by L_i, minimizing the negative log likelihood loss.

$$L_i = -log(p_{i,k})(y_i = k)$$

The main advantage of this method is the clarity of the results and the simplicity of its explanation of the relationship between dependent and independent features. In comparison with random forest and gradient boosting, it requires considerably less processing power, and is general faster. We train our model with Logistic Regression with Boosting, using the method LogitBoost. With LogitBoost, the generalized additive model is used with a cost function from logistic regression [12]. In order to improve prediction and reduce variance, boosting is required. The tuning parameter for this method is $nIter$, the number of boosting iterations.

4.4.3 Bagging with Trees

Bagging, or bootstrap aggregation, is a technique for reducing the variance of statistical learning methods. Bagging with trees is a type of decision tree for supervised machine learning. The disadvantage of decision trees is that they are high-variance estimators. A few additional training observations can have a dramatic effect on the predictive performance of a learned tree.

The algorithm constructs N classification trees from bootstrapped training sets, then combines the predictions from all models. Each model is then used to generate a prediction for a new sample and these N predictions are averaged to give the bagged model's prediction [19]. Individually, each tree has high variance, but low bias. This process of averaging N trees, however, reduces variance. For classification problems, the predicted value for an observation is the mode of the trees.

Bagging models provides several advantages over models that are not bagged. The aggregation process of bagging effectively reduces the variance of a prediction. In models that produce instabile predictions, such as regression trees, aggregating over multiple versions of the training data actually reduces the variance in the prediction and, therefore, makes it more stable. We train the data using treebag, from the caret package. There are no hyper-parameters to tune the model.

4.4.4 Random Forest

Random forest is a supervised learning algorithm. It is an extension of bagging. An ensemble of decision trees are built, using the bagging method. A random selection of features are used in each sample. The random forest is a collection of decision trees, but there are some differences [19].

In a decision tree, training data with features and labels will formulate some set of rules, which will be used to make predictions. Random forest algorithms randomly select features and observations for building multiple decision trees, and then averaging the results. In addition, deep decision trees may suffer from overfitting. Random forests prevent this by creating random subsets of features and building smaller trees based on those subsets. It then combines the subtrees. Both bagging and random forests have been found to be effective on a wide range of different predictive modeling problems. We train out model using caret's rf .

The tuning parameter for this method is $mtry$, randomly selected predictors.

4.4.5 Gradient Boosting Machine

Gradient boosted machines (GBMs) are a popular machine learning algorithm that have proven successful across many domains. In contrast to random forests, GBMs build an ensemble of shallow and weak trees that build on each other, each improving on the previous one. By combining these weak successive trees, you can create a model that is highly accurate at predicting the response variable. GBM creates a lot of weak learners that have little predictive power, but stacking them together gets a powerful model as we will see in [17]. This algorithm works as follows:

Algorithm 3 Gradient Boosting

```
1: Fit a model to the data: F_1(x) = y
2: while true do
3:     Fit a new model to the residuals: h_1(x) = y − F_1(x)
4:     Create a new mode: F_2(x) = F_1(x) + h_1(x)
5: end while
```

Multiple iterations of the algorithm result in a continually improving model [3]. These models are very accurate. The function fit can be optimized on different loss functions, and has several hyperparameter options that make it very flexible. Since no imputation is required, raw categorical and numerical values can be used without preprocessing. Because GBMs constantly improve to minimize all errors, this can lead to overemphasizing outliers and overfitting. To counteract this, you must use cross-validation.

We train two multinomial gradient boosting machines (GBM). The first model is using GBM default settings and 5-fold CV, the second one uses an alternate tuning grid for parameters. The method uses the same approach as a single tree, but sums the importances over each iteration of boosting. The Boosted Trees Model uses the gbm function in the gbm package. For a gradient boosting machine (GBM) model, there are three main tuning parameters, that can be set with tunGrid. The hyper parameters to tune the model are

$interaction.depth$, the complexity of the trees, $n.trees$ tuning the number of trees (boosting iterations), and $shrinkage$, the learning rate.

Chapter 5

Experimental Results

5.1 System Performance to Machine Learning Models

HAR, or Human Activity Recognition, has become a key research area in the past few years and is gaining attention for its applications. In our research, we consider two types of problems: a binary classification problem for our CMBM system, as well as a multiclass classification problem. The first measures floor vibrations to determine if a fall has occurred or not. A second classification problem has to do with determining if a person fell, an object fell, or if they walked or sat. All sensors were used except for the vibration sensor, which showed poor results for both problems. For each activity, 25% of the 48 data points, 12 in each quadrant, were withheld in order to test the systems' performance. These 12 segmented windows per activity were withheld for evaluating the accuracy of our algorithm and sensors. There are 3 segmented windows per quadrant per activity. Based on its accuracy at predicting classes correctly on the remaining 25% of test data, we choose the best model.

Test accuracy is the ability to distinguish and correctly predict between various activities [9]. The goal of the system is to increase the number of true positives and true negatives and to reduce the number of false positives and false negatives. True positives (TP) are classes which are correctly predicted as their actual class. False positives (FP) occur when the model predicts the activity that occurred incorrectly. True negatives (TN) occur when the model correctly predicts that an activity didn't occur. False negatives (FN) occur when the model predicts an activity that did not occur. As seen in Figure 5.1, a confusion matrix represents how to extrapolate those values, where the x-axis represents predictions, while the y-axis represents the class label.

Figure 5.1: Confusion Matrix

This system is intended to predict when a user will fall or not. We introduce the other activities, sitting, walking, and falling an object, only so we can analyze which activities the system incorrectly predicts falling (false positives) and when it fails to recognize a human fall (false negatives). To measure model performance, we will use the accuracy of predicting human falls using the following formula.

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$$

The results from the selected models are presented in two levels of accuracy, fall detection accuracy, which is a binary classifier, falling or not, and activity classification, which is a four level classification problem. Both are presented in Figure 5.2.

	Fall Detection Accuracy	Overall Accuracy	Object Falling Sensitivity	Sitting Sensitivity	Walking Sensitivity
Bagging with trees	89.64%	84.45%	82.8%	77.47%	81.11%
Boosted Logistic Regression	93.95%	95%	94.12%	97.62%	90%
Boosted Support Vector Machines	96.15%	77.08%	80.83%	68.68%	60.56%
Gradient Boosting Machine	93.58%	97.68%	93.78%	98.90%	100%
Gradient Boosting Machine Tuned	93.54%	97.82%	94.30%	99.45%	100%
Random Forest	93.05%	91.18%	86.53%	88.46%	92.22%

Figure 5.2: Accuracy and Sensitivity of Model

There are several models that produce at least 90% accuracy for the multiclass classification problem. The best is the Gradient Boosting Machine, which has 97.68% accuracy. On walking and sitting, it shows incredible sensitivity, the measure of a model's ability to predict true positives for each of the available categories. The model uses 5 fold cross validation, the learning rate is set to 0.1, the number of boosting iterations or trees is 150, and the depth of those trees is 3.

In the binary classification problem, boosted support vectors are superior. This model uses five fold cross validation at a cost of 1. The model has a very high accuracy rate of 96.15%, and is computationally conservative. In spite of its high accuracy in predicting falling on our testing data, it is less effective at predicting the other 3 activity classes. When it comes to classifying all 4 activities using SVM, we get an accuracy rate of 77.08%.

5.2 Sensor Fusion Results

Human-computer interaction using sensor fusion has been considered as a powerful technique to address the complexity of detecting falls. Research shows that synthesizing signals from multiple sensors could produce superior accuracy and reduce false alarms. A fusion based approach will result in a more robust classification. The original system included 16 data points to analyze the impact of activities on the sensor. While beneficial this descriptive power with a large amount of data results in a high computation speed and power consumption. As previously mentioned, the vibration sensor was removed from the final model leaving us with 15 sensor outputs.

From the training data, as seen in Figure 5.3 a scatter plot matrix (SPLOM) is made. The lower diagonal draws scatter plots, the diagonal histograms, and the upper diagonal displays the correlation between sensors. We aim to remove any sensors that are unnecessarily redundant by checking to see if any of the sensor outputs are highly correlated. It showed no severe correlation between variables. However, there is a relatively high correlation between the unweighted film piezo and the weighted film piezo (0.73), as well as the ceramic disc piezo and the unweighted film piezo (0.73).

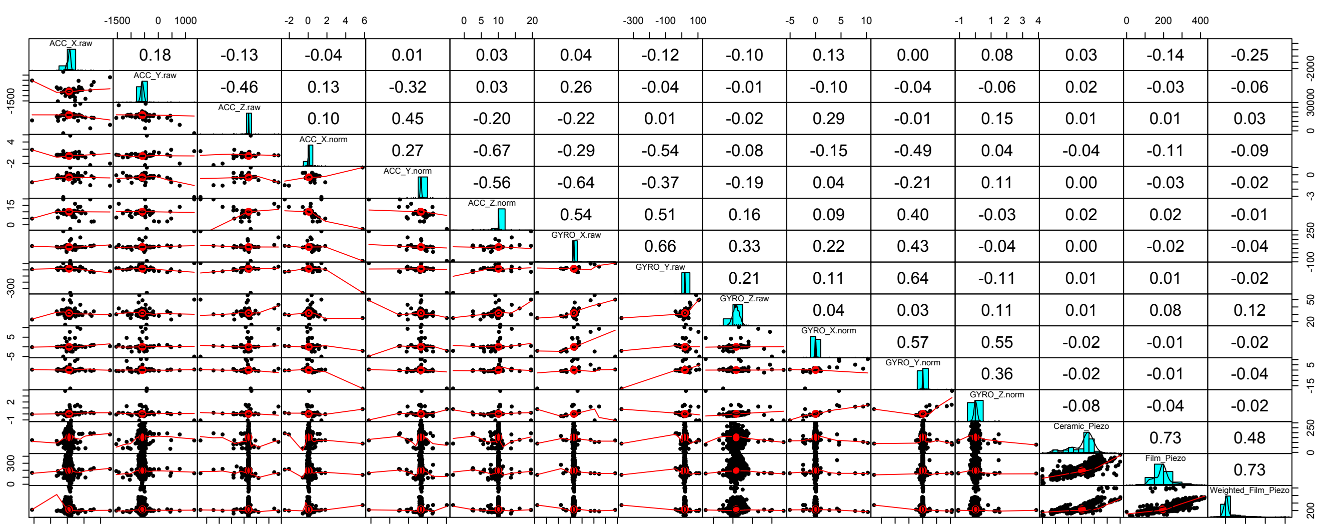


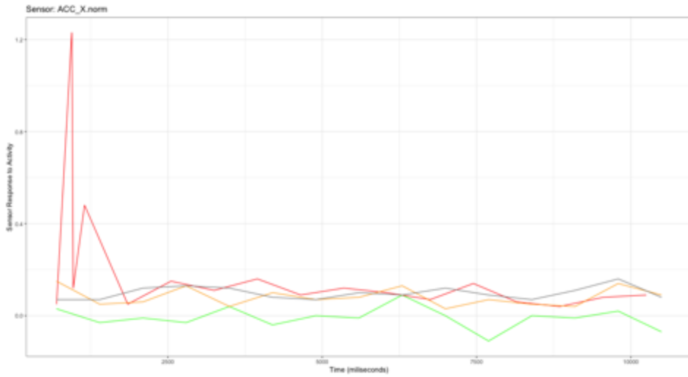
Figure 5.3: Sensor Correlation

Using this information, the performance of our models assessed was assessed without the use of the data from the unweighted piezo sensor. This is because of the 73% correlation to both the weighted piezo, and the ceramic disc piezo. Figure [**?**] show for our binary classification problem, fall accuracy declines for SVMs and Random Forest, even slightly so for Bagged Trees and GBMs. It only improves for Logistic Regression and the tuned version of the GBM. For our multiclass classifiction problem for SVMs and both GBMs. It gets better for Bagging with Trees, Logistic Regression, and Random Forest. We do not see improvements in our accuracy in either of these classification problems, which makes removing the sensor from the final system unjustifiable.

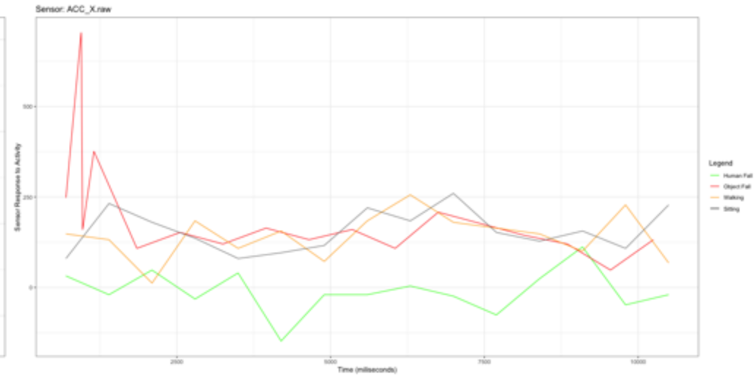
	Fall Detection Accuracy	Overall Accuracy	Object Falling Sensitivity	Sitting Sensitivity	Walking Sensitivity
Bagging with trees	89.17%	86.36%	81.87%	81.32%	85.56%
Boosted Logistic Regression	94.50%	96.63%	94.59%	97.60%	96.77%
Boosted Support Vector Machines	92.59%	75.17%	78.24%	71.43%	52.78%
Gradient Boosting Machine	92.59%	97.27%	91.71%	99.45%	100%
Gradient Boosting Machine Tuned	94.08%	97.41%	93.26%	98.90%	99.44%
Random Forest	88.71%	91.54 %	86.53%	87.91%	95.00%

Figure 5.4: Accuracy without Unweighted Film Piezo

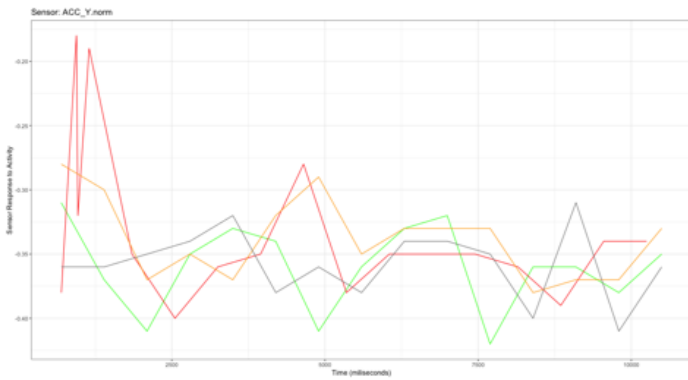
Figure 5.5 shows the impact and sensor behavior of the different activities. Each subgraph has data from one observation for 4 activities stacked on top of each other. The green lines represent human falls, the red lines represent an object falling, the orange lines are walking, and the black line is sitting. It is relevant to note that despite all having the same time window of 10 seconds, activities such as human falls, object falls, and sitting occur at different times. This is because each activity experiment is discrete. It is difficult to visually decipher the behavior of each activity on each sensor, but there are evident differences in the frequency and amplitudes of peaks and troughs. All 3 piezo sensors however, have very obvious differences between falling and all other activities.



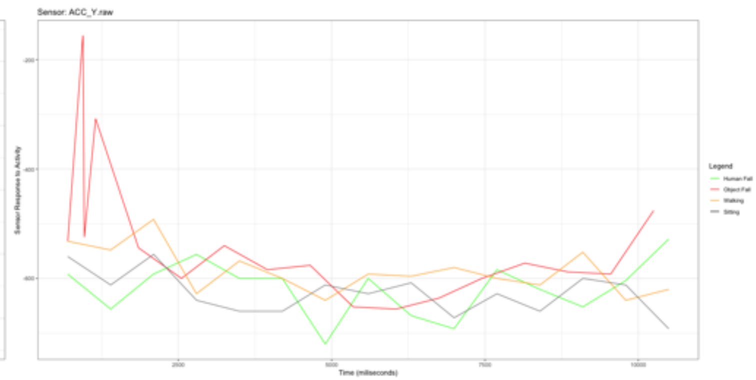
(a) Accelerometer x-axis norm



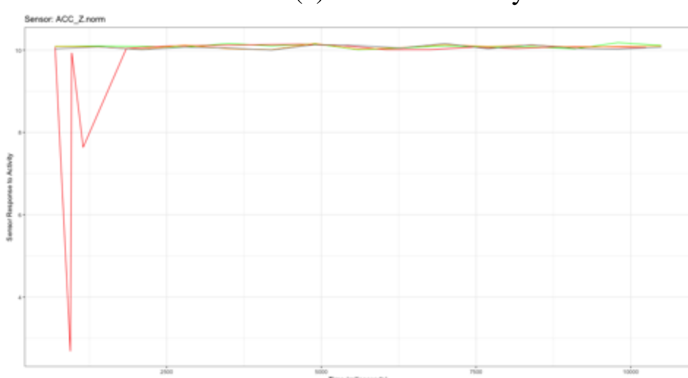
(b) Accelerometer x-axis raw



(c) Accelerometer y-axis norm

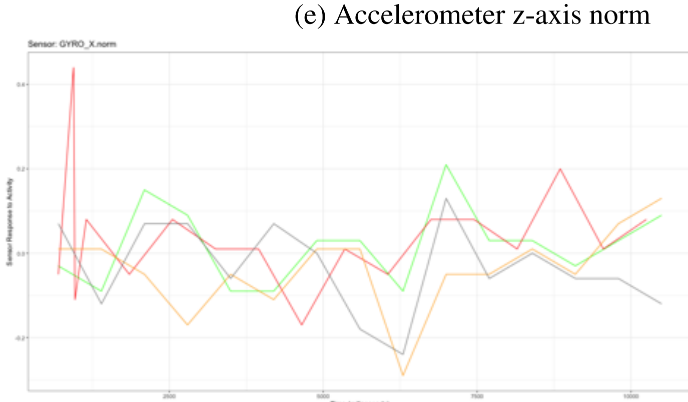


(d) Accelerometer y-axis raw

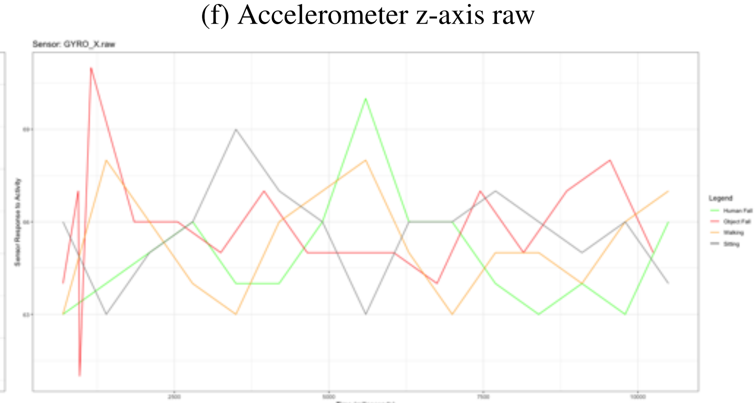


(e) Accelerometer z-axis norm

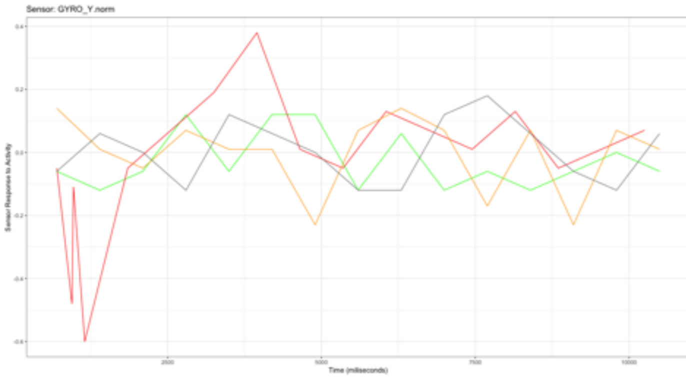
(f) Accelerometer z-axis raw



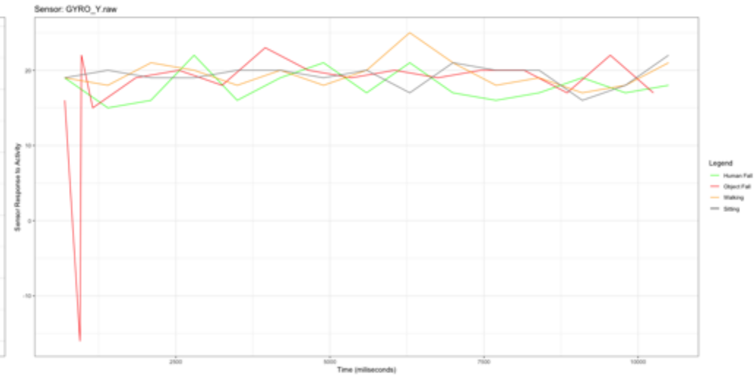
(g) Gyroscope x-axis norm



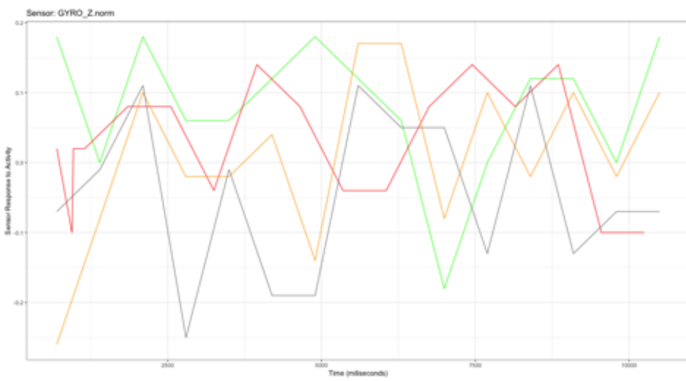
(h) Gyroscope x-axis raw



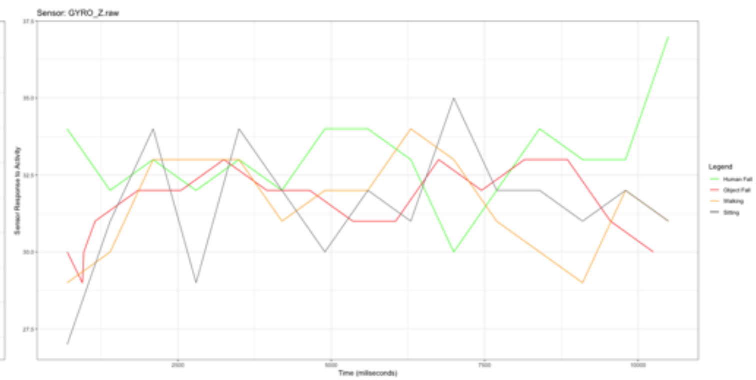
(i) Gyroscope y-axis norm



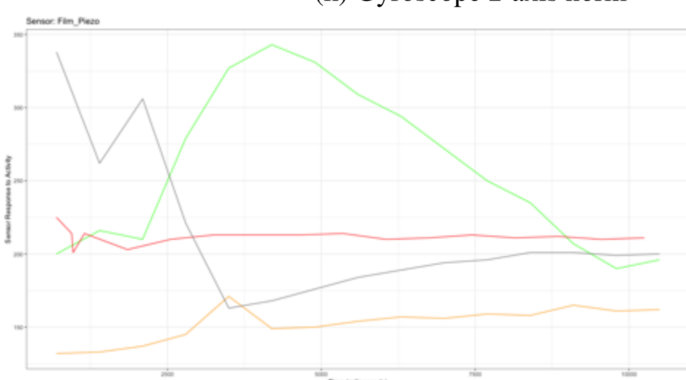
(j) Gyroscope y-axis raw



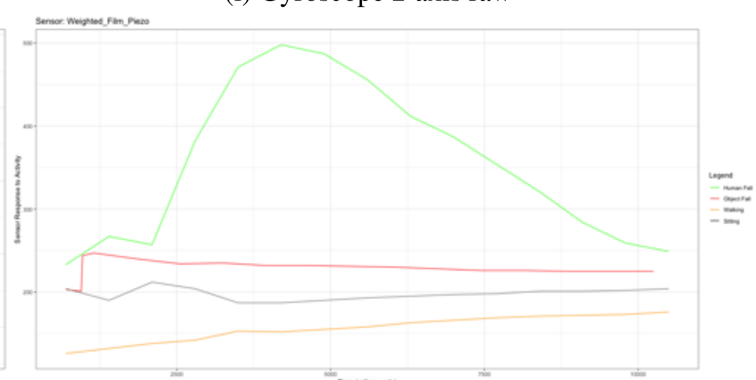
(k) Gyroscope z-axis norm



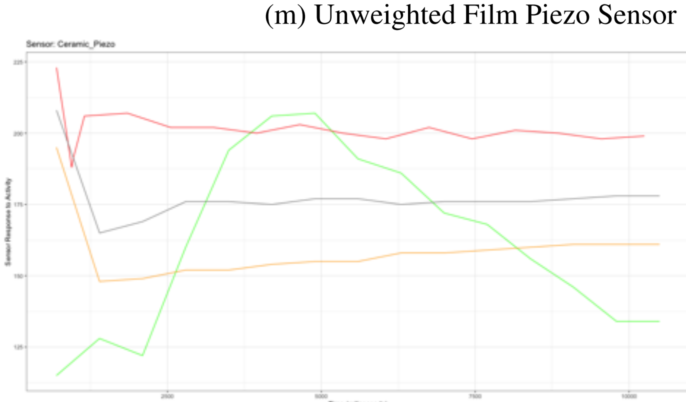
(l) Gyroscope z-axis raw



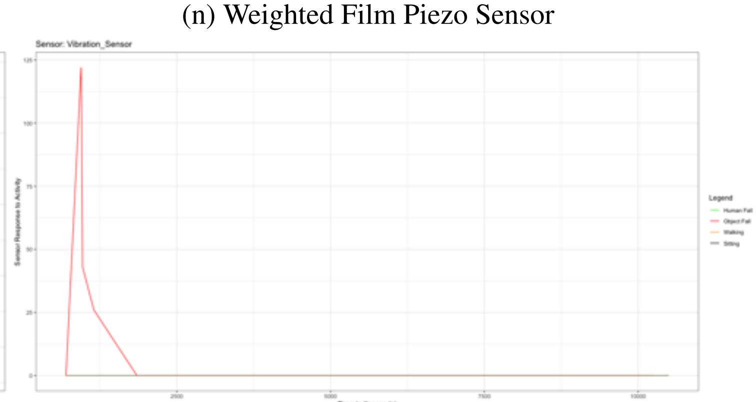
(m) Unweighted Film Piezo Sensor



(n) Weighted Film Piezo Sensor



(o) Ceramic Piezo Sensor



(p) Vibration Sensor

Figure 5.5: Graph of 4 Activities Superimposed, Per Each Sensor

We further investigate the our senors for our selected classification models. To be able to choose the final sensor model it is critical to use the optimal number and configuration of sensors. Figure 5.6 and 5.8 show sensor importance to the SVM model and GBM model, respectively. In order from least to greatest, gyroscope y-axis raw, x-axis norm, y-axis norm, and x-axis norm are shown to have less than 20% importance in the SVM model. As indicated in Figure 11, the accuracy obtained from Boosted Support Vector Machine was slightly larger, however not significant enough. This led to the decision to perform backward feature elimination on the gyroscope x and y axis sensors from order of least importance. In Figure 5.7 we can see that removing all 4 sensors mentioned does not effect the accuracy of the Fall Detection system using SVM.

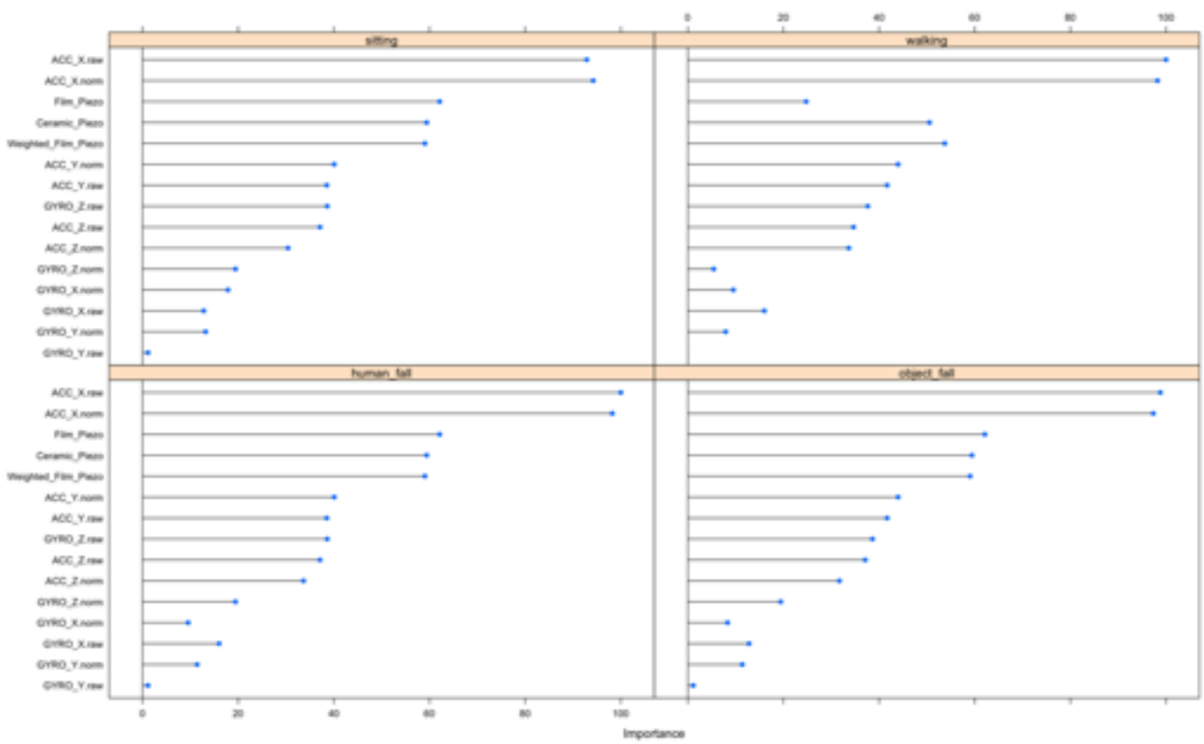


Figure 5.6: Boosted Support Vector Machine Variable Importance

	ORIGINAL	A	B	C	D
	ALL Sensors	Without Gyro Y-raw	Without A & Gyro X-norm	Without A,B Gyro Y-norm	Without A,B,C Gyro X-raw
Boosted Support Vector Machines	96.15%	95.62%	96.15%	96.15%	96.15%

Figure 5.7: Fall Detection SVM Accuracy: Removing Sensors

Using the same technique as above we view the variable importance of the sensor outputs to our multiclass classification problem using GBM. GBM shows less than or equal to 20% variable importance to the following sensors: gyroscope x-axis raw, accelerometer y-axis norm, gyroscope z-axis raw, accelerometer z-axis norm, y-axis raw, z-axis raw, and gyroscope y-axis raw, y-axis norm, x-axis norm, unweighted film piezo, and weighted film piezo, in order from least to greatest importance. As seen in Figure 5.9 we performed backward feature elimination and found that the accuracy peaked when removing the gyroscope x-axis raw. However, when removing 9 more sensors we only lost 0.41% accuracy.

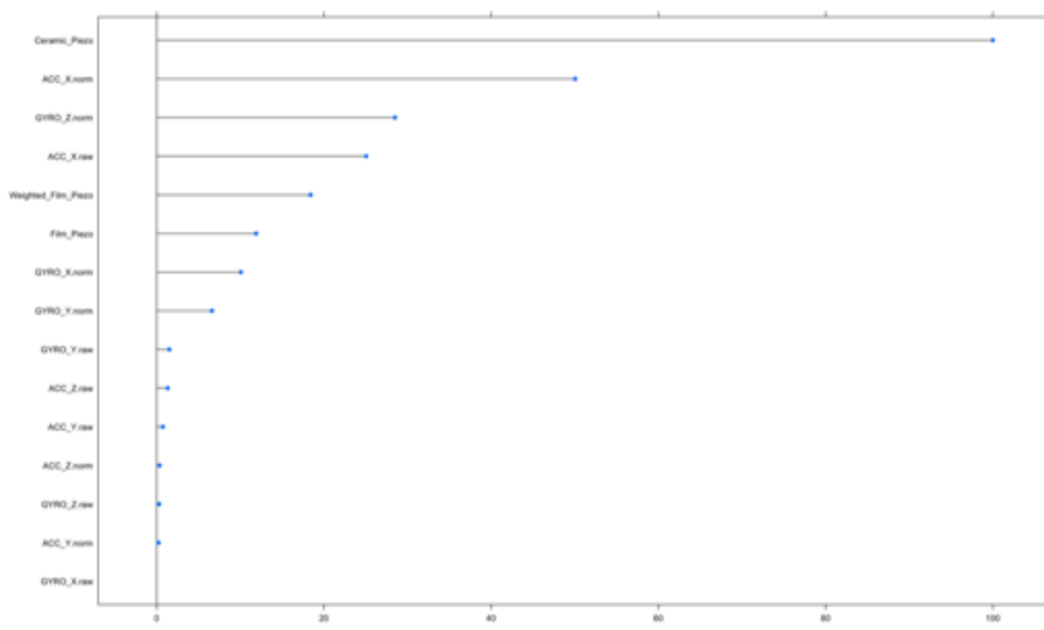


Figure 5.8: Gradient Boosting Machine Variable Importance

	ORIGINAL	A	B	C	D	E
	ALL Sensors	Without Gyro X-raw	Without A & Acc Y-norm	Without A,B & Gyro Z-raw	Without A-C & Acc Z-norm	Without A-D & Acc Y-raw
GBM	97.68%	97.95%	97.41%	97.41%	97%	97.27%

	F	G	H	I	J	K
	Without A-E & Acc Z-raw	Without A-F & Gyro Y-raw	Without A-G & Gyro Y-norm	Without A-H & Gyro X-norm	Without A-I & Film Piezo	Without A-J & Weighted Film Piezo
GBM	97.14%	97.41%	96.73%	97.14%	97.54%	96.32%

Figure 5.9: 4 Class Detection GBM Accuracy: Removing Sensors

5.3 Accuracy

Out of 4 ML algorithms we apply, Bagging with Trees, Boosted Logistic Regression, Boosted Support Vector Machines, Gradient Boosted Machines, Tuned Gradient Boosted Machines, and Random Forest, Boosted SVM at 96.15% yields the highest accuracy rate for our binary classification problem. Many human activity recognition algorithms have been researched and implemented. Additionally, different combinations of non-obtrusive sensors are used for the Catch Me If You Bath Mat system. The final model is made up of 4 sensors and 11 sensor outputs. The following outputs were removed without affecting system accuracy: both raw and normalized values of the gyroscope x-axis, and y-axis. The sensitivity of the system is 98.3 %, and specificity of 99.28%.

For our multiclass classification problem we have an accuracy 97.54% using Gradient Boosted Machines. This robust sensor system uses only 3 sensor inputs, and has 6 sensor outputs. Figure 5.9 indicates that across the various configurations, the optimal sensor outputs are the ceramic piezo sensor, the weighted film piezo, accelerometer x-axis (both raw and normalized values), as well as the gyroscope y-axis norm, and z-axis norm. Compared to the original model, which had 15 sensor outputs. These activity recognition algorithms did not suffer from that common error susceptibility associated with sensor data and noise. Both classification models proved to be very efficient at deciphering the signals generated by the sensors.

Chapter 6

Conclusion

6.1 Summary

In order to live independently, it is critical to minimize the risks associated with fall injuries. Fall analyses have shown that the consequences of a fall are strongly influenced by how quickly you respond and retrieve the user. To provide reliable automatic fall detection, a system must not be dependent on the user's configuration or involvement. The majority of fall detection systems rely either on wearable technology or cameras, both of which are unattractive and easily misused.

The Catch Me If You Bath Mat system is non-intrusive, user-friendly, cost-effective, and does not require an expensive and complex installation process. As it is designed for use in a bathroom, it can distinguish falling from other activities like sitting/lying, walking, or falling objects. This system will use a sensor fusion of an accelerometer, a gyroscope, and three different types of piezoelectric sensors. This will enable it to detect falling up to two feet away from the microcontroller. This system consists of an array of unobtrusive sensors that lie beneath a bath mat. The sensors are quiered at a rate of approximately 2 Hz.

This human activity recognition problem involved multiple different configurations of sensors used in multiple machine learning classification algorithms. After training data was collected and the signals were enhanced. The system started with 5 sensors, the weighted ceramic piezo disc, the unweighted and weighted film piezo, a vibration sensor, and the MPU-6050 module. The resulting binary classification system uses 4 sensor inputs. The vibration sensor was eliminated. The optimal classification model, boosted support vector, is based on sensor outputs, with the exception of raw and normalized values for the gyroscope y-axis and x-axis. It has an accuracy of 96.15%.

Accuracy for the multiclass classification model was 97.54%. Gradient boosting machine is used to perform classification on the resulting three-piece sensor system comprising MPU-6050, ceramic, and weighted film piezoelectric. Data from only the ceramic piezo, the weighted piezo, and the gyroscope y-axis normalized and z-axis normalized were considered. Calculations show that the system put forth is efficient, accurate, and easily implemented in bathrooms. The proposed system, CMBM, offers maximum comfort under a reliable, cost-effective, and robust system.

6.2 Future Work

As part of this paper, we focused only on human-emulated falls using a dummy. The dummy used to collect fall data was 75 pounds and composed of sand, polyester wadding, and pool noodles. In order to ensure the most accurate data to be fed into machine learning algorithms, it would be advisable to redo the fall experiments using real-life human falls. Those falls could be verified by using a watch that can detect when a real fall occurs versus an artificial fall. It is just as important to choose test subjects with a variety of body types, fat distributions, heights, weights, and bone and muscle density. The accuracy of this system suggests that it might be able to detect falls at a distance greater than two feet. To find out the maximum range, further data can be collected in 1 foot increments from the sensors. Figures 6.1 and 6.2 show the experimental setup for this.

In the future, we can enhance our approach by transitioning to a low power board, like the Sparkfun Edge. In addition, we can use a custom PCB that only uses the necessary components. Furthering the idea of being low power, a low-power microphone or infrared sensor can be used to wake up the rest of the system based on a threshold trigger. It will be important to consider communication when moving forward with this project. If a fall is classified, a board with built-in communication, like the Sparkfun Edge, can send data via Bluetooth to a call center. It is important to mention the possibility of false positives and false negatives when discussing communication. In these cases, the system should run through the following procedure where a stationary button that is normally solid green blinks red and beeps if a fall is classified. Users can disable the call for help communication protocol if the system misclassifies a fall.

Furthermore, when considering GMBs for the multiclass classification problem, it is necessary to keep in mind that although these models are highly accurate, they are equally computationally expensive. They commonly require a large number of trees that are time consuming and memory intensive. Its high flexibility results in many parameters influencing the behavior of the approach.

Figure 6.1: Large Scale Floor Setup: 1-5 ft, 1 ft increments

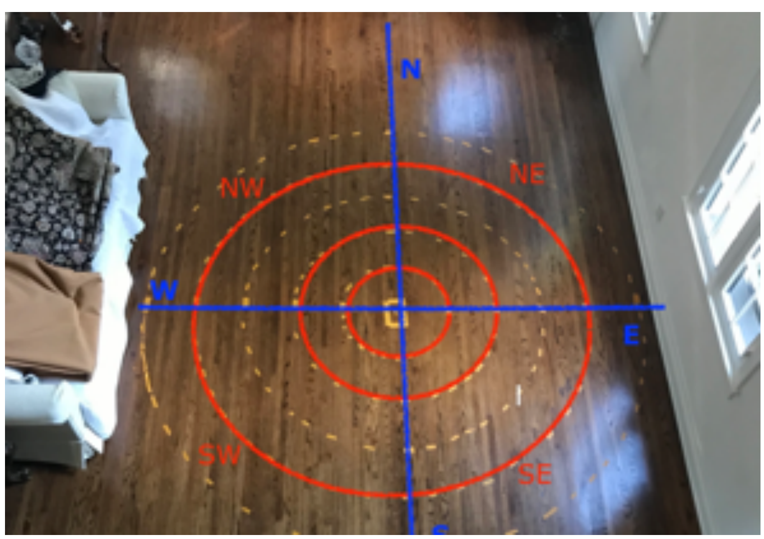


Figure 6.2: Large Scale Floor Setup: Divided into Quadrants

Bibliography

[1] Stefano Abbate, Marco Avvenuti, Guglielmo Cola, Paolo Corsini, J. Light, and Alessio Vecchio. Recognition of false alarms in fall detection systems. pages 23 – 28, 02 2011.

[2] Yogesh Angal and Arti Jagtap. Fall detection system for older adults. In *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, pages 262–266, 2016.

[3] Mark Brackenrig. Gradient boosting algorithms in r. March 2018.

[4] CDC. Older adult falls. 2017.

[5] Comprehensive Integrated Care Center. Vasovagal syncope. 07 2020.

[6] Kabalan Chaccour and Rony Darazi. Smart carpet using differential piezoresistive pressure sensors for elderly fall detection. 10 2015.

[7] Jose Clemente, WenZhan Song, Maria Valero, Fangyu Li, and Xiangyang Liy. Indoor person identification and fall detection through non-intrusive floor seismic sensing. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 417–424, 2019.

[8] Mhd Daher, Ahmad Diab, Maan El Badaoui El Najjar, Mohamad Khalil, and François Charpillet. Elder tracking and fall detection system using smart tiles. *IEEE Sensors Journal*, PP:1–1, 11 2016.

[9] Baratloo A et al. Part 1: Simple definition and calculation of accuracy, sensitivity and specificity. pages 48–49, 2015.

[10] Jane Fleming and Carol Brayne. Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90. *BMJ*, 337, 2008.

[11] Agency for Healthcare Research and MD Quality, Rockville. Healthcare cost and utilization project (hcup). 2012.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337 – 407, 2000.

[13] Sebastian Fudickar, Alexander Lindemann, and Bettina Schnor. Threshold-based fall detection on smart phones. 03 2014.

[14] Mohammad Habib, Mas Mohktar, Shahrul Kamaruzzaman, Kheng-Seang Lim, Maw Tan, and Fatimah Ibrahim. Smartphone-based solutions for fall detection and prevention: Challenges and open issues. *Sensors (Basel, Switzerland)*, 14:7181–208, 04 2014.

[15] Yi He and Ye Li. Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. *International Journal of Distributed Sensor Networks*, 2013, 04 2013.

[16] U.S. DEPARTMENT OF HEALTH, HUMAN SERVICES Centers for Disease Control, and Prevention National Center for Health Statistics. Anthropometric reference data for children and adults: United states, 2015–2018. *Analytical and Epidemiological Studies*, January 2021.

[17] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28, November 2008.

[18] C.J. Lord and D.P. Colvin. Falls in the elderly: Detection and assessment. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society Volume 13: 1991*, pages 1938–1939, 1991.

[19] Kjell Johnson Max Kuhn. *Applied Predictive Modeling*. Springer, 2016.

[20] Noriyuki Okumura, Yusuke Yamanoi, Ryu Kato, and Osamu Yamamura. Fall detection and walking estimation using floor vibration for solitary elderly people. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1437–1442, 2019.

[21] World Health Organization. Falls. 2021.

[22] J. Wiegley. Csunthesis.cls, 2020.

[23] X Yu. Approaches and principles of fall detection for elderly and patient. in proceedings of the 10th international conference on e-health networking, applications and services. *HealthCom*, page 42–47, July 2008.