

# Posture Correction Assistant: Wearable System Using 3 Wireless IMU Sensors to Detect Spinal Posture

Jennifer Choi

*Received June 08, 2024*

*Accepted November 08, 2024*

*Electronic access November 30, 2024*

With the growing use of technology, poor posture has become increasingly common, leading to pain and injury. Current solutions, like postural exercises or posture correctors, can be time-consuming and uncomfortable. Existing methods using inertial sensors to detect posture can be bulky and impractical. This novel approach uses a wearable system with inertial measurement unit (IMU) sensors and machine learning for practical spinal posture detection. The user wears a shirt with sensor pockets for small, lightweight Bluetooth IMU sensors. This system allows users to wear the shirt easily and comfortably. The computer processes the sensor output data using a logistic regression algorithm and returns a posture category of good, mild, or bad. A logistic regression algorithm was used due to its high accuracy compared to other algorithms after testing with the LazyPredict library. Data on the three postures were collected through simulations using a spine model and real human posture data. After comparing twenty-seven machine-learning models, logistic regression had the best accuracy of these models and was used as the final classification model. This approach focuses on sitting posture, as many people sit slouched. No data was collected from the elderly, infants, or people with spinal deformities. Human-subject data, including six participants, were used as training data, and the other five were used as testing data. The posture classification accuracy was 95.8%, and runtime was 0.18 seconds. This wearable system suggests the potential for a user-friendly solution for people lacking awareness of their posture. Possible future studies include collecting more data for increased reliability and transforming it into real-time posture classification software for better practicality.

## Introduction

Poor posture, or postural kyphosis, is the rounding or hunching of a back and has affected 31 million Americans in their lifetimes<sup>1</sup>. Especially with the sedentary lifestyles that many live, people tend to sit in bad posture while working on their computers, scrolling on their phones, watching TV, etc. Poor posture requires phasic skeletal muscle fibres, which quickly run out of energy. While overusing phasic fibres, the longer-lasting static fibres used to maintain good posture are not used and tighten, further worsening posture. These bad sitting habits often cause joint stress since joints are misaligned during bad posture, creating friction that damages cartilage<sup>2</sup>. In a study by Baker, it was discovered that there were “acute negative effects during two hours of prolonged sitting with clinically meaningful increases in discomfort in the low back and hip/thigh/buttock areas. Regarding cognitive function, some deterioration in creative problem solving was observed”<sup>3</sup>. Poor posture can also lead to long-term health issues such as back pain, spinal dysfunction, herniated discs, and other detrimental health conditions. During poor posture, “loss of the natural curve of the cervical spine leads to incrementally increased stresses about the cervical spine. These stresses may lead to early wear, tear, degeneration, and possibly surgeries”<sup>4</sup>. While “mild” postures may not cause as bad of symptoms as “bad” posture initially, it may later transform

into “bad” posture causing the long-term health issues described above. However, many people lack postural awareness and fail to correct themselves, continuing the cycle of bad posture. Practical and technical posture-correcting solutions have been proposed. For example, shoulder and back braces are a potential solution to forcibly correct your posture without having to straighten your back consciously. Nevertheless, overusing these braces without doing the correct exercises can weaken one’s muscles. Regularly doing postural exercises and wearing posture braces is a possible solution; however, it can take up considerable time. Posture braces can also be uncomfortable and not ideal for everyday use. Another possible solution is posture sensing devices such as the Upright Go S. The product consists of a small sensor and an adhesive or necklace the user sticks to their back, which vibrates when it detects bad posture. However, sensor-to-skin contact can cause irritation for some users, and the adhesives are unsuitable for humid conditions<sup>5</sup>. It also only uses one sensor, which does not always provide enough information to determine posture accurately. For example, it can be hard to classify between bending or keeping the back straight but leaning forward.

Another proposed idea is using image classification to detect the type of posture. For example, a proposed approach uses the smartphone camera to detect the type of posture based on facial feature ratios<sup>6</sup>. When one holds their phone at the same height as their face, their features are not distorted. However, when one

---

looks down at their phone, which is at a lower height than their face, their mouth and nose appear larger than their eyes. Based on this knowledge, images can be classified based on facial feature proportion to detect posture. However, this approach is invasive and quickly drains one's phone battery. A similar proposal uses an app called Zen to detect posture through image processing through a computer webcam. The user first sits in good posture so the machine learning system can calibrate its system for the user. To detect poor posture, it uses PoseNet, a convolutional neural network model designed to identify poses through body part recognition. Though it runs through offline artificial intelligence and workers cannot spy on individuals, it could still feel invasive for some individuals. Sitting in front of a camera lens for a long time may be uncomfortable for privacy-conscious users<sup>7</sup>. Using a cell phone or web camera could be accurate, but it is also inefficient for battery life and can feel invasive.

A more non-invasive posture detection method is using IMU (inertial measurement unit sensors) and machine learning. An IMU sensor is an electronic device with an accelerometer, gyroscope, and magnetometer. The three sensors directly measure linear acceleration, rotation rate around axes, and changes in magnetic flux, respectively. These measurements are inputted to a Kalman filter, which fuses the data to output angular acceleration, angular velocity, and the strength/direction of the Earth's magnetic field. These smart wearable devices provide more accurate detection of posture. However, the recent proposals use bulky sensors and materials, making the system less practical. A proposal by Vikas Kumar Sinha uses inertial sensors built into the smartphone to detect posture<sup>8</sup>. The smartphone is attached to the rear upper trunk to monitor posture. However, the smartphone could be heavy, causing the user to feel uncomfortable and making it harder for them to sit in good posture. A study done by Hao-Yuan Tang required three 9-degree-of-freedom inertial sensors, which include a three-axis accelerometer, three-axis gyroscope, and three-axis magnetometer sensor. These sensors were manually fixed to a piece of cloth and then strapped to the individual's body<sup>9</sup>. Time-dependent signals were collected from the sensors, labeled, and then used as data to train the machine-learning model. A long-short-term memory-based recurrent neural network (LSTM-RNN) model was used to classify the posture types and resulted in an accuracy of 98.1%. This high accuracy suggests the potential for machine learning and inertial sensors to classify posture accurately. RNNs are a type of deep learning model that uses sequential data to predict output. However, as the sequence of input data increases, the model's performance decreases. The LSTM model manages the flow of information through gating mechanisms (input, forget, and output gates) to retain long-term dependencies and reduce the risk of vanishing gradients, which minimizes the RNN model's performance loss. While the accuracy was high, only data from six adult

participants was used to train and test the model, causing poor generalization ability; accuracy decreased when testing with the excluded subject rather than 10-fold cross-validation, going from 99.0% to 81.2% accuracy. Furthermore, Another study performed by Wai Yin Wong used three tri-axial accelerometers, sensors that provide acceleration measurements in the x,y, and z-axis. However, these sensors had to be secured to the body using elastic straps.<sup>10</sup> The sensors were secured at three places: between T1 and T2 bones, between T11 and L1, and at the S1 position of the spine. The subject was directed to bend to a certain posture, with sensors measuring acceleration and a Vicon system monitoring the bend angle. The sensors measured acceleration outputs, which were mathematically converted to tilting angles  $\alpha$  and  $\beta$ . The resulting angles used as the observed value were compared with angles from the Vicon system used as the truth value by calculating the root-mean-square error (RMSE), a type of error that measures the average difference between predicted and actual values, for two conditions: quasi-static (sitting still) and dynamic (changing posture) conditions. It had an RMSE of within 5 degrees in quasi-static conditions, which was acceptable for use in a clinical setting for sitting posture. In dynamic conditions, it had a slightly greater RMSE of 8 degrees, meaning using only an accelerometer may not be sufficient for accurate posture angle detection. While both studies required a complex set-up of sensors, they demonstrated a high accuracy of posture detection using IMU sensors. While IMU systems like these may be suitable and accurate for a hospital setting that does not need patients to be very mobile, they may not be practical in an everyday setting and interfere with daily tasks.

The aim of this study is to create a more user-friendly and practical system using IMU sensors and machine learning to classify spinal posture. IMUs were attached to a black t-shirt, distributed in three locations along where the spine would be. Data on three different spinal postures—good, mild, and bad—was collected with the user wearing this shirt. Logistic regression, a machine learning model, was used to classify posture. To raise postural awareness in everyday life, this wearable system may provide the potential for a comfortable and practical method of detecting poor posture.

## Results

The model with the highest accuracy was logistic regression; it had an accuracy of 95.8% and a runtime of 0.18 seconds. It had a weighted-averaged F1 score of 0.958 when averaging all classes' scores. The F1 score combines precision and recall scores and is calculated using the formula:  $2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$ . Unlike accuracy, it captures class-wise performance as well. While the F1 scores for the mild and bad classes were slightly lower than that for the good category, all F1 scores were close to 1, the highest F1 score

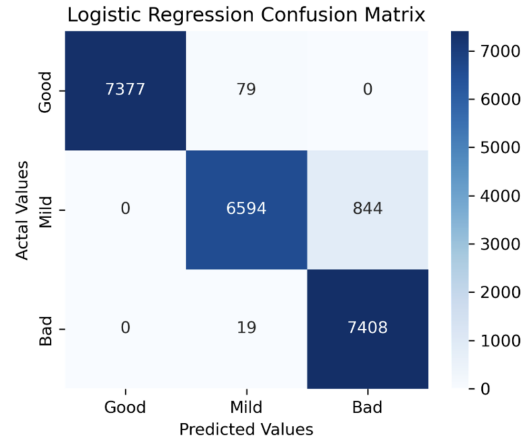
F1 Scores for Logistic Regression

Class	F1
Good	0.995
Mild	0.933
Bad	0.945
Weighted-Average	0.958

**Fig. 1** A table of F1 scores for the logistic regression model for individual classes and weighted average of all classes.

possible, suggesting the model had good recall and precision scores across classes and is a reliable classification method.

The confusion matrix (Fig. 2) was also created to see detailed results, and the precision scores for each class were calculated. A confusion matrix is a table used to visualize the performance of a machine-learning model. The rows represent the true values, while the columns represent the predicted values. Each of the three rows and columns represents a different class: good, mild, or bad posture. Each cell represents the number of predicted values for that category. For example, nineteen mild posture values were predicted in the bad posture class. Most values were true positive, meaning the machine learning algorithm correctly labeled them. The good posture category had 79 false negatives, where the good posture was incorrectly classified as mild posture; this accounted for 1.06% of the total good posture data values. The mild and bad posture categories had 844 and 19 false negative values, respectively, but those false negatives were a small portion of the total labeled data points. For mild posture, around 11.4% of samples were false negatives. For bad posture, around 0.25% of the samples were false positives. True negatives, or correct predictions that a condition does not exist, mainly occur in opposite postures (i.e., good to bad). However, there were cases where the mild correctly predicted false negatives for good vs. mild posture (not predicting good posture for mild posture data points), meaning the model could also differentiate between similar postures. Precision measures observational error, calculating the true positives divided by the sum of true positives and false positives. It measures the accuracy of positive predictions for each class. In this proposal, the classes were the different postures: good, mild, and bad. Good and bad postures were fairly accurate, with few erroneous predictions between categories. Good posture had a precision score of 98%, and bad posture had a precision score of 99%. There was no observed error classifying between good and bad posture with this dataset, but the model had difficulty distinguishing between mild and bad postures. Mild posture had a precision score of 88%, indicating it is slightly less reliable. Another method of model assessment is analyzing recall scores. Recall scores measure how well the model identifies true positives out of all instances of a class and are calculated using the formula: true positives/(true positives+false negatives). The model had recall scores of

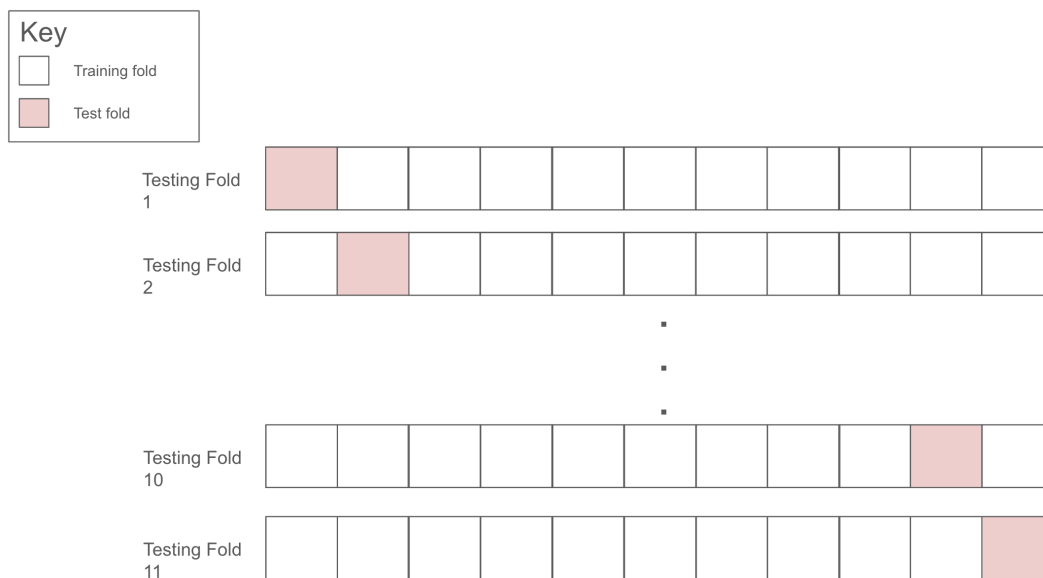


**Fig. 2** A confusion matrix from the logistic regression model

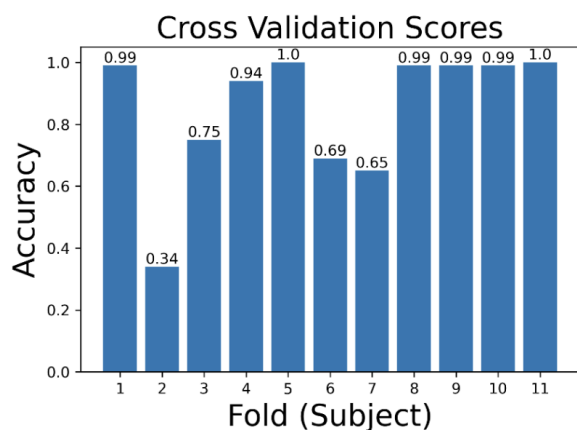
the following: good-98.9%, mild-88.7%, bad-99.7%. Like the precision scores, the model accurately classified most instances of the good and bad classes but struggled with the mild class, meaning the model occasionally misclassified a mild class value as a different class. While the model identifies good and bad posture types, it is less reliable for identifying mild posture.

To further check the reliability of the logistic regression model, a modification of K-fold cross-validation, where K=11, was conducted. Original K-fold cross-validation is a method to evaluate the performance of a machine learning model. It splits the dataset into K-sections, called folds. One fold would be a certain portion of the data. For example, if the data were split into 10 sections, 1 of those sections would be considered a fold. It trains on K-1 to K-9 of the folds and then tests on the K-10 fold. This process is repeated K-times, each time training on a different fold. The percent accuracies from each training and testing are averaged for overall accuracy. For this experiment, the dataset was split into eleven folds, each containing a different human subject’s data. The model was tested using 1 of the folds and trained using the remaining data.

This modification ensured that each person’s training data had similar patterns. If all folds had generally high accuracy, it would imply that the training data was well-suited for this model. If only some folds had high accuracy and others had very low accuracy, it would mean the training data had similar patterns to some subjects but did not represent others’ data. The average cross-validation score was 85%. Compared to the 95.8% testing accuracy, the model did not generalize as well to data from the excluded participant. The individual fold accuracies (Fig. 4) fluctuated depending on the subject. The subject with the highest accuracy had 100% accuracy, while the lowest had 34% accuracy, demonstrating that the model performs well on some subjects but poorly on others. Only one out of eleven subjects



**Fig. 3** A visual representation of K-fold cross-validation where k=11



**Fig. 4** Bar graph of the different k-fold cross-validation scores per fold for the logistic regression model.

had an accuracy of less than 50%, making them an outlier. This suggests that the model may have difficulty with certain unique subjects.

## Discussion

This study suggests that this wearable posture-detection system has the potential to be a comfortable, user-friendly interface that effectively detects the user’s posture. However, the model was slightly overfitted and struggled to classify some mild postures. The lower cross-validation compared to the accuracy suggests overfitting of the logistic regression model. Overfitting is where

a machine learning model is catered to certain training data but does not perform well for testing data. Instead of learning the data and its patterns, it memorizes the specific training data and creates a model precisely for that data. Since it fits the training data too well, it performs poorly for other data. The largely varying individual fold accuracies also show overfitting, as the model was well-suited for certain subjects but performed poorly for others.

Some possible reasons for overfitting may have been the lack of data or variation between subjects. As it was impossible to align different postures precisely and everyone had different heights, the clusters from the data were positioned differently. For example, when graphing the data in a 3d scatter plot, a good posture cluster of one subject may have been in a different location than a good posture cluster for another subject. This may have caused it to be difficult for the model to find patterns separating the three postures. The model may only have suited a certain group of clusters. Furthermore, there were only 75,196 data points (52,874 training data points and 22,322 testing data points), so the model had fewer data to learn from. This could also have led to the overfitting of the model.

A possible reason for the confusion between mild and bad posture is improper human data collection for mild posture. During human data collection, the subject was directed to sit in different postures. However, directing all subjects to sit in the same mild posture was impossible. Since not all subjects sat in the same mild posture, some may have sat in a harder-to-classify posture. This would have resulted in difficulty differentiating mild and bad posture, leading to lower accuracy while classifying mild posture. A possible solution may be

---

implementing a system to measure the angle of the spine bend and establish a range of acceptable angles for each posture. The subjects would be instructed to keep their posture within that range during data collection. While the model may still struggle to differentiate the two postures due to differences in spine formation and height, any difficulty due to unclear postures from data collection would be eliminated.

Limitations to this proposal include imperfect data collection, lack of data, and the inability to perform live posture detection. During the spine model and human data collection, bending the model to imitate the posture definitions shown in Figure 5 was impossible. While imperfect data collection can introduce variability that may simulate real-world scenarios, since everyone sits in different postures and the three posture definitions are more general, it could also result in inaccurate data collection. The spine model/human subject could accidentally have modeled a different posture than what they were indicated to model, resulting in inaccurate data. More precise posture definitions could be established instead of rough boundaries. Precise posture definitions would make it possible to accurately determine whether the subject was sitting in the right posture. There was also a lack of data, with only 11 human subjects' data and 5 spine model trials. Only using a few human subjects led to a lack of variation and limited representation of different people. Collecting more data from people of varying ages, heights, and body frames could make a more realistic and reliable model for a more accurate population representation. The spine model was also only one size, so it was unable to simulate a variety of people. Different size spine models could be used to collect data for a more accurate representation. Data were collected in three locations: a sofa, a chair at a dining table, and chairs designed for lab desks. In real life, there are many more locations people sit in. In future studies, collecting data in more locations, such as a bed or floor, could help create a more reliable posture detection method. The amount of data was also smaller, which may have contributed to the overfitting of the model and a lower cross-validation accuracy. However, more data collection could easily solve this problem. Finally, the proposed system could not detect posture live: data had to be collected through a Raspberry Pi and then manually sent to the computer for classification. In the future, data could be collected and analyzed in real-time using a mobile app. With improvement in future studies, a final prototype could be used for accurate, real-time posture monitoring with IMU sensors. This device could be used daily to stay aware of one's posture or for healthcare professionals to monitor patients' postural habits.

Despite the drawbacks, the system's 95.8% accuracy suggests that if integrated into a device to work in real-time, it could autonomously detect postural changes and work as an efficient reminder to improve one's posture. The high precision and accuracy scores of 98% and 99% for "good" and "bad" postures, respectively, show an accurate ability to differentiate between the

two postures; there is minimal confusion when differentiating these from each other and the mild category. While the "mild" posture category had a precision score of 88%, the only confusion was with the "bad" category. The primary goal of this research is to determine when the user's posture needs adjustment. Since "mild" posture is considered similar to a warning, it is also in the unacceptable category along with "bad" posture. Since the model accurately distinguishes between the unacceptable and acceptable postures ("good" category), it effectively serves its purpose. It uses noticeably smaller sensors and a more flexible system than existing methods. Tang's study required three Next Generation IMUs, which are 46g each, to be directly fastened on top of the user's shirt using tightly secured orange straps and zip ties to achieve an accuracy of 98.1%<sup>9</sup>. Similarly, Wong's approach requires sensors to be attached directly to the skin<sup>10</sup>. Han's approach requires sensors to be attached directly to the ear and neck<sup>11</sup>. This proposal uses three METAMOTIONRL sensors, which are only 0.2g per sensor, and are put into sewed-on shirt pockets for more flexibility in movement and comfort for 95.8% accuracy. While there is a slight trade-off for accuracy, the difference is relatively small, and this proposal can effectively differentiate between acceptable and unacceptable postures.

This proposal suggests the potential for a more comfortable, practical posture detection system using inertial sensors and machine learning; the machine learning model has a 95.8% accuracy and accurately differentiates between the unacceptable ("bad"/"mild") and acceptable ("good") category. While the lack of data and the inability to use real-time functionality make it unsuitable for use in its current state, future studies could address these issues. Furthermore, compared to previous studies, this system utilizes a lighter, more flexible, and more comfortable system, which may be more suitable for use in daily life to provide an accurate and user-friendly posture-detecting system.

## Methods

### I. Sitting Posture Definitions

This proposal aims to classify posture based on 3 definitions: good, mild, and bad. These posture categories are based on the severity of one's posture. The posture definitions were motivated by Han's approach to monitoring forward head posture using a magnet-magnetometer pair, which included a "Normal," "Warning," and "Danger" category<sup>11</sup>. The clear categorization allows for practical usage for alerting the user of overall posture severity, compared to categories like right-side bend or back movement. The "mild" category in this proposal was to warn that the user may begin to be in an unhealthy posture. Good posture is medically defined as when an individual's ears are on the same plane as their shoulder. However, there are no official definitions for mild and bad posture. Two definitions for mild



**Fig. 5** Posture definitions for the three established postures while sitting on a chair on level-floor: good (left), mild (middle), and bad (right)

and bad posture were made to address this issue. Mild posture was defined as a slight arch of the back and a head slightly forward of the shoulders. A slight arch is where most of the lower back relies on support from the back of the chair but is still relatively straight. Bad posture is with a severe arch of the back, forward head posture, and rounded shoulders. A severe arch is where the lower quarter of the back is touching the back of the chair, but the rest of the body is bent forward, and gravity is towards the front. Figure 5 illustrates the three types of posture. During human subject data collection, the subject's posture was compared to the respective posture picture, then told to adjust their posture accordingly so it matches as best as possible.

## II. Sensing Device

METAMOTIONRL sensors were used to log and stream data on movement during different postures. Using three sensors compared to two or one resulted in a higher accuracy of 98.1% in a proposal to detect upper body posture with inertial sensors and recurrent neural networks<sup>9</sup>. Due to this high accuracy, three sensors were used in this proposal. The sensors come in a white plastic case and are lightweight, Bluetooth, and convenient for everyday use. These 9-DOF (degrees of freedom) sensors get measurements from a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer. An accelerometer measures acceleration ( $meters^2/second$ ), a gyroscope measures angular velocity (degrees/second), and a magnetometer measures the strength and direction of the magnetic field (tesla). It also comes with automatic calibration, which provides accurate sensor outputs for various individuals. Compared to the manual calibration from Wong's proposal<sup>10</sup>, this provides quicker and more convenient calibration of the sensor devices.

During data collection, these sensors were connected to a Raspberry Pi Model B via Bluetooth to receive the

sensor outputs. Based on human posture recognition using accelerometer sensors, the Raspberry Pi had a lower power consumption than an Odroid O2+ board<sup>12</sup>. The data from the Raspberry Pi was then sent to a Macbook Air for further analysis in Jupyter Notebook, a web application for developing code.

## III. Data Collection

This study conducted two types of data collection: human data collection and spine model data collection. The purpose of also collecting data from the spine model was to simulate human data in a more accessible manner, reducing the need for a greater number of human participants. Human data collection requires personnel and additional time, but the spine model can be used multiple times as needed without another person, providing a more time-efficient and simpler data collection. For human data collection, the subject must wear a shirt, insert the sensors, and sit on a chair. This process took a few minutes. For spine model data collection, the only set-up needed was inserting the sensors, which took approximately 20 seconds to do. The data collection produced nine data files for each subject's posture: accelerometer, gyroscope, and magnetometer value files for each of the 3 sensors. The accelerometer measured acceleration ( $meters\ per\ second^2$ ), the gyroscope measured angular velocity (radians per second), and the magnetometer measured magnetic field strength (tesla). Each data file (accelerometer, gyroscope, and magnetometer) contained columns of timestamps and x, y, and z measurements.

### A. Human Data Collection

Motion data using the three IMU sensors were collected for eleven people. Subjects included eight teenagers and three adults, all of various heights. Subjects were required not to have spinal deformities. All subjects signed an informed consent form before data collection, and all data was de-identified so no personal information could be extracted from the data collected. No discussions directly addressed a specific individual, and data was not shared with others. Data for each posture was collected for 30 seconds (30 seconds of good, 30 seconds of mild, and 30 seconds of bad). Out of all the subjects, three had two trials: sitting on a sofa and sitting on a chair at a dinner table. The remaining eight subjects had one trial: sitting on an industrial chair for labs. A shirt with sensor pockets was created to make it easy to wear the sensors (Fig. 6). 32 Degrees Heat Tees were used in three sizes: small, medium, and large. Surgical tape was sewn on the back of the shirt to create sensor pockets.

During the actual data collection, the subject was directed to wear the shirt and sensors. Afterward, they would sit on a chair and bend according to the posture (Fig. 7): 30 seconds of good, 30 seconds of mild, and 30 seconds of bad posture. The subject was shown each picture in Figure 5 and told to replicate and hold the posture shown for 30 seconds to the best of their ability. An example of bad posture data collection is shown in Figure 7.



**Fig. 6** Size medium 32 Degree Heat short-sleeve shirt with attached sensor pockets made up of surgical tape along the spine location. The shirt is laid flat on a level surface.



**Fig. 7** Example of bad posture data collection: sitting on a chair, wearing a size Medium shirt over regular clothes. The back and neck are hunched, ears being in front of shoulders rather than aligned

While sitting in these postures, the subject was directed to move front/back and side to side slightly, as if they were doing their daily tasks (i.e., working on a computer). The Raspberry Pi was connected to a laptop and placed nearby to receive and record the data. Separate data files were recorded for each posture, and the subject's posture was adjusted to best match the posture shown in the photo definitions (refer to Fig. 5) before data was recorded.

### **B. Spine Model Data Collection**

Sensors were tightly secured onto a life-sized spine model using paper towels and Scotch tape to make pockets and surgical tape to secure the pockets for spine model data collection (Fig. 8). The location of the bottom sensor was at L5 of the spine, the same as the placement of the sensors on the bottom spine, as used in an approach for upper body recognitions using inertial sensors and neural networks<sup>9</sup>. Furthermore, the top sensor was placed on T1, while the middle sensor was placed at T10. Sensor locations were picked so the sensors were as evenly spread out on



**Fig. 8** Spine model on spine model hanger with sensors secured with surgical tape, paper towel, and tape.

the spine as possible. Similar to human data collection, separate data files were recorded for each posture. The spine model was seated, then manually bent for 30 seconds to model each posture (Fig. 9). It was seated on a chair and sofa, two common locations many people spend much time at. For each posture, five trials were performed, and the spine model was slightly shaken and moved for more realistic data collection.

#### **IV. Classification Method**

The dataset used the combined data from human and spine model data collection; there were a total of 57,341 data points, with 20,836 in the good category, 17,790 in the mild category, and 18,715 in the bad posture category. Raw data included occasional missing values, and the first and last few time stamps were often incomplete or included values from the person or spine model moving into the correct posture position. Those data points were deleted from the dataset.

Due to the timestamp difference between the three sensors, an estimation method called straight linear interpolation was used to predict values at the same timestamps. Interpolation estimates unknown values that fall within the range of known data points. Straight linear interpolation, specifically, fits data points with straight lines, ensuring the interpolated points form

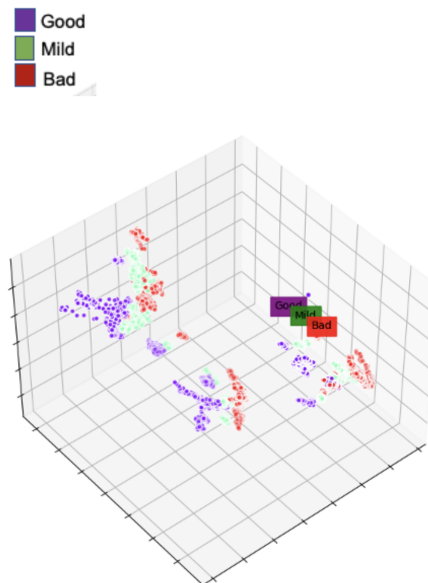


**Fig. 9** The spine model with attached sensors is on a chair, manually bent to mild posture (the hands bending it are not shown in picture).

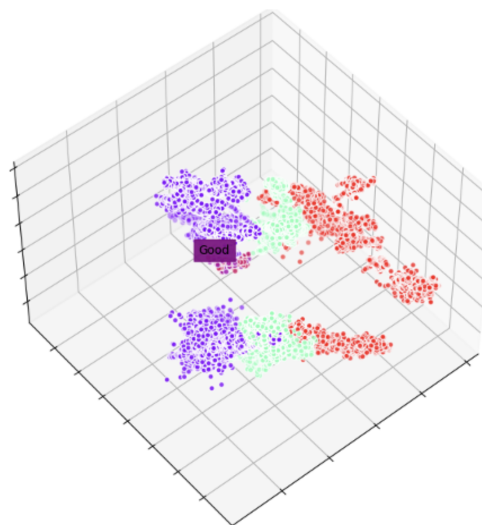
a line between two data points. Afterward, the data were merged, creating 1 data file per posture recording.

To further simplify the data for better classification, Principal Component Analysis (PCA), a dimensionality reduction method, was used to transform the 27-dimension data into 3 dimensions. Like this approach, Balli also used PCA to transform raw data before inputting it into classification models for human activity recognition<sup>13</sup>. Furthermore, when different dimensionality reduction methods, including multidimensional scaling and t-distributed stochastic neighbor embedding, were applied, and the data was visualized on a 3d scatter plot, the points were not organized into clusters by posture and had no clear pattern. PCA is a technique for reducing dimensionality while maintaining general patterns and minimizing loss of information, which allows processed data to carry variability in data. Each point from the dataset contained 27 dimensions, which could also be seen as variables. The most important variables that explained the variability in the dataset were found through PCA. The original data was then transformed into a three-dimensional dataset using this information, so the variability and clustering were kept as much as possible. The principal component number was set as three for computational efficiency and a more straightforward dataset for the model to work with. Furthermore, the individual sensors (accelerometer, gyroscope, magnetometer) had three axes, so having a principal component of three represents variability in a 3d space. The initial graphing using PCA on all 27 dimensions was sparsely clustered into small, nonoptimal groups for classification (Fig. 10). While the different postures were in separate clusters, the different subjects and trials were also separated. The additional distinction between subjects and trials would make it hard for the machine learning algorithm to find patterns distinguishing each posture. To solve this problem, PCA was attempted on different combinations of features (accelerometer, magnetometer, gyroscope) and compared to see which had the best clustering between postures. After comparing different feature combinations, using PCA on only accelerometer data was found to have the best clustering (Figure 11). While there were some overlapping and outliers, the data was clearly clustered into the three posture types.

LazyPredict, a public Python package, was used to automatically test and find the accuracies of 27 different machine learning models to find the best classification method. Data was split into train and test sets with 36,739 training data and 20,602 test data. LazyPredict only utilizes train and test set data and does not use validation data, so no validation set was separately created. LazyPredict also recorded the accuracy, F1 score, and runtime for each model. Out of all the models, logistic regression had the highest accuracy of these models and, thus, was used for the final classification algorithm. Logistic regression is a widely used model that performs various mathematical functions on the data to find the class with the highest probability. The input data



**Fig. 10** The top image shows the plot legend: good posture is purple, mild is green, and bad is red. The figure below displays 3D scatter plot of accelerometer, magnetometer, and gyroscope data after PCA clustering.



**Fig. 11** 3d scatter plot of PCA clustering on only accelerometer data. Purple data points are for good posture, green is for mild posture, and red is for bad posture.

---

was the PCA-performed accelerometer data, and the output was a type of posture: good, mild, or bad.

## Conclusion

This study presents a novel wearable system using IMU sensors and machine learning for spinal posture detection, divided into three categories: good, mild, and bad. Bluetooth-enabled IMU sensors are attached to a flexible shirt, which collects time-series motion data. Data was collected using a spine model that was manually bent, and 11 human participants were told to bend to resemble the three different postures. After cleaning and preprocessing the data with PCA, it was fed into LazyPredict, which tested it for 27 machine-learning models. Logistic regression produced the best results with an accuracy of 95.8% and a runtime of 0.18 seconds. Upon k-fold cross-validation (k=11), the accuracy was 85%, which shows overfitting. However, this was most likely due to a lack of data shown by the contrasting accuracies for each fold, and it could be improved with further data collection from more individuals. This system has the potential to become a robust, real-time posture-detecting device with further development and data collection.

## Acknowledgments

I would like to thank my teachers, Mrs. Messenger and Ms. Lee, for guiding me through this process.

## References

- 1 J. Hull, *Top 5 most common causes of poor posture*, 2021, Retrieved from <https://www.hullchiropractic.com/blog/283705-top-5-most-common-causes-of-poor-posture>.
- 2 *Posture*, 2015, Retrieved from <https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/posture>.
- 3 R. Baker, P. Coenen, E. Howie, A. Williamson and L. Straker, *The Short Term Musculoskeletal and Cognitive Effects of Prolonged Sitting during Office Computer Work*, 2018.
- 4 K. K. Hansraj, *Assessment of Stresses in the Cervical Spine Caused by Posture and Position of the Head*, 2014.
- 5 J. Walters, *Upright GO2 posture trainer review*, 2023, Retrieved from <https://www.forbes.com/health/body/upright-go-2-posture-trainer-review/>.
- 6 W. Lawanont, M. Inoue, P. Mongkolnam and C. Nukoolkit, *Neck posture monitoring system based on image detection and smartphone sensors using the prolonged usage classification concept*, 2018.
- 7 N. Lomas, *Back pain? This app uses your webcam to detect bad posture and says it doesn't spy...*, 2022, Retrieved from <https://techcrunch.com/2022/04/29/zen-pre-seed-posture-correction/>.
- 8 V. K. Sinha, K. K. Patro, P. Pławiak and A. J. Prakash, *Smartphone-based human sitting behaviors recognition using inertial sensor*, 2021.
- 9 H. Tang, S. Tan, T. Su, C. Chiang and H. Chen, *Upper body posture recognition using inertial sensors and recurrent neural networks*, 2021.
- 10 W. Y. Wong and M. S. Wong, *Detecting spinal posture change in sitting positions with tri-axial accelerometers*, 2008.
- 11 H. Han, H. Jang and S. W. Yoon, *Novel wearable monitoring system of forward head posture assisted by magnet-magnetometer pair and machine learning*, 2020.
- 12 A. Leone, G. Rescio, A. Caroppo, P. Siciliano and A. Manni, *Human postures recognition by accelerometer sensor and ML architecture integrated in embedded platforms: benchmarking and performance evaluation*, 2023.
- 13 S. Balli, E. A. Sağbaş and M. Peker, *Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm*, 2018.

Model	Model description	Accuracy	F1 Score	Time Taken
LogisticRegression	Models the probability of a binary outcome using a logistic function	0.96	0.96	0.18
LogisticRegressionCV	Same as LogisticRegression but uses built-in cross-validation to find the best regularization parameter automatically.	0.93	0.93	3.41
QuadraticDiscriminantAnalysis	Models each class with its covariance matrix and assumes that the data distribution is Gaussian. Fits a quadratic decision boundary between classes	0.92	0.92	0.06
GaussianNB	A variant of Naive Bayes that assumes features follow a Gaussian (normal) distribution	0.92	0.92	0.05
SVC	Finds a decision boundary that maximally separates the classes	0.92	0.92	21.09
LinearDiscriminantAnalysis	Assumes different classes generate data based on different Gaussian distributions with the same covariance matrix, creating a linear decision	0.92	0.92	0.07
NearestCentroid	classifies each sample to the class of the nearest centroid (geometric center) of the data points.	0.90	0.90	0.07
MLPClassifier	Learns a non-linear function by using one or more hidden layers to classify the categories	0.90	0.89	13.34
CalibratedClassifierCV	Estimates parameter of and calibrates classifier using cross-validation	0.89	0.88	4.39
PassiveAggressiveClassifier	Makes changes to the model when predictions are incorrect and keeps model the same when correct.	0.86	0.85	0.12
LinearSVC	A variant of the SVC algorithm that uses a linear kernel.	0.83	0.82	1.31
BaggingClassifier	Fits multiple base classifiers on random subsets of the data and combines their predictions	0.80	0.78	0.99
SGDClassifier	Linear classifier that uses stochastic gradient descent to optimize the model parameters	0.80	0.78	0.24
DecisionTreeClassifier	Non-parametric, uses a tree-like data structure to classify data	0.79	0.77	0.69
GradientBoostingClassifier	Builds multiple decision trees sequentially, where each tree corrects the errors of its predecessor	0.79	0.79	22.95
RandomForestClassifier	Builds multiple decision trees and merges results	0.79	0.78	4.71
XGBClassifier	Scalable implementation of gradient boosting, combines prediction of decision trees using weighted average for final prediction	0.77	0.76	7.08
RadiusNeighborsClassifier	Variant of KNeighborsClassifier, decision is based on the number of neighbors within a fixed radius rather than a set number of neighbors	0.77	0.77	14.71
Perceptron	Linear binary classifier that updates its weight parameters based on the misclassified examples	0.76	0.76	0.11
BernoulliNB	Naive Bayes for binary features. Assumes feature independence and Bernoulli distribution.	0.75	0.75	0.06
ExtraTreesClassifier	Trains decision trees on subsets of data, averages results for prediction	0.74	0.73	1.50
KNeighborsClassifier	Classifies based on the majority vote of data neighbors	0.73	0.72	0.68
RidgeClassifier	Linear classifier that penalizes large weights to avoid overfitting	0.67	0.53	0.09
RidgeClassifierCV	Same as RidgeClassifier but automatically finds the best penalty strength through cross-validation	0.67	0.53	0.11
ExtraTreeClassifier	A single decision tree that splits randomly	0.62	0.59	0.06
AdaBoostClassifier	Combines multiple weak classifiers to increase the accuracy of classifiers	0.53	0.53	2.02
DummyClassifier	Predicts based on simple rules like most frequent class, stratified, or random	0.33	0.17	0.04

**Fig. 12** LazyPredict results, including the model name, brief model description, respective accuracy, F1 score, and model runtime. Accuracy, F1 score, and runtime were rounded to the nearest hundredth.