Smart Moves: Decoding Hand Gestures through Machine Learning Techniques

Arun Veeriampalayam Soundararajan
MSc. Computer Science with AI
psxav7@nottingham.ac.uk

Nidhi Saini
MSc. Data Science
psxns5@nottingham.ac.uk

Girija Suresh Dahibhate
MSc. Data Science
psxgd2@nottingham.ac.uk

***Abstract*—Gesture recognition systems have emerged as pivotal components in human-computer interaction, promising intuitive user experiences across digital platforms. This study investigates the utilization of smartphone sensors and machine learning techniques for decoding and classifying hand gestures. By analyzing accelerometer data collected via the Phyphox application, various classification models including Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, AdaBoost Classifier, and Support Vector Classifier (SVC) are evaluated for their efficacy in identifying hand gestures accurately. Encompassing a diverse array of gestures such as circular motions and directional cues like beckoning and dismissing, the study addresses challenges like variations in execution styles and user diversity. Rigorous data collection procedures yield a comprehensive dataset, serving as the foundation for exploring machine learning methodologies. The findings underscore the effectiveness of Gradient Boosting Classifier and Random Forest Classifier in achieving high accuracy in gesture classification tasks, thus advancing human-computer interaction.**

***Index Terms*—Gesture recognition, smartphone sensors, machine learning, accelerometer data, classification models, human-computer interaction, Feature Extraction, Classification models**

I. Introduction

In the realm of human-computer interaction, the integration of gesture recognition systems has emerged as a pivotal area of research, promising to revolutionize user experience across various digital platforms. Hand gestures, serving as a natural and intuitive form of communication, offer compelling opportunities for seamless interaction, particularly within the mobile technology landscape. By deciphering and interpreting these gestures, devices can respond in real-time, offering users a more intuitive and immersive interface.

This study embarks on an exploration of gesture recognition, with a specific focus on leveraging smartphone sensors and machine learning techniques to decode and classify hand gestures. Utilizing accelerometer data collected via the Phyphox application, the objective is to evaluate the effectiveness of various classification models, including Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, AdaBoost Classifier, and Support Vector Classifier (SVC), in accurately identifying and categorizing distinct hand gestures.

The scope of gestures under examination encompasses a diverse range, including circular motions, waving gestures, and directional cues such as beckoning and dismissing. Each gesture presents its own set of challenges, from variations in execution styles to the inherent variability introduced by different users. The study endeavors to capture and account for this natural variability, thereby enhancing the robustness and accuracy of the gesture recognition system.

Meticulous data collection procedures, involving multiple participants and sessions, have facilitated the creation of a comprehensive dataset encompassing various scenarios and gesture executions. This dataset serves as the foundation for the exploration of machine learning methodologies.

The subsequent sections of this report delve into the intricacies of the machine learning methodologies employed, presenting the findings and insights gleaned from this exploration. Through this endeavor, the study aims to contribute to the ongoing discourse surrounding gesture recognition systems, driving innovation and advancement in the field of human-computer interaction.

II. Literature Review

This review examines the advancements in accelerometer-based gesture recognition systems, focusing on their application to enhance human-computer interaction across mobile and wearable technologies. The review explores studies that address efficient classification and recognition of gestures, showcasing technological progress and identifying critical research gaps.

A. Technological Advancements in Gesture Recognition

- **Innovative Neural Network Architectures:**
  Studies like [6] have introduced advanced models such as the Residual PairNet and PairNet with Inception, which leverage accelerometers and gyroscopes to recognize sequences of hand gestures, proving effective for smart device applications.
- **Novel Sensor Applications:**
  Research documented in [8] demonstrates how accelerometer data can be integrated into digital TV remote controls, enhancing user interaction through intuitive hand gestures.

B. Methodological Innovations

Dimensionality Reduction Techniques:
Principal Component Analysis (PCA) is highlighted in [7] as a method to effectively manage high-dimensional accelerometer data, optimizing real-time gesture recognition processes. **Efficiency in Sensing Techniques:**
According to [5], effective analysis of single-sensor data can significantly determine device orientation and movement, eliminating the need for multiple sensors and highlighting advancements in sensor data processing.

C. Applications in Real-World Settings

Consumer Electronics:
Papers [3] and [1] explore the application of gesture recognition in mobile devices and wearables, respectively, emphasizing the importance of computational efficiency and integration of multimodal data for enhanced user interaction. **Enhancing User Experience:**
These technologies facilitate more natural user interactions, reducing reliance on traditional input methods and enhancing accessibility, particularly in dynamic environments such as driving [3], [1].

D. Comparative Studies and Evaluations

Model Comparisons:
Various studies, including [6], demonstrate the effectiveness of different gesture recognition models. Innovations like the PairNet model modifications have shown improved efficiency over traditional CNNs and RNNs.
Technological Assessments:
The evaluations reveal both strengths, such as increased intuitiveness in user interfaces [8], and limitations, including hardware dependency and gesture diversity [6], [8].

E. Synthesis of Findings

The literature reviewed highlights significant technological and methodological advancements in gesture recognition, emphasizing the evolution toward more integrated and user-friendly systems. Innovations in neural networks and sensor technologies are enhancing consumer electronics' interactivity, making interfaces more intuitive and engaging. Research Gaps and Future Directions Challenges remain in expanding gesture diversity, achieving cross-device generalization, and integrating multimodal data. Future research should focus on enhancing gesture recognition capabilities and exploring energy-efficient solutions for broader application.

F. Summary

The review underlines how technological advancements in gesture recognition are transforming interactions with digital devices, making them more intuitive and efficient. Innovations in neural network architectures and sensor applications are paving the way for broader and more effective integration into consumer electronics. Implications of the Review Developers and researchers are encouraged to focus on creating adaptable, scalable systems that can function across various devices and environments. End-users can expect enhancements in device usability that make daily interactions more natural and engaging. As technology progresses, addressing privacy, security, and energy efficiency will become increasingly important to fully realize gesture recognition technologies' potential in improving human-computer interaction.

III. Methodology

A. Data Collection Procedure

The aim of this project was to collect gesture data using the Phyphox smartphone application Phyphox. Four primary gesture classes were defined for this study:

- Moving the phone in a circle
- Waving
- Gesturing "come here"
- Gesturing "go away"

Additionally, a general movement artifact category was considered. Data was recorded using the "Acceleration (without g)" feature in the Phyphox app. Each gesture was performed continuously for 15 repetitions without stopping. Approximately 30 datasets for each gesture were collected, accounting for various scenarios such as walking while performing the gesture. Each individual exhibited unique gesture styles, enhancing the variability and robustness of the dataset. The collected data were subsequently mixed to create a comprehensive dataset.
After visualizing the raw data, it was evident that trimming and smoothing were required to enhance the quality. Therefore, data preprocessing was initiated.

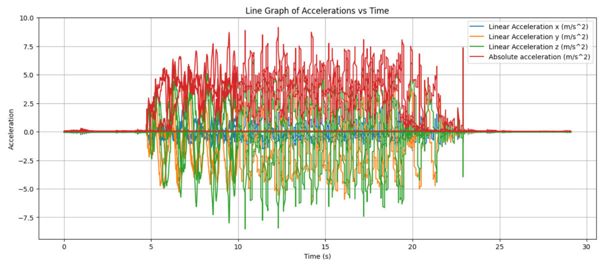

Fig. 1: Visual representation of Raw data

B. Data Pre-processing and Feature Extraction

The preprocessing steps included filtering, trimming, and window segmentation to remove noise and isolate meaningful patterns. A Butterworth low-pass filter was applied to eliminate high-frequency noise, and each gesture recording was segmented into overlapping windows for further analysis.
Feature extraction was then conducted, utilizing time-domain and frequency-domain features, as well as wavelet decomposition, to capture essential gesture characteristics. This thorough preprocessing ensured a high-quality dataset, paving the way for accurate and robust gesture classification.

- **Butterworth Filter Implementation:**
  A Butterworth low-pass filter was applied to remove high-frequency noise from the accelerometer data. The filter was designed using the SciPy library with a customizable cutoff frequency and order. Filter coefficients were calculated to ensure minimal phase distortion.
- **Data Trimming and Filtering:**
  Next, gesture data was trimmed and filtered to isolate meaningful movements. Thresholds for each acceleration axis were computed based on the mean and standard deviation to detect significant gestures. The start and end indices of meaningful gestures were

determined, and the filtered results were added as new columns.

- **Sliding Window Segmentation:**
  The filtered data was then segmented into fixed-size overlapping windows to analyze gesture patterns comprehensively. The window size and step size were configured to ensure meaningful yet manageable data segments.
- **Wavelet Feature Extraction:**
  Statistical features were extracted using discrete wavelet decomposition with the Daubechies 4 wavelet. For each level, the mean and standard deviation were calculated, capturing essential temporal patterns.

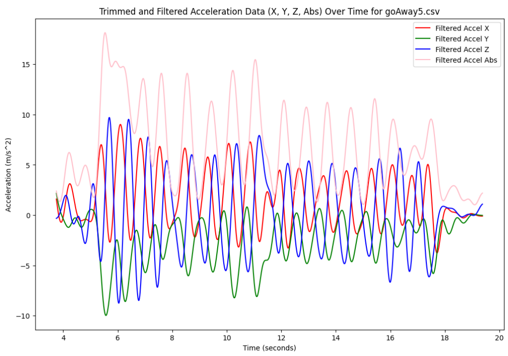

Fig. 2: Gesture Pattern for Go Away Gesture

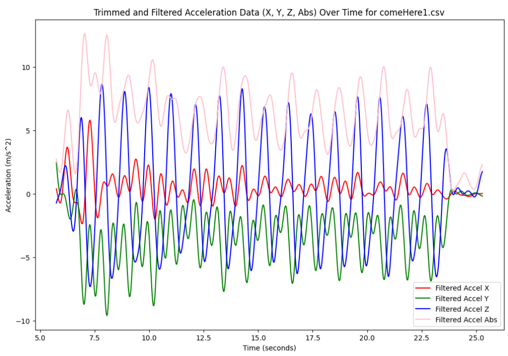

Fig. 3: Gesture Pattern for Come here Gesture

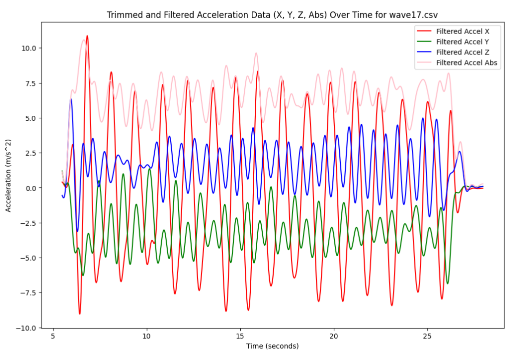

Fig. 4: Gesture Pattern for Wave Gesture

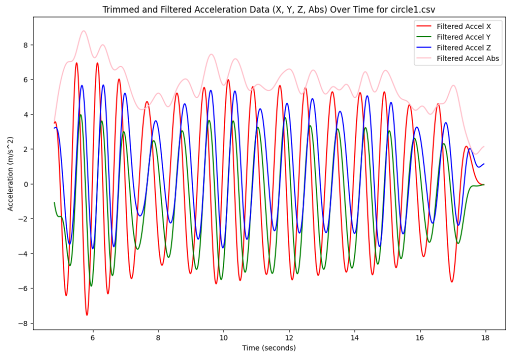

Fig. 5: Gesture Pattern for Circle Gesture

Filtered acceleration data for each axis (x, y, z, and absolute) was plotted over time, enabling visual inspection of gesture patterns and ensuring data quality.

C. Feature Extraction

After obtaining smoothened, filtered, and trimmed data,extraction of the best features for gesture recognition models was carried out.
Following data cleaning and filtering, the team proceeded to extract features and visualize accelerometer data for gesture recognition. This step aimed to uncover meaningful insights and enhance analysis by identifying relevant features within the dataset.

Feature Calculation:
A comprehensive set of features was extracted from the accelerometer data segmented into windows. For each window, statistical features (mean, standard deviation, min, max, skewness, kurtosis) and signal features (zero-crossing rate, FFT peak frequency and magnitude, spectral energy) were computed for each axis (x, y, z).

SMA and SVM:
Signal Magnitude Area (SMA) and Signal Vector Magnitude (SVM) were calculated to capture the overall magnitude of acceleration signals.

Wavelet Features:
Discrete wavelet decomposition with the Daubechies 4 wavelet was used to extract additional temporal features, with the mean and standard deviation calculated for each level, capturing essential temporal patterns.

The extracted features from all windows were collected into a DataFrame for further analysis, providing a robust foundation for developing accurate gesture recognition models.

D. Normalization and standardization

After the feature extraction, the team worked on data normalization and scaling to ensure consistent feature scaling across the dataset. This step was crucial to eliminate biases arising from varying data ranges.

Normalization and Scaling:
The team implemented a comprehensive data processing pipeline to enhance gesture recognition performance. First, normalization and standardization were applied to ensure consistent feature scaling across the dataset.

Dimensionality Reduction and Feature Selection:
Next, Principal Component Analysis (PCA) and t-SNE were used for dimensionality reduction, enabling visualization and identification of underlying gesture patterns. Clustering visualizations using these techniques helped assess data separability, guiding the model training process.

Cluster Visualization and Evaluation:
Cluster visualization using PCA and t-SNE provides insights into the effectiveness of the models and highlights areas requiring further refinement.

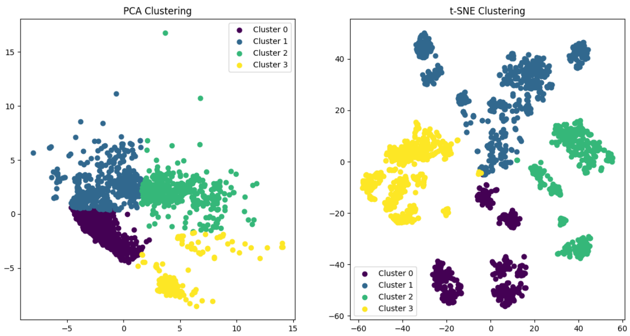

Fig. 6: PCA and t-SNE Clustering

Ultimately, this approach aimed to ensure the development of a highly accurate and generalizable gesture recognition system by carefully selecting features, balancing the dataset, and employing advanced machine learning techniques.

E. Balancing the Dataset

To address potential class imbalance and enhance model performance, the Synthetic Minority Oversampling Technique (SMOTE) is employed. This technique generates synthetic samples for underrepresented gestures, resulting in a balanced dataset suitable for training.

F. Feature Selection

After performing clustering with PCA and t-SNE, the team assessed whether the gestures were well-separated. To further enhance model performance, Recursive Feature Elimination (RFE) was used to identify and retain the most important features. By focusing on the most relevant data points, the model aimed to achieve optimal performance while reducing noise.
The cross-validation score increased sharply up to around four features and then plateaued or increased only slightly with more features. This suggests that increasing the number of features beyond four does not significantly improve the model's performance. The elbow appears to be at around four features, indicating that these features capture most of the information necessary to perform well on the gesture classification task, as can be inferred from the figure below.

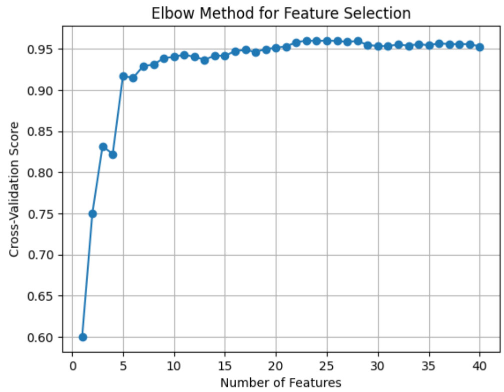

Fig. 7: Elbow method for feature selection

Ultimately, this approach aimed to ensure the development of a highly accurate and generalizable gesture recognition system by carefully selecting features, balancing the dataset, and employing advanced machine learning techniques. After performing all the data preprocessing steps, the team identified the most necessary features and began building the machine learning models for gesture classification.

G. Model Training and Evaluation

A range of machine learning models, including logistic regression, Support Vector Classifier (SVC), gradient boosting, random forests, and AdaBoost classifiers, were trained and evaluated using cross-validation. Grid search optimization identified the best hyperparameters for each model, ensuring optimal accuracy.
The results are discussed in the following sections.

IV. Results from Each Stage

Initially, nine datasets per gesture were collected, but due to inadequate performance and signs of overfitting, the dataset was expanded to 30 sets per gesture. Moreover, to enhance dataset variability, data collection scenarios were diversified, including sitting, standing, and walking while performing gestures.
Data preprocessing was critical for enhancing data quality. Filtering, trimming, and window segmentation techniques were applied to remove noise and isolate meaningful patterns. Butterworth low-pass filtering effectively eliminated high-frequency noise, while threshold-based trimming isolated significant gestures. Sliding window segmentation ensured comprehensive analysis of gesture patterns.
Feature extraction encompassed time-domain, frequency-domain, and wavelet decomposition techniques to capture essential gesture characteristics. Statistical features like mean, standard deviation, and signal features such as zero-crossing rate were computed for each axis. Wavelet decomposition with Daubechies 4 wavelet provided additional temporal insights.
Normalization and standardization were applied to ensure consistent feature scaling across the dataset, while dimensionality reduction techniques like PCA and t-SNE aided in visualizing and identifying underlying gesture patterns.
Synthetic Minority Oversampling Technique (SMOTE) addressed class imbalance, further enhancing model performance.

Recursive Feature Elimination (RFE) identified and retained the most important features, reducing noise and improving model performance. The elbow method determined that around four features captured most relevant information for gesture classification.
Machine learning models including Logistic Regression, RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier, and Support Vector Classifier (SVC) were trained and evaluated using cross-validation. Grid search optimization identified optimal hyperparameters for each model.
Results indicated varying performance across models, with **GradientBoostingClassifier** exhibiting the highest accuracy (test: **0.928, training: 0.967)** and RandomForestClassifier closely following (test: 0.906, training: 0.928). Logistic Regression and SVC also achieved competitive results (test accuracy around 0.86). However, AdaBoostClassifier performed relatively lower (test accuracy: 0.714), suggesting potential for further optimization.
Throughout the project, meticulous attention was paid to data quality, model performance, and mitigating overfitting risks. Diversified data collection scenarios, extensive preprocessing, feature selection methods, and model evaluation techniques ensured the development of a highly accurate and generalizable gesture recognition system.

V. Conclusion and future Recommendations

In this study, a sophisticated gesture recognition system was developed using data collected through the Phyphox smartphone application. Through rigorous experimentation and evaluation, the **GradientBoostingClassifier** emerged as the most effective model, achieving a **high test accuracy of 0.928** and a weighted **F1-score of 0.93**. This performance underscored the classifier's robustness in capturing the nuances of gesture-based interactions. The Random Forest Classifier also performed admirably, demonstrating strong capabilities in classifying gestures with a test accuracy of 0.906 and a weighted F1-score of 0.90. Meanwhile, the Support Vector Classifier (SVC) showed balanced performance, proving to be a reliable option for gesture recognition tasks with minimal overfitting.
The strength of the methodology lay in its comprehensive approach to data preprocessing and feature extraction. The use of a Butterworth low-pass filter, data trimming based on statistical thresholds, and the segmentation into overlapping windows facilitated a robust platform for feature extraction. These pre-processing steps ensured that the features fed into the machine learning models were reflective of the intrinsic patterns of the gestures, enhancing model performance.

However, certain limitations were noted. The AdaBoostClassifier did not perform as well as other models, suggesting it may not be as suitable for this specific type of data. This discrepancy highlighted the need for continued exploration of different modeling techniques and their compatibility with complex gesture recognition tasks.
Future Recommendations include:
Advanced Feature Engineering:
There is potential to explore more sophisticated time-frequency analysis techniques to further enhance model accuracy. This could involve integrating additional signal processing methods or exploring new feature extraction algorithms that could better capture the complexity of human gestures.

Deep Learning Models:
The employment of deep learning frameworks such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks could significantly improve the recognition of complex gesture patterns by leveraging their ability to extract spatial and temporal dependencies.

Real-Time Implementation:
The development of models that can operate in real time on mobile devices would make gesture recognition technology more applicable in practical scenarios, such as in interactive applications or for accessibility enhancements.

Multimodal Data Integration:
The integration of data from additional sensors, such as gyroscopes and magnetometers, could enrich the models' input and improve accuracy, providing a fuller picture of the gesture dynamics.

Dynamic Gesture Recognition Systems:
Future research could explore the integration of Convolutional Neural Networks (CNNs) with real-time processing capabilities to develop dynamic gesture recognition systems that adjust and learn from continuous user interaction.

Collecting Data from Smart Devices:
Making data collection more accessible from smartwatches or smart appliances would significantly expand the available dataset, and new gestures can be incorporated to improve the training and recognition process.

Expanding the Gesture Vocabulary:
Adding more gestures for the model to train and recognize would lead to a more comprehensive and versatile gesture recognition system, adaptable to a broader range of applications.

Ultimately, these strategies will contribute to building an accurate, robust, and user-friendly gesture recognition system suitable for a wide range of real-world applications.

VI. Discussion

The comparative analysis of various machine learning models for gesture classification yielded insightful results. Gradient Boosting emerged as the best performer, achieving a test accuracy of 91.%, due to its high training accuracy of 95.1%. It excels in consistently recognizing all gesture classes with high precision and recall, particularly for class 3 (f1-score: 0.98).
RandomForestClassifier, which attained a test accuracy of 88.8%, demonstrated balanced performance across all classes but showed slight overfitting with a training accuracy of 90.%. Support Vector Classifier (SVC) followed

closely with a test accuracy of 85.5%, offering robust classification performance for most classes, although it struggled with gesture 4 (recall of 0.57).
Logistic Regression, despite being a competitive baseline model with a test accuracy of 83.3%, showed balanced classification across classes but had difficulty with gesture 4 (precision of 0.73). Meanwhile, AdaBoostClassifier, achieving a test accuracy of 76.7 lagged behind due to its challenges in accurately recognizing gesture classes 2 and 4.

In summary, **Gradient Boosting** stands out as the leading model for accurate and consistent gesture classification, delivering superior performance across all classes. While RandomForestClassifier and SVC offer reliable alternatives with balanced performance and manageable overfitting, Logistic Regression and AdaBoostClassifier, though competitive, exhibit noticeable weaknesses in specific gesture classes.

Model	Best Hyperparameters	Cross-Validation Score	Training Accuracy	Test Accuracy	Precision (Weighted Avg)	Recall (Weighted Avg)	F1-Score (Weighted Avg)
Logistic Regression	{'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}	0.868	0.87	0.858	0.86	0.86	0.85
Random Forest Classifier	{'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 100}	0.908	0.928	0.906	0.91	0.91	0.9
Gradient Boosting Classifier	{'learning_rate': 0.01, 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}	0.927	0.967	0.928	0.93	0.93	0.93
AdaBoost Classifier	{'learning_rate': 0.05, 'n_estimators': 50}	0.739	0.753	0.714	0.75	0.71	0.71
Support Vector Classifier	{'C': 0.1, 'kernel': 'linear'}	0.877	0.88	0.865	0.87	0.86	0.86

Fig. 8: Tabular Summary for the models

REFERENCES

[1] N. Siddiqui and R. H. M. Chan, "Multimodal hand gesture recognition using single IMU and acoustic measurements at wrist," PLOS ONE, 13-Jan-2020. [Online]. Available:https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0227039. [Accessed: 15-Apr-2024].

[2] S. Zhao, H. Cai, W. Li, Y. Liu, and C. Liu, "Hand gesture recognition on a resource-limited interactive wristband," MDPI, 25-Aug-2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/17/5713. [Accessed: 15-Apr-2024]

[3] F. Hong, S. You, M. Wei, Y. Zhang, and Z. Guo, "MGRA: Motion gesture recognition via accelerometer," MDPI, 13-Apr-2016. [Online]. Available: https://www.mdpi.com/1424-8220/16/4/530. [Accessed: 15-Apr-2024]

[4] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on Computer Vision: A review of techniques," MDPI, 23-Jul-2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/8/73. [Accessed: 15-Apr-2024]

[5] O. D. Incel, "Analysis of movement, orientation and rotation-based sensing for phone placement recognition," MDPI, 05-Oct-2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/10/25474. [Accessed: 17-Apr-2024]

[6] Y.-C. Chu, Y.-J. Jhang, T.-M. Tai, and W.-J. Hwang, "Recognition of hand gesture sequences by accelerometers and gyroscopes," MDPI, 18-Sep-2020. [Online]. Available:https://www.mdpi.com/2076-3417/10/18/6507. [Accessed: 17-Apr-2024]

[7] T. Marasović and V. Papić , "Accelerometer-based gesture classification using Principal Component Analysis" IEEE xplore, 17-Sep-2011. [Online]. Available: https://ieeexplore.ieee.org/document/6064356/. [Accessed: 17-Apr-2024]

[8] J. Ducloux, P. Colla, P. Petrashin, W. Lancioni, and L. Toledo, "Accelerometer-based hand gesture recognition system for interaction in digital" *IEEE Xplore*, 15-May-2014. [Online]. Available: https://ieeexplore.ieee.org/document/6861003. [Accessed: 17-Apr-2024]