Implementation Based Comparative Analysis of Hand Gesture Recognition using Acceleration Data

1st Parth Ashwinbhai Bhalodiya
MSc Computer science
University of Nottingham
Nottingham, United Kingdom
psxpb12@nottingham.ac.uk

2nd Deepak Kumar Singh
MSc Computer science
University of Nottingham
Nottingham, United Kingdom
psxds7@nottingham.ac.uk

***Abstract*—This research gives a comparative analysis of acceleration-based hand gesture recognition algorithms. The study's goal is to assess several implementation approaches to determine the most efficient and accurate way. The study includes collecting acceleration data with an accelerometer sensor and processing it with various algorithms such as linear SVC, Decision Trees, random forest classifiers, and Artificial Neural Networks (ANNs). The comparison was based on each implementation's recognition accuracy, computational complexity, and response time. In terms of accuracy and response time, the results demonstrate that Gradient boosting decision tree and ANN (feed forward network) outperform the other algorithms. The study delves into the possibilities of employing acceleration data for hand gesture detection and emphasises the importance of selecting the best implementation approach to achieve improved accuracy and efficiency.**

***Index Terms*—Human Gesture Recognition, ANN, acceleration data(without g), signal smoothing, ML classifier**

I. Introduction

Recent years have seen an increase in the number of mobile devices with sensors, which has resulted in the availability of huge volumes of data that may be used for numerous purposes. The accelerometer is one such sensor that monitors acceleration along three axes. Accelerometer data has found widespread application in sectors such as activity recognition, motion analysis, and gesture recognition. In this regard, the dataset generated by the Phyphox app is a useful resource for investigating the potential of acceleration data in several disciplines.

The Phyphox app is a versatile tool that allows users to collect sensor data from their smartphones, including acceleration measurements. The created dataset offers 4 different gesture such as circle, wave, come here and go away. Firstly, it provides 10 trials of each gesture performed by two users (5 each), enabling the exploration of inter-user variations and generalization of recognition models. Each trial consists of 15 iterations of gesture signal. This dataset, obtained using the Phyphox app without gravitational information, captures the raw acceleration values in three dimensions (x, y, and z) as recorded by the device. The acceleration data is in the form of signals which has a sampling frequency of 100Hz.

What are the techniques used in preprocessing stage to get the best results for live feed data? What are the most effective approaches to increase the accuracy of hand gesture recognition while working on real time data? How many features from signal data we need to make better model? How to segment each gesture's signal?

II. Literature Review

The paper ”Sensor Dataglove for Real-time Static and Dynamic Hand Gesture Recognition” by presents a system for real-time recognition of hand gestures using a sensor dataglove [2]. They have collected data from 35 volunteers performing 14 static and 3 dynamic gestures wearing the dataglove, including go away, come here. The paper describes the hardware setup of the sensor dataglove and explains the data acquisition process. The collected sensor data is processed and analyzed using machine learning techniques. The authors employ a Support Vector Machine (SVM), decision tree, random forest classifier to classify the hand gestures based on the extracted features [2]. In ”Fig. 1” demonstrates the effectiveness of the proposed approach, achieving high accuracy in recognizing a wide range of hand gestures. The system shows robustness to various environmental conditions and real-time responsiveness. For static gestures, the SVM classifier faces difficulties classifying ‘Come Here’, ‘Five’, ‘Fist’, and ‘Stop’ gestures and RnF classifier faces difficulties classifying the ‘Come Here’ gesture. Decision Tree shows quite a good accuracy and does not face any crucial complication for any particular gesture [2].

Static Gestures		Dynamic Gestures	
Classifier	Accuracy	Classifier	Accuracy
KNN (K = 3)	99.53%	KNN (K = 7)	98.64%
Decision Tree	98.74%	SVM (RBF)	97.96%
RnF	97.48%	Decision Tree	97.96%
SVM (RBF)	94.97%	RnF	97.28%

Fig. 1. Comparative Results [2]

The paper titled ”Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching” presents a novel approach for hand gesture recognition using accelerometer data [1]. The authors propose a two-stage framework consisting of a neural network-based classification stage followed by a similarity matching stage. In the first stage, a neural network model is trained to classify hand gestures based on accelerometer data. The neural network architecture includes multiple hidden layers to capture the intricate relationships between input features and gesture classes [1]. In the second stage, similarity matching is employed to refine the classification results. This stage utilizes a similarity measure to compare the input gesture with a database of reference gestures and selects the most similar gesture as the final recognition output. In ”Fig. 2” experimental results demonstrate the effectiveness of the proposed approach, achieving high accuracy in hand gesture recognition tasks [1]. The detailed user-independent basic gesture recognition results are listed in Table 2, and the average recognition accuracy is 98.88

COMPARISON OF BASIC GESTURE RECOGNITION ACCURACIES (%)

	R	UR	U	UL	L	LL	D	LR
User-Dependent	99	100	100	100	100	100	100	100
User-Independent	97	100	100	100	98	100	99	97

Fig. 2. Comparison of accuracies [1]

The paper titled ”Automated classification of hand gestures using a wristband and machine learning for possible application in pill intake monitoring” presents a study on the automated classification of hand gestures using a wristband device and machine learning techniques. The objective of the research is to explore the potential application of this technology in monitoring pill intake [3]. The authors propose a system that utilizes a wristband equipped with various sensors to capture hand movements during specific gestures. They extract features from the sensor data, including acceleration and angular velocity, to represent the hand gestures. These features are then used as inputs to train machine learning models [3]. ”Fig. 3” gives the list of features. The findings suggest that the developed system has potential for application in pill intake monitoring, where hand gestures can be used as a means of tracking medication intake [3].

	Features
Time domain	Root mean square (RMS)
	Variance
	Mean absolute deviation (MAD)
	Kurtosis skewness
	Interquartile range (IQR)
Frequency domain	Energy
	Spectral entropy
	Mean frequency of power spectrum
	Median frequency of power spectrum

Fig. 3. Set of features used in model training [3]

III. METHODOLOGY

In this section, two approaches for the classification of hand gestures using accelerometer has been shown below each approach has three section.

A. First Approach

a) Pre-processing: After plotting the data, noise is detected in the data at the beginning and finish of the signal, which does not demonstrate the signal data. As a result, for each gesture trial, the signal data points are averaged per 10 window size. This was done to remove the signal's undesired high frequency components, which are noise and provide no value to our classifier. Because the beginning and finish data are not part of the gestures, they may be simply cut from the signal. Take note that 1/4 sec of the signal is cropped from the beginning, but 1/2 sec of the signal is cropped from the conclusion. This was chosen since stopping the Phyphox app takes longer. To bring the entire signal to the origin of the y axis, the mean of the signal was subtracted from the signal data points. This was done to bring the signals to the same scale and improve visualisation ”Fig. 4”. The length of gestures signal was plotted for all four gestures of each trial. The visualisation of signal length data reveals an imbalance in signal length ”Fig. 5”. This imbalance data should be eliminated since it will generate an imbalance in the number of rows for each gesture, causing the model to not fit properly and produce unsatisfactory results. To correct the imbalance, some of the signals were clipped from both ends to make their lengths equal. Because the majority of the faulty data is located at the ends, signals were clipped from both ends ”Fig. 6”.

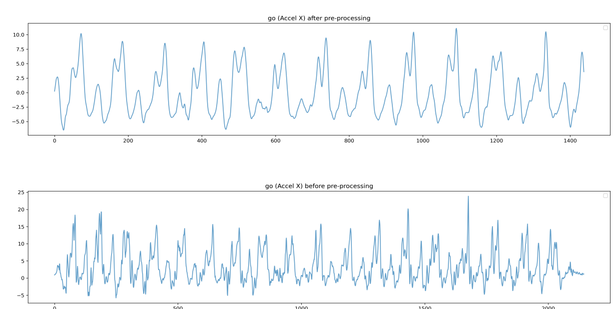

Fig. 4. Comparison after Initial noise cleaning

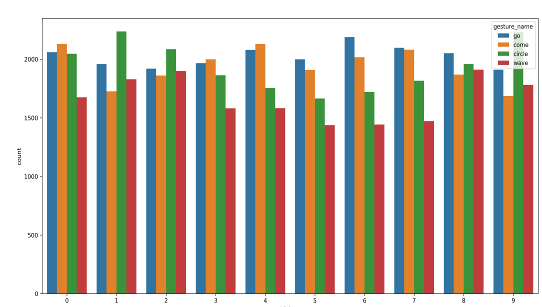

Fig. 5. Bar plot of Imbalanced data

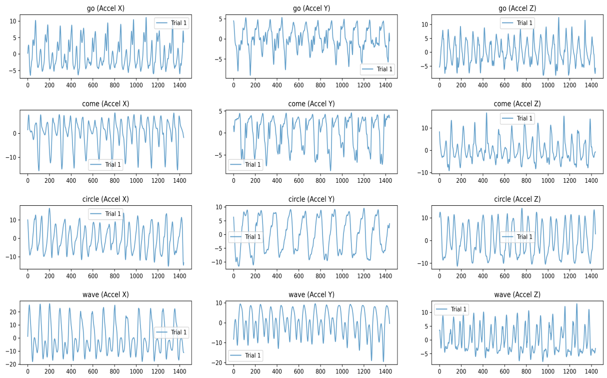

Fig. 6. Data after final signal filtering

b) Exploratory data analysis: A summation acceleration graph is created "Fig. 7", with each data point representing the sum of all prior accelerations for the magnitude signal of the trials.

$$A[i] = sum(a[i] + a[i-1] + \ldots\ldots + a[1] + a[0]) \quad (1)$$

This graph shows whether each gesture's acceleration is predominantly positive or mostly negative. The circle graph indicates a steady increase in acceleration with each repeat of the movements made in the trials. This signifies that there is more positive acceleration than negative acceleration for each repeat of the circle gesture. Gesture come and go follows a similar path but negative trend. It could be due to the similarities in how we conduct the two gestures. The wave gesture nearly sticks to the origin, implying that the positive and negative acceleration in each gesture repetition is nearly equal. This demonstrates a difference in the mean of summation acceleration in the circle gesture against other gestures. We plotted the box chart to check the discrepancies; now we can clearly see the difference in mean of circle and other gestures, but we can also detect several outliers in the wave signals. The goal of creating this graph was to compare the acceleration of four different movements.

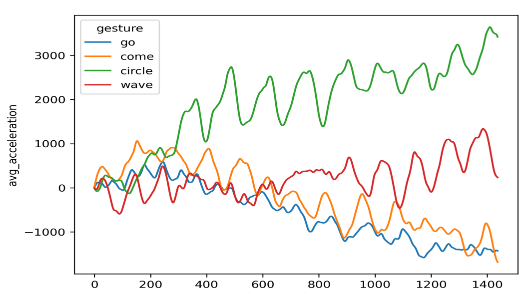

Fig. 7. Average acceleration each gesture

c) Data Modelling/Classification: Each signal was separated into two-second frames with a one-second overlap. This overlapping was carried out to increase the resolution analysis. Then, from each frame, features were retrieved. In total, 30 features were retrieved from the frame data in three dimensions, including mean, correlation, and acceleration values. The feature extraction section was inspired by [1] and was researched and implemented on three axes of the signal, whereas it was only implemented on two axes in [1]. For feature selection, a correlation matrix was used. Nine features in the correlation matrix have no association with any other features. Those features were eliminated, and 21 features were used for classification.

Linear SVC is the first classifier employed. It was utilised with the assumption that if the data is linearly separable by decision boundary, linear SVC will provide high accuracy in less time than other classifiers. When Linear SVC did not produce the expected results, it indicates that nonlinear interactions between features and target variables must be modelled. The Random Forest classifier was then utilised because it predicts using numerous Decision Trees, and Decision Trees are good at modelling nonlinear relationships. Random Forest produced competitive results; however Gradient Boosting Decision Trees may have outperformed it by achieving high predictive accuracy by progressively training decision trees that repair the flaws of the preceding trees. Grid search is used for hyperparameter tuning in all classifiers for systematically exploring different hyperparameter combinations to find the best-performing model. The parameters in the paramgrid specify the hyperparameter search space for the all model. These parameters are chosen based on domain knowledge and related studies. The exhaustive search allows for comprehensive evaluation through cross-validation, ensuring robust generalization and simplifying the hyperparameter selection process.

B. Second Approach

a) Pre-Processing: Down sampling was used to reduce the data imbalance since the signal information should not be damaged because it would be used later in frequency analysis. Trimming the data reduces the length of the signal and removes some signal information. Down sampling removes some information as well, but it has no effect on the frequency analysis.

b) Exploratory data analysis: The frequency graph of the signals shows no significant variation between the motions "Fig. 8". As a result, the energy of the signals was estimated, and the difference in the energy of all four gestures was plainly apparent. The entropy graph was then constructed based on the energy trend "Fig. 9".

c) Data Modelling/Classification: As a result, the best qualities that can be retrieved from the signals are energy and entropy. However, two features each axis for a total of 6 features in 3 axes for a signal frame is insufficient for a classifier. As a result, the signals were routed via four low pass filters before the energy and entropy of those four signals were calculated with a total feature count of 24. The wavelet transform was utilised to create four low pass filters since it automatically calculates filter frequencies. "Fig. 10" depicts the perplexity diagram of filtered signal.

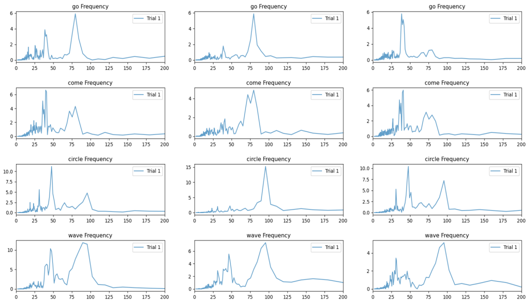

Fig. 8. Frequency graph

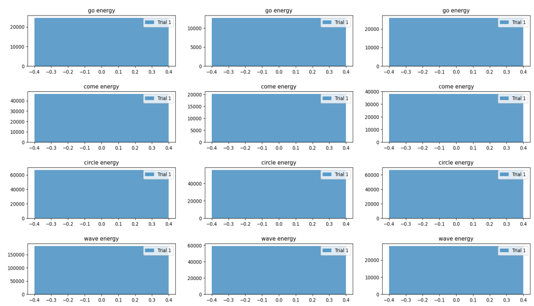

Fig. 9. Energy graph

Because accelerometer data from hand gestures frequently contain nonlinear correlations between the input features (e.g., accelerometer readings) and the corresponding gestures, a Feed Forward network is employed for classification. Through the application of activation functions in their hidden layers, FFNs are capable of capturing and modelling these nonlinear interactions. This enables the network to learn sophisticated mappings between the accelerometer data input and the corresponding hand motions.

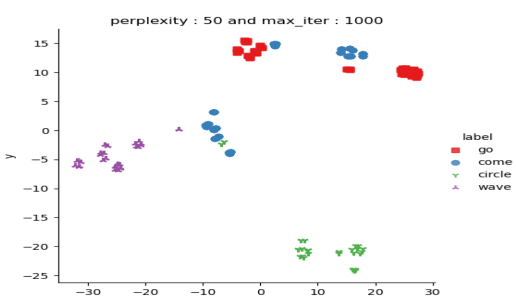

Fig. 10. Perplexity diagram

IV. Results

a) Decision Tree with Gradient Boosting: The decision tree with gradient boosting ensemble demonstrated impressive results in gesture recognition "Fig 11". It achieved an accuracy of 95 %, precision of 0.95, recall of 0.95, and F1-score of 0.95. The model showcased excellent performance in distinguishing between different gestures, with a well-balanced classification performance across the classes. It showed high precision and recall values for most gestures, indicating its effectiveness in correctly identifying the intended gestures.

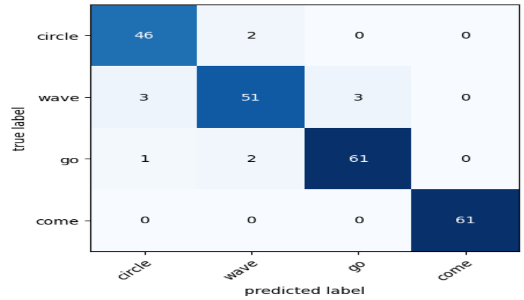

Fig. 11. Confusion metrics of Decision Tree with Gradient Boosting classifier

b) Artificial Neural Network (ANN): The ANN classifier achieved an accuracy of 99%, precision of 0.99, recall of 0.99, and F1-score of 1.0. The model demonstrated competitive performance in gesture recognition but fell slightly behind the decision tree with gradient boosting. The ANN "Fig. 12" showed good precision and recall values for most gestures, but it exhibited relatively lower performance for a few specific gestures, resulting in slightly lower overall metrics.

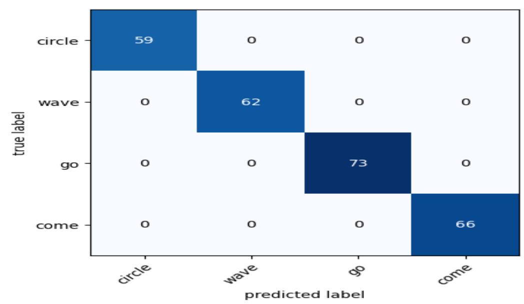

Fig. 12. Confusion metrics of FNN classifier

c) Random Forest: The Random Forest model delivered promising results, achieving an accuracy of 93%, precision of 0.93, recall of 0.93, and F1-score of 0.93. It showcased strong performance in gesture recognition, with balanced precision and recall values across most gestures. The ensemble nature of Random Forest allowed it to capture a comprehensive representation of the dataset, resulting in accurate classification "Fig. 13".

d) Linear SVC: The Linear SVC classifier achieved an accuracy of 86%, precision of 0.85, recall of 0.85, and F1-score of 0.85. While it exhibited competitive performance, it fell slightly behind the other models in terms of overall metrics. The Linear SVC "Fig. 14"showed good precision and recall values for most gestures but struggled comparatively with a few specific gestures.

In the end "Fig. 15" comparative analysis of the classifier which have been used in this research.

V. Discussion

We conducted a comparative analysis of four popular machine learning models, namely Decision Tree with Gradient

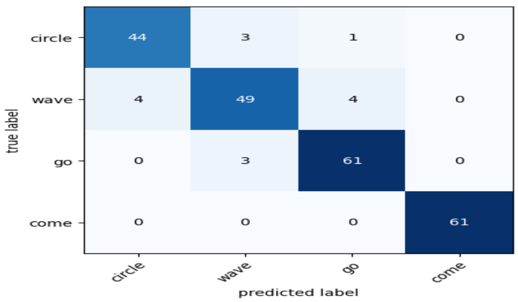

Fig. 13. Confusion metrics of Random forest classifier

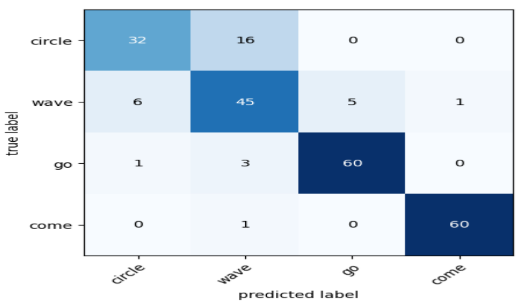

Fig. 14. Confusion metrics of Linear SVC classifier

Boosting, Feed Forward Network (FNN), Random Forest, and Linear SVC, for gesture recognition using accelerometer sensor data. The objective was to assess their performance and determine their suitability for this specific application. Our dataset consisted of accelerometer sensor readings for four gestures circle, wave, come here and go away. The dataset was pre-processed and carefully prepared for classification, ensuring a balanced distribution of classes. To evaluate the models' performance, we employed standard classification metrics, including accuracy, precision, recall, and F1-score. Additionally, we examined the confusion matrices to gain insights into the models' performance for different gesture classes.

The decision tree with gradient boosting, Random Forest, and Linear SVC exhibited similar and strong performance in gesture recognition, with accuracy ranging from 85% to 95% and F1-scores ranging from 0.84 to 0.95. FNN, while achieves the accuracy of 99% and F1 score 1.0. FNN is giving better result than all the other classifiers is because the features used in FNN model is completely different from features used in other classifiers. Energy and Entropy features used in FNN is making the task of classifier simpler. The reason for better performance of FNN mostly depends on the extracted features. Both Random Forest and gradient boosting with decision trees share the decision tree structure. In both cases, decision trees are the base models used to build the ensemble. Since decision trees are the fundamental building blocks, it's possible that the individual decision trees in both models are making similar decisions and capturing similar patterns in the accelerometer data.

FNN and Linear SVC often require more computational resources and training time compared to decision tree-based models. Decision tree-based models (decision tree with gradient boosting and Random Forest) typically have faster training and prediction times compared to FNN and Linear SVC. This advantage may be particularly important in real-time gesture recognition applications where quick responses are required. Linear SVC gives less accuracy than Random Forest and gradient boosting with decision trees but the accuracy is competitive. The reason for its less accuracy could be its high sensitivity to outliers as outliers can have significant impact on decision boundary. In smaller datasets, the limited number of samples may not fully represent the underlying distribution of the data. As a result, Linear SVC may struggle to find an accurate linear decision boundary that can generalize well to unseen data points.

The FNN model in paper [1] has achieved an accuracy 99% for dependent user and 98% for independent user. The dataset taken for [1] is very large whereas dataset taken for this experiment was very small and made by only 2 persons and both the persons data were used to train the model so it is a case of dependent user. FNN classifier used here achieved the accuracy of 99% for dependent user which same accuracy obtained in the paper [1]. The difference in both the model is feature extraction step. In feature extraction the paper focuses on the acceleration data in different dimensions but our features are energy and entropy of 4 low pass filter signals in x, y and z axes. The Random Forest and SVM model is getting 95% and 97% accuracy in paper [2]. In this paper the accuracy is for Random Forest comes out to be 93% and Linear SVC 86%. Linear SVC model can be compared with SVM because both works on decision boundary. The reason behind better accuracy could be the difference in sensor data as this paper's [2] sensor data has gyroscope And orientation angle data as well but that is not the case with this experiment.

Classifier	Accuracy	Precision	Recall	F1-score
FNN	100%	1	1	1
Linear SVC	86%	0.85	0.84	0.85
Random Forest	93.5%	0.93	0.93	0.93
GBDT (Gradient Boosting Decision tree)	95%	0.95	0.95	0.95

Fig. 15. Comparative Analysis

VI. Conclusion and Future Recommendation

The performance of each algorithm is assessed using metrics such as accuracy, precision, recall, and F1-score. The results are analyzed and compared to identify the most effective algorithm for hand gesture recognition using acceleration data. Additionally, the impact of different feature extraction methods and parameter settings on the recognition performance is investigated. The findings of this study provide valuable insights into the implementation aspects of hand gesture recognition systems using acceleration data. The comparative analysis helps in understanding the strengths and weaknesses of different machine learning algorithms in this context.

Hand gesture recognition using analytics from accelerometer data holds immense promise for the future. The area poised to gain from this innovation spans diverse fields such as human-computer interaction, virtual reality, and robotics applications; thus presenting endless possibilities for its adoption moving forward. With respect to robotics specifically, implementing this advanced technology would facilitate smoother interactions between humans and robots by accurately understanding the signals communicated via their gesticulations. Consequently resulting in robust cooperation between these two entities with improved efficacy in following human instructions flawlessly.

Additionally , health experts also have much to gain from deploying these gadgets as they can closely monitor patients' movements during physiotherapy sessions. The potential for hand gesture detection utilising accelerometer data is enormous in the future. This technology is anticipated to grow more precise and effective as it develops, increasing the likelihood that it will be adopted widely across a range of industries.

References

[1] R. Xie and J. Cao, "Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching," in IEEE Sensors Journal, vol. 16, no. 11, pp. 4537-4545, June1, 2016, doi: 10.1109/JSEN.2016.2546942.

[2] M. A. A. Faisal, F. F. Abir and M. U. Ahmed, "Sensor Dataglove for Real-time Static and Dynamic Hand Gesture Recognition," 2021 Joint 10th International Conference on Informatics, Electronics Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision Pattern Recognition (icIVPR), Kitakyushu, Japan, 2021, pp. 1-7. doi: 10.1109/ICIEVicIVPR52578.2021.9564226.

[3] Moccia, S., Solbiati, S., Khornegah, M., Bossi, F.F. and Caiani, E.G., 2022. 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, p.106753. doi: 10.1016/j.cmpb.2022.106753