Classifying common hand gestures using gyroscopic sensor data from a mobile device

**Ashley Hunt**   
Computer Science  
University of Nottingham

**Amit Kumar**  
Data Science  
University of Nottingham

**Shreeya Kumbhoje**  
Data Science  
University of Nottingham

*AI is the future, and it basically involves the usage of computers to ease out daily human tasks, which makes it crucial for the computer to understand how a human carries out a task. However, for the computer to perform accurately it must be trained. ML is the subset of artificial intelligence which focuses on the training of the computer either by using Supervised(labelled data) learning, Unsupervised(unlabelled data), Semi-supervised Learning or Reinforcement Learning. Our aim here is to train the computer using the labelled data so that it can classify between different hand gestures the human being performs.*

*The data was collected using a phyphox app compatible on smartphones which uses the accelerometer sensors inside the device. The data has gone through a set of preprocessing techniques like application of a low-pass filter, Fourier transformation and PCA to refine the extracted features given to the model for training. We have deployed ML models like SVM, Random Forest Classifier, XGB, KNN and some Recurrent Neural Network Models like LSTM and GRU looking for a optimum fit model for multi class classification between gestures 1. Drawing a circle, 2. Waving, 3. Gesturing come here, 4. Gesturing go away.*

*The outcome shows a good \_\_% accuracy for \_\_\_ ML model in comparison with other ML models implemented, and the accuracy of \_\_% LSTM(RNN model) and \_\_\_% for GRU(RNN model).*

# Introduction

Making a computer understand human gesture is a step towards it[3].Gesture recognition pertains to recognizing meaningful expressions of motion by a human, involving the hands, arms, face, head, and/or body[8]. With a variety of application across different fields including monitoring activities and vital signs, enhancing gaming experience, and supporting applications in robotics and wearable technology. Our intent is to use smartphones which is a practically used device to establish communication possibilities between humans and machines differing from the traditional data collection methods. we further explore the usage of this technology to create accessibility features for individuals with disabilities which could be a potential project within the scope of man machine interaction.

In recent years, gesture recognition system has become very popular in the field of research, especially facial and hand gesture recognition system[3] gesture classification for hand movement can be classified into static and dynamic depending on the data acquisition method. In the past a significant amount of research has been carried out which investigates machine algorithms like Support Vector Machine[1][3] , Random Forest[1] , Hidden Markov Model which analyses data to discover patterns. A few Deep Learning models like CNN[1][6] and some hybrid models like CNN-LSTM[1] mostly focusing on image processing, which has now made it a common subject in the domain. However a limited amount of research focuses on using signal data collected as a .dataset(.csv file) using a normally accessible device among humans today.

# Background

Quote dump:

"The highest performance was achieved using a convolutional neural network with long- short term memory (CNN-LSTM)"

"The pre-processing step consisted of a fourth-order low-pass Butterworth filtering with a cut-off frequency of 5 Hz. Such frequency was chosen due to the frequency content of the acquired signals, which was below 5–6 Hz, as verified by Fourier power spectrum analysis. The pre-processing also included raw signal standardization: each signal acquired with the MMR sensors was centred to have zero mean and standard deviation equal to one."

"Concerning ML approaches, inspired by human activity recognition work in the literature, the following classifiers were evaluated: Support Vector Machine (SVM) [19,20], Random Forest (RF) and K-Nearest Neighbour (KNN)."

"The complete dataset, composed of 22 subjects, was divided into training/validation and test sets using leave-one-subject-out cross validation. As concerns ML methods, the training/validation dataset was split into 5 folds, and grid search-based 5-fold cross validation was used for tuning the hyperparameters. As concerns DL methods, validation was performed on 20% of the training/validation set."

"In multi-gesture classification, CNN and CNN-LSTM performed better than ML methods, with CNN-LSTM resulting in the highest balanced accuracy"

"In the binary classification, the highest f1-score and precision values were obtained with CNN" (Moccia, Sara et al.)

"Support vector machine (SVM) and hidden Markov model (HMM) are two typical machine learning methods in pattern recognition."

"SVM usually needs careful feature extraction and selection, as well as other traditional ML algorithms like naïve Bayes (NB), K-nearest neighbors (KNN), and decision tree (DT)"

"For time sequential signals, recurrent neural networks (RNN) allow information to persist. Thus, we can make full use of the contextual information of the sequence. However, RNN might suffer from a vanishing gradient problem with long data sequences. To solve this problem, Hochreiter et al. designed a special RNN, the long short-term memory (LSTM) network that can learn long dependencies" (Zhao, Shenglin et al.)

# Methodology

To achieve our aims of producing a classification model for four gestures (come, go, wave and circle), the raw gyroscopic data is heavily pre-processed and undergoes several key transformations to contribute to the classification process. Only the most useful features are taken through to the model training phase.

## Data Collection

Data was collected using the Phyphox mobile application[[1]](#footnote-1). Each member of the team attempted to perform the same four gestures for eight iterations per recording, and five recordings each per gesture – 40 iterations per person per gesture for a total of 480 complete gestures.

These gestures were initially recorded continuously, without stopping, as per the coursework specification but after trial and error we discovered that segmenting the data was a lot more difficult when there were no clear spaces between the gestures. Instead, we opted to leave a distinct one second gap between performing each gesture in each recording. Adding these pauses allowed us to segment the recording more easily into its separate gesture patterns.

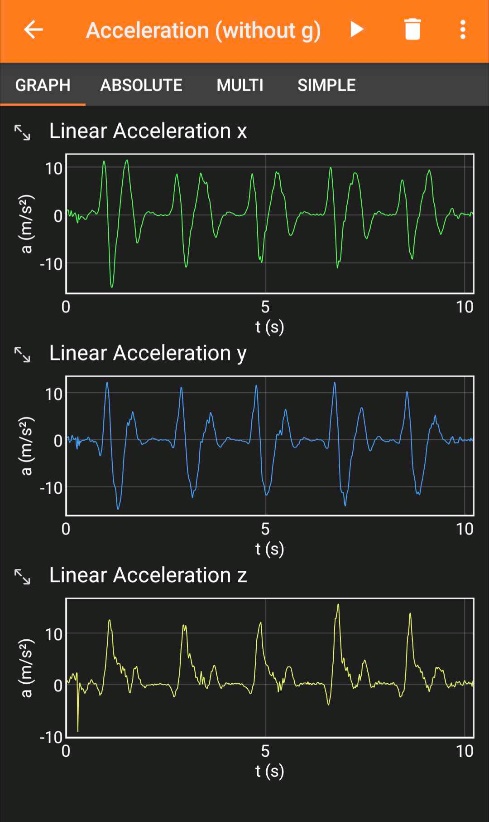


Figure 1: Screenshot of the Phyphox recording interface.

## Data Visualization

Once the data was recorded and the associated csv files were imported into *pandas* *DataFrame* we applied a basic trim to the data. Inspecting, from the beginning of the file, every 20 data points. If the mean acceleration was < 0.3 then the data was removed. We did this from the beginning of the file as well as the end of the file. This would help us reduce unnecessary noise at the beginning and end of the file, making segmentation easier.

Next, we began to explore the data by inspecting the raw sensor data and applying visualization techniques. This would allow us to understand the data and what features of the data would play a significant role in classification.

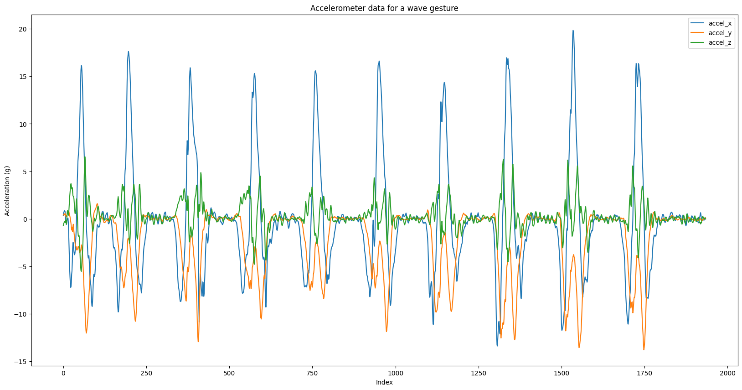


Figure 2: Raw gyroscopic data for a wave gesture recording.

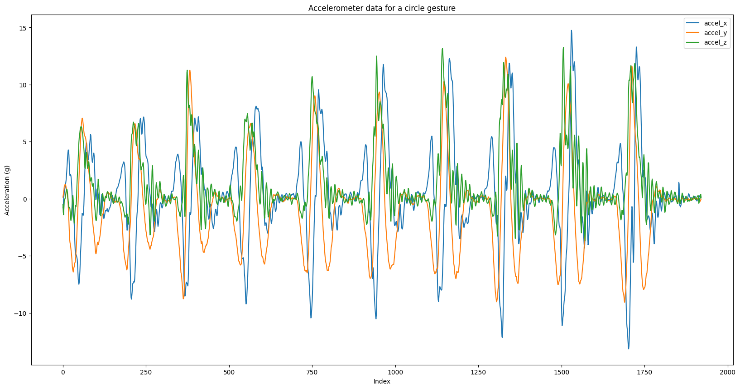


Figure 3: Raw gyroscopic data for a circle gesture recording.

## Data Pre-Processing

Firstly, to balance the dataset and ensure that the model would not prioritise one gesture we found which gesture had the least number of recordings and then dropped all recordings greater than that number.

To begin our pre-processing, we used the *scipy.signal* *butter* and *filtfilt* functions, applying a low-pass frequency filter to the data. This filter would apply dampening to the sensor data and reduce noise. Visually, this made the waveforms smoother and less jittery. This was done to ensure that the model was not simply reading all the noise in the system but understanding the more generic shape of the data.

We also applied an exponentially weighted moving average (EWMA) filter at this stage. EWMA filtering was performed so that spikes in the data were dampened. This dampening was especially important when considering recordings that contained many different types of gestures as some gestures tended to have much higher acceleration than other, simply due to the nature of the gesture.

Next, we applied a *scikit-learn* *MinMaxScaler* to the data – which transforms the data to values between 0 and 1 whilst maintaining the relationships between the datapoints, this is done using the below formula:

We performed this transformation to ensure that the machine learning models did not perceive higher values as more important for training.

Next, we applied a Fourier transformation. Fourier transformation can be used to decompose a signal into its underlying frequencies, which can be useful for removing noise from the gyroscopic data as well as determining the most important parts of the signal that define the gestures.

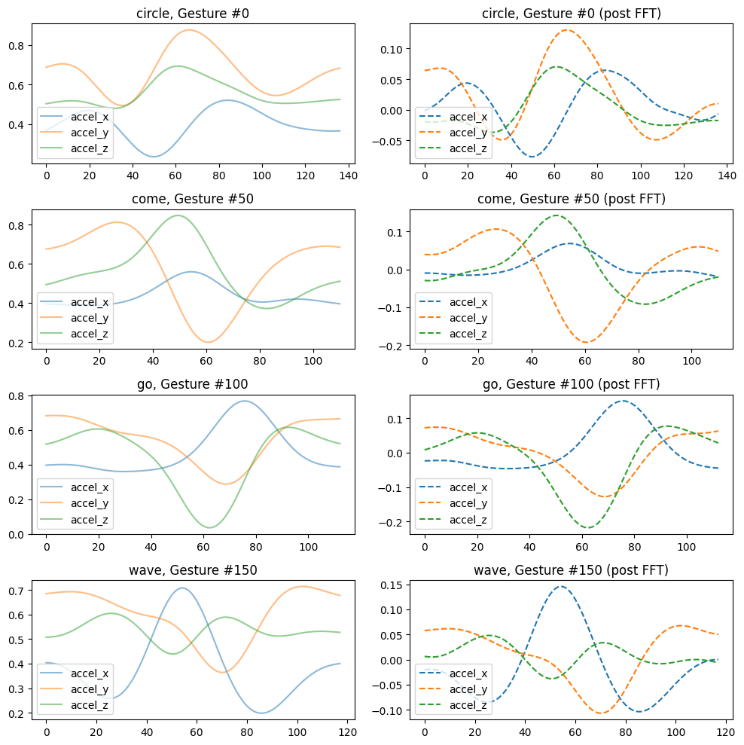


Figure 4: Example gesture data before and after Fourier transformation and filtering.

Dynamic time warping (DTW) can also be used to find the similarities/distance between different waves, which can be used to shift rotate the waves so that slight misalignments in the time domain are not factors in the segmentation and eventual feature extraction of the gestures. DTW was considered a stretch goal for the project.

## Segmenting the Data

The next step of processing was to separate the individual gestures from the files that contained 6-9 gestures of the same type. This would be required so that each gesture was separated in its own block of data to be associated with a label. Our first attempt to apply this separation was to visually inspect the data and its rolling average and deviation – this would help us identify an appropriate value to decide on when one gesture was starting, and another was ending.

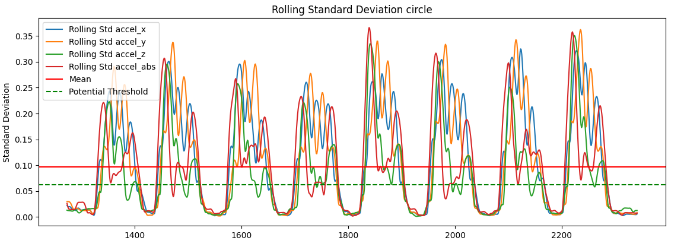


Figure 5:Rolling standard deviation of a circle gesture recording.

This figure outlines our suggested splitting point (any points where the acceleration standard deviation is lower than the threshold, in green). Using these points, we could segment the file into its separate gestures ready for feature extraction.

An alternative approach was then adopted where we utilised *scipy.signal find\_peaks* to automatically detect peaks in absolute acceleration, and from those peaks expand it’s boundaries until the acceleration was below some threshold (indicating the end of the gesture).

A graph of a graph

Description automatically generated

Figure 6: Absolute acceleration graph for a circle recording, showing the peaks in the data that represent the gestures.

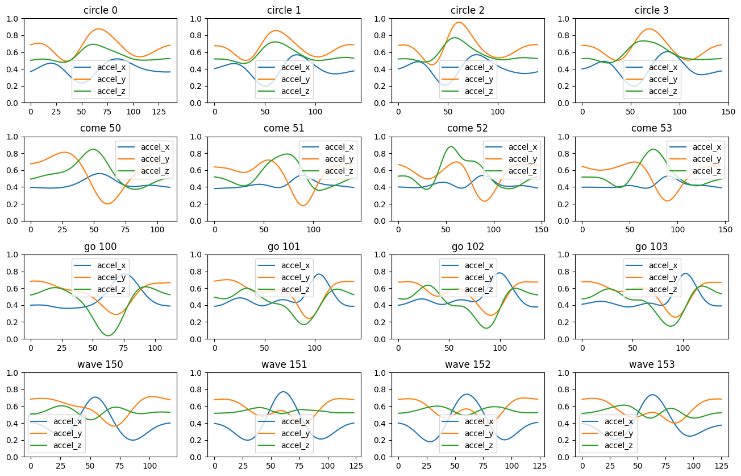


Figure 7: A preview of the segmented gesture data.

## Feature Extraction

For the model to accept our data we would need to transform the data once again so that each gesture’s data was the same shape. To achieve this requirement, we would segment the data, this time with each gesture being broken up into distinct sections with some overlap. Then using these slices of data, we could extract key features such as: mean, standard deviation, median, skew and kurtosis. This would result in data that is a summary of each slice of the gesture. These features could then be visualised and taken through the feature selection process.

## Feature selection

Once all features were extracted, we visually inspected the averages of the features to find the features that had the clearest disparity between the gestures. Feature selection is critical to reduce the computational and time cost of classification, as well as increase the model’s accuracy.

A graph of different colored lines

Description automatically generated

Figure 8: Feature extraction/visualisation for the mean of each segment.

A graph with different colored lines and dots

Description automatically generated

Figure 9: Feature extraction/visualisation for the standard deviation of each segment.

A graph with colorful lines and dots

Description automatically generated

Figure 10: Feature extraction/visualisation for the kurtosis of each segment.

Originally, we had planned to use these visualisations to find features that provided clear separation between the gestures and to drop the features that did not provide strong correlation. However, this process allowed room for ambiguity and did not allow easy automation. Instead, principal component analysis was used to automate the process of feature aggregation and selection.

A group of graphs with different colored dots

Description automatically generated

Figure 11: Principal component analysis visualisation for n=3.

## Model selection and parameter hypertuning

HERE…

# Results

Our model performance was drastically reduced when adding the fourth gesture type, including both come and go.

## Machine learning Models

### K-means Clustering

Despite the appearance of well-defined clusters, the model has a silhouette score of only 0.62, an accuracy of 0.31 on testing data and a validation accuracy of 0.23. Increasing the number of principle components in the data drastically reduces the accuracy of the model to almost zero.

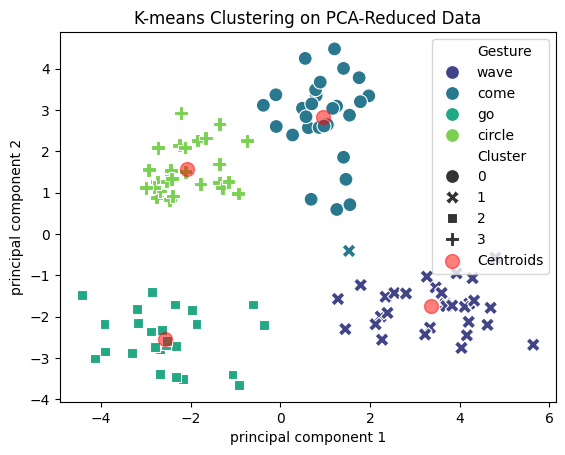


Figure 13: K-means clustering visualisation for four gestures.

Blah

### K-nearest neighbours (KNN)

Blah

### Random Forest Classifier

The Random Forest classifier was implemented to predict gestures using data from an accelerometer sensor. The dataset was partitioned into training, and testing sets using train\_test\_split function. Features were standardized using StandardScaler to ensure uniformity across different features. A *RandomForestClassifier* was used with a predefined random state for reproducibility. Grid search was applied to fine-tune hyperparameters such as the number of estimators, maximum depth, minimum samples split, and minimum samples leaf. Cross-validation (cv=3) was used with balanced accuracy as the scoring metric. The best parameters obtained from the grid search were {'n\_estimators': 250, 'max\_depth': None, 'min\_samples\_split': 10, 'min\_samples\_leaf': 2}. The optimal model was then used to predict gestures on the testing set. The test set achieved a balanced accuracy of 69%, thereby providing an indication of the model's overall predictive ability.

### Support Vector Machines (SVM)

Bl

The Support Vector Machine (SVM) classifier was used to classify gestures using a dataset that was split into training and testing sets using train\_test\_split function. The data was pre-processed using standard scaling, and a grid search with cross-validation was carried out to optimize the SVM's hyperparameters, focusing on the regularization parameter C, kernel type, and gamma value. The optimal parameters were C=0.1, using an 'rbf' kernel, and 'scale' for gamma.

After training, the SVM model achieved a balanced accuracy of approximately 54.84% on the test set, indicating its ability to generalize to unseen data. While this accuracy may be considered modest, it provides valuable insights into the classification performance of the SVM for the given gesture recognition task.

It's worth noting that the process involved substantial computational resources, and attempts were made to fine-tune the parameters by experimenting with various kernal types and C values, but these adjustments did not yield significant improvements in performance.

Overall, accuracy for SVM model has shown its potential for gesture classification, although further exploration, feature engineering, or alternative algorithms could be considered to enhance accuracy further.

### XGBoost Classifier

The XGBoost classifier was used for gesture classification on the dataset. The dataset was divided into training and testing sets, and a grid search was performed to optimize the hyperparameters of the XGBoost model. The parameters used were number of estimators (trees) in the ensemble, maximum depth of each tree, learning rate, and subsampling ratio. The best parameters identified through the grid search were an ensemble of 300 trees with a maximum depth of 20, a learning rate of 0.1, a subsampling ratio of 0.8.

After training, it achieved a balanced accuracy of approximately 68.17% on the test set. This performance indicates the model's ability to classify gestures based on the features.

In comparison to SVM, the XGBoost model demonstrated a higher balanced accuracy, suggesting its potential as a robust classifier for gesture recognition. Further exploration could involve feature engineering, or additional parameter tuning to potentially enhance classification performance even further.

## Neural Network Architectures

### Gated Recurrent Unit (GRU)

Blah

### Long Short-Term Memory layer (LSTM)

Blah

| Table Head | EXAMPLE TABLE HEADER | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

# Discussions

* Limitations
* Stretch goals (DTW)

# Conclusion

Classifying gestures from mobile device sensor can be difficult due to the variation in movement that is possible from user on a mobile device.

# References

1. Akhavian, R. and Behzadan, A.H. (2015) ‘Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers’, *Advanced engineering informatics*, 29(4), pp. 867–877. Available at: https://doi.org/10.1016/j.aei.2015.03.001.
2. Moccia, Sara et al. “Automated Classification of Hand Gestures Using a Wristband and Machine Learning for Possible Application in Pill Intake Monitoring.” Computer methods and programs in biomedicine 219 (2022): 106753–106753. Web.
3. Odhiambo, Chrisogonas Odero et al. “Detecting Medication-Taking Gestures Using Machine Learning and Accelerometer Data Collected via Smartwatch Technology: Instrument Validation Study.” JMIR human factors 10 (2023): e42714–e42714. Web.
4. Zhao, Shenglin et al. “Hand Gesture Recognition on a Resource-Limited Interactive Wristband.” Sensors (Basel, Switzerland) 21.17 (2021): 5713. Web.
5. Siddiqui, Uzma Abid et al. “Wearable-Sensors-Based Platform for Gesture Recognition of Autism Spectrum Disorder Children Using Machine Learning Algorithms.” Sensors (Basel, Switzerland) 21.10 (2021): 3319. Web.
6. Yu, Yi et al. “Exploration of Chinese Sign Language Recognition Using Wearable Sensors Based on Deep Belief Net.” IEEE journal of biomedical and health informatics 24.5 (2020): 1310–1320. Web.
7. Han H, Yoon SW. Gyroscope-Based Continuous Human Hand Gesture Recognition for Multi-Modal Wearable Input Device for Human Machine Interaction. Sensors (Basel). 2019 Jun 5;19(11):2562. doi: 10.3390/s19112562. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603535/>
8. Stancic, I. et al. (2022) ‘Comparison and Evaluation of Machine Learning-Based Classification of Hand Gestures Captured by Inertial Sensors’, Computation, 10(9), p. 159. Available at: <https://doi.org/10.3390/computation10090159>.

1. https://phyphox.org/ [↑](#footnote-ref-1)