Hand activity recognition using wearable sensors

Sandy Tam, Nathan Dinh, Angie Do, Hannah Tran, and Emily Liu

Keywords: time-series analysis, human activity recognition (HAR), wearable sensors, machine learning classification, hand activity recognition.

Abstract. In this document there are the main guidelines for preparing your contribution for the electronic proceedings of meeting. This document will help you to produce the PDF file of your paper. The paper length must be from 4 to 6 pages. The language used should consistently follow one of the English variants (Canadian English, British English, US American English, Australian English, etc.).

1 Introduction

1.1 *Background and Motivation*

Human Activity Recognition (HAR) has become a crucial part of modern wearable sensing technology, allowing systems to interpret human behavior using smartphone and smartwatch sensors. While many HAR studies focus on large, broad movements such as walking or sitting, there are few that focus on the fine-grained hand-oriented motions. These subtle movements of the hand are highly relevant for applications in digital health, dietary monitoring, rehabilitation, and habit tracking.

The WISDM-51 dataset provides an opportunity to study these detailed hand activities. The dataset includes both smartphone and smartwatch accelerometer and gyroscope sensors collected at 20 Hz from 51 subjects across 18 distinctive activities. The smartwatch is worn on the dominant wrist, capturing the nuanced rotations and movement of the hand during varying activities.

1.2 *Project Goals*

The primary goal for this project is to analyze and prepare the WISDM-51 dataset for fine-grained hand activity recognition. This includes:

- determining the structure and quality of the dataset
- identifying characteristic motion signatures for hand-based events
- extracting important features from the data
- training models to differentiate hand-based events

2 Literature Review

[1] "Smartwatch-based Activity Recognition: A Machine Learning Approach."

Smartwatches and smartphones both contain motion sensors that enable activity recognition, but research shows that smartwatches excel at identifying hand-based activities such as eating, achieving 93.3% accuracy for drinking detection compared to a smartphones' 77.3%. While activity recognition research has shifted from specialized body-worn devices to smartphones in recent years, smartphones still face limitations due to the inconsistent pocket placement

Article version: December 5, 2025 16:38 PST

and orientation, particularly for women who often do not carry their phones in their pockets. Smartwatches overcome these challenges through consistent wrist positioning, which is ideal for tracking hand movements. Since smartwatches also pair with smartphones, both devices' sensors can be combined for enhanced activity recognition in biomedical and health applications.

[2] "Activity recognition from user-annotated acceleration data."

Researchers developed algorithms to detect 20 physical activities using five small biaxial accelerometers worn simultaneously on different body parts, achieving 84% accuracy with decision tree classifications that analyzed the features. The study collected unsupervised data from 20 subjects performing everyday tasks in semi-natural conditions, with subjects labeling their own activities rather than researchers observing them. Results showed that while multiple accelerometers improved recognition through conjunctions in acceleration features, performance dropped only slightly when using just two sensors (thigh and wrist), and some activities required subject-specific training data rather than subject-independent models. This work addressed limitations in prior activity recognition research, which usually relied on constrained lab settings, limited dataset from single individuals, or focused on fewer than nine activities mostly involving ambulatory motions and basic postures.

[3] "A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors."

This tutorial provides a complete introduction to human activity recognition for newcomers, particularly focusing on recognition using on-body inertial sensors and addressing the challenges that have emerged as the field has matured over the past 20 years. The authors introduce the Activity Recognition Chain (ARC) as a general-purpose framework for designing and evaluating activity recognition systems, discussing both challenges shared with general pattern recognition and thos unique to human activity recognition. The tutorial details each component of the ARC framework while referencing related research and best practices developed by the activity recognition community. To show practical application, the authors present an educational example of recognizing hand gestures from inertial sensors on the upper and lower arm, demonstrating how different implementations of framework components compare and impact overall recognition performance.

[4] "A new method for measuring meal intake in humans via automated wrist motion tracking."

This paper presents a new method for measuring energy intake using a watch-like device with a gyroscope that automatically tracks wrist motion to detect and count bites of food, addressing the difficulty of accurate calorie measurement in daily living where traditional methods achieve only 60-80% accuracy compared to laboratory tools at 95%. Testing showed the device achieved 94% sensitivity in controlled meal settings and 86% in uncontrolled settings, with one false positive per five bites in both conditions, and preliminary data indicated a positive relationship between measured bites and caloric intake. The research is motivated by the obesity epidemic and the need for better intake measurement tools, as current methods like calories labels and serving size estimates commonly produce errors of 50% or more when individuals attempt to count calories on their own .While further research is needed to validate the relationship between bites taken and kilocalories consumed, the experiments across different subjects and food varieties demonstrate the method's promising potential as an automated measure of energy intake for daily use.

[5] "A Review of Hand Function Rehabilitation Systems Based on Hand Motion Recognition Devices and Artificial Intelligence."

The article examines modern hand rehabilitation systems for stroke patients, analyzing both computer vision-based approaches that use cameras and virtual reality games for comfortable, equipment-free training, and wearable sensor systems that provide high-precision motion tracking for fine motor skills despite higher costs and comfort issues. Artificial intelligence methods such as convolutional neural networks and support vector machines significantly enhance recognition accuracy and enable personalized, adaptive rehabilitation through gesture recognition and robot control. The authors conclude that future systems must address limitations including bulky hardware, unstable signals, and insufficient clinical validation while improving human-robot integration and patient comfort to optimize recovery outcomes.

[6] "Deep Neural Networks for Time Series Classification in Human Activity Recognition."

The article explores deep learning models for classifying human activities from mobile sensor time series data, addressing limitations in previous research that used restricted activity sets, single sensors, and subject-dependent splits that leaked test information into training. The authors trained four architectures (bidirectional LSTM, CNN, CNN-LSTM, and ConvLSTM) on the WISDM dataset containing accelerometer and gyroscope data from fifty-one participants across thirteen hand-oriented and non-hand-oriented activities, using subject-independent splits and sliding windows of eighty time steps. Results show all models perform better on accelerometer than gyroscope data, smartwatch sensors capture more distinctive patterns than phones, and CNN-LSTM consistently achieves the highest accuracy at eighty-six percent while effectively distinguishing hand motions despite occasional confusion between similar activities like dribbling and playing catch. The study concludes that smartwatch accelerometer data provides the most informative signals and CNN-LSTM is the most promising architecture for real-world activity recognition with minimal preprocessing.

[7] "Machine-learned wearable sensors for real-time hand-motion recognition: toward practical applications."

The article examines how machine learning, soft electronics, and wearable electromechanical sensors enable accurate hand-gesture recognition for human-machine interfaces, overcoming earlier challenges where complex hand motions produced noisy signals that were difficult to interpret reliably. Modern artificial intelligence algorithms can now extract meaningful features from massive datasets and recognize nuanced gestures even with limited training data, providing rapid real-time feedback suitable for wearable technologies by mimicking the human proprioception system that enables precise motor control. Practical applications include EMG-based systems using XGBoost to wirelessly control robotic cars and graphene-based electronics with CNNs achieving over 99% accuracy for real-time robotic hand control across six gesture classes. The authors conclude that combining innovative materials with advanced machine learning creates highly responsive and intelligent gesture-recognition systems with enormous potential for healthcare, robotics, wearable computing, and assistive devices.

[8] "A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems."

The article provides a comprehensive review of hand gesture recognition techniques, dividing approaches into non-vision methods using wearable sensors that are stable in difficult environments but uncomfortable, and vision-based methods using cameras that are more natural and convenient for applications in robotics, virtual reality, and smart devices. The paper examines three key stages: detection using skin-color modeling and shape segmentation, tracking through optical flow and Kalman filters, and recognition comparing traditional machine learning (Naive Bayes, Hidden Markov Models) with modern deep learning approaches (CNNs, RNNs) that achieve significantly higher accuracy through automatic feature learning. The authors identify real-world challenges including inconsistent lighting, occlusion, distracting backgrounds, and individual gesture variations, highlighting future research directions such as lightweight real-time neural models, multi-sensor fusion, stronger feature descriptors, and user-adaptive systems.

[9] "Hand Gesture Recognition using Machine Learning."

The article "Hand Gesture Recognition using Machine Learning" presents a comprehensive overview of the field, emphasizing the transition from traditional image processing to advanced Machine Learning (ML) and Deep Learning (DL) techniques required to interpret the complex manual and non-manual components of sign language. In their review of methodologies, the authors discuss a variety of approaches, including Convolutional Neural Networks (CNNs) for image categorization , Multidilated Convolution DenseNet (MDCDN) for automatic feature extraction , and sensor fusion techniques that combine visual data with electromyographic (EMG) signals via neuromorphic processors. Innovative control methods are also highlighted, such as the use of Reinforcement Learning algorithms like Deep Q-learning to classify sensor signals for robotic platform manipulation. The literature relies on diverse datasets, ranging from standard databases like ASL and ISL to the authors' own generated dataset of 54,000 RGB images representing 26 Latin alphabet letters and non-gestures. Results across the domain are promising, with studies demonstrating effective human–computer interaction using minimal hardware and the current study achieving a classification accuracy of 92.3% using a Random Forest model based on MediaPipe landmarks. However, the field still faces significant limitations, particularly the challenge of distinguishing between gesturally similar letters such as 'M' and 'N' or 'S' and 'T' due to subtle variations in finger positioning , as well as the ongoing need to balance high recognition accuracy with the processing speeds required for real-time applications.

[10] "A Review on Sensor-based HAR Models Using GNN: AI in Healthcare."

The article "A Review on Sensor-based HAR Models Using GNN: AI in Healthcare" highlights the transformative potential of Graph Neural Networks (GNNs) in advancing Healthcare 5.0. By modeling wearable sensor data as graphs, GNNs overcome the limitations of traditional machine learning in handling noisy and heterogeneous data, effectively capturing intricate spatial and temporal dependencies. The authors detail a methodology that progresses from acquiring data via IMUs and smartphones to complex preprocessing and feature extraction using architectures like GCNs, GATs, and Spatio-Temporal GNNs. Evaluated against benchmark datasets such as UCI-HAR and PAMAP2, as well as custom clinical datasets, these models demonstrate high performance, with some studies achieving near-perfect accuracy in activity recognition and robust results in fall detection. Despite these successes, the review identifies significant hurdles, including the high computational cost prohibitive for edge devices, limited generalizability across diverse user groups, and a "black box" nature that hinders clinical interpretability. Future research directions are suggested to focus on lightweight, energy-efficient models and explainable AI to ensure these advanced systems can be sustainably integrated into real-world healthcare ecosystems.

[11] "Human activity recognition based on hybrid learning algorithm for wearable sensor data."

To address the challenges of manual feature engineering and capturing long-term dependencies in time-series sensor data, Athota and Sumathi proposed a Hybrid Learning Algorithm (HLA) approach for Human Activity Recognition (HAR). The authors developed two specific frameworks: the Convolution Memory Fusion Algorithm (CMFA), which integrates 1D Convolutional Neural Networks (CNN) with Bidirectional Long Short-Term Memory (Bi-LSTM) layers to extract both spatial and temporal features, and the Convolution Gated Fusion Algorithm (CGFA), which utilizes Bidirectional Gated Recurrent Units (Bi-GRU). The methods were evaluated using the WISDM-HARB dataset, which includes accelerometer and gyroscope data from 51 subjects performing 18 diverse activities, distinguishing between simple locomotion and complex hand-oriented tasks such as eating or folding clothes. Experimental results indicated that the CMFA model achieved superior performance, attaining 97.76% accuracy on smartwatch data and 94.98% on smartphone data, thereby outperforming the CGFA model and existing baselines like DeepConvLSTM. While the study successfully demonstrated that smartwatches are more effective than smartphones for recognizing hand-oriented activities, it also identified limitations, specifically that smartphones struggled with complex hand tasks and that the models occasionally misclassified non-hand-oriented behaviors due to signal similarities.

[12] "Machine learning models for wearable-based human activity recognition: A comparative study."

The comparative study by Ciortuz et al. addresses the gap between Human Activity Recognition (HAR) research and practical healthcare implementation by systematically evaluating nine distinct modeling approaches, ranging from traditional Support Vector Machines (SVM) with hand-crafted features to advanced Deep Learning (DL) architectures including CNNs, LSTMs, GRUs, and Transformers. Utilizing the sensor-rich OPPORTUNITY dataset and the consumer-device-based CogAge dataset, the authors analyzed classification performance across "state" (e.g., walking) and "behavioral" (e.g., drinking) activities, specifically investigating the impact of the "NULL" class and device placement. The results demonstrated that while most models achieved high accuracy for state activities (94.10%–96.48% Average F1-Score), DL models—particularly hybrid LSTM-GRU variants—outperformed SVMs on complex behavioral tasks within the large OPPORTUNITY dataset, whereas the SVM performed best on the smaller CogAge dataset. However, the study identified significant limitations, including the Transformer model's poor performance due to data scarcity, the detrimental effect of class imbalance introduced by the NULL class, and the difficulty models faced in distinguishing reciprocal actions like opening versus closing doors.

[13] "A smartwatch-based framework for real-time and online assessment and mobility monitoring."

This paper focuses on using a smartwatch with the framework ROAMM to perform an assessment on the wearer's movement patterns. The smartwatch platform collects sensor data and processes it, and the framework it uses supports the smartwatch's ability to collect data and analyze it in real time. The study performed in the paper captured the participants' daily activities through the smartwatch sensors. The results show that the framework is a feasible option for capturing movement and activity patterns for participants with a smartwatch. A challenge faced during this study is the limited battery life of the smartwatches.

[14] "Enhanced Hand-Oriented Activity Recognition Based on Smartwatch Sensor Data Using LSTMs."

This paper focuses on using hybrid deep learning model, CNN-LSTM, to perform human activity recognition. It specifically focuses on tracking hand-oriented activities using smartwatch sensor data. The authors built a model that combines Convolutional Neural Networks and Long Short-Term Memory networks. They used a public dataset with sensor data that comes from smartwatch sensors as their data. Their model achieved high accuracy on test data, which demonstrates that deep learning can significantly improve the recognition of hand-oriented activities. A challenge faced during the study relates to the data itself. Hand-oriented activities are often done in controlled settings, which limits generalization.

[15] "Smartwatch Based Activity Recognition Using Active Learning."

This paper focuses on using active learning to perform human activity recognition. The goal of using active learning is to limit the amount of labeled data required for human activity recognition. The authors designed an active learning framework where the system uses a small labeled set and queries the user when met with unlabeled instances. The study uses a dataset with data collected from smartwatch sensors. The active learning approach does reduce the amount of labeled data needed, but it assumes that a user will be available when met with unlabeled instances.

[16] "Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances."
This paper reviews why deep learning is effective when used in human activity recognition through wearable sensors. It surveys deep learning architectures applied to human activity recognition, such as CNN, RNN, LSTM, and hybrid models. The paper covers many publicly available datasets used in human activity recognition, including WISDM and many more. The paper concludes that deep learning based approaches outperform other machine learning approaches for human activity recognition tasks. Hybrid models have been proven to be especially effective. However, the paper emphasizes limitations on generalization and labeling cost when using deep learning approaches for human activity recognition tasks.

[17] "Real-Time Activity Recognition on Smartphones Using Deep Neural Networks."
Human Activity Recognition (HAR) has shifted from hand-engineered features towards deep learning models that learn directly from the raw sensor data. Traditional machine-learning approaches relied on manually engineered statistical and frequency features but these lacked generalizability. This approach also struggled with real-time performance on mobile devices. This study demonstrated that deep neural networks can outperform feature-based methods in terms of classifying common daily activities. It also can run more efficiently on phones which showcase practical real-time recognition. This study demonstrates that CNN, LSTM and hybrid models can capture temporal motion patterns more effectively than conventional pipelines. Overall, HAR systems are improving in terms of automated feature learning and are becoming more robust to sensor variability.

[18] "Smartphone based human activity recognition irrespective of usage behavior using deep learning technique."
With the continuous advancement of technology, advancements in HAR have been made to keep up with different devices, variability in sensor characteristics, or inconsistent data collection. Traditional machine learning approaches struggle in these circumstances because handcrafted features cannot accurately capture complex temporal distortions caused by variability in the devices. Deep learning has shown promise but typically assumes fixed hardware or smooth data collection. This paper addresses these challenges by introducing a CNN based framework that converts inertial time-series data into 2D frequency domain images. This would then allow the model to learn both temporal signatures and inter-axis relationships. The authors also introduce an ensemble of conditional CNN classifiers that are made to operate across different types of devices and inconsistent data collection from phones. This study yielded 94% accuracy even if training and testing devices change. This shows that when deep learning is combined with device agonistic features, HAR is able to improve its robustness and be more practical in real-time uses.

[19] "Human Activity Recognition Using Smartphone Sensors Based on XGBoost Model."
As an alternative to dedicated wearable devices, machine learned based HAR has started to use smartphone inertial sensors to collect data. This allows HAR to be more accessible to the general public as smartphone inertial sensors are low cost and practical in daily life. Traditional HAR pipelines rely on handcrafted statistical and frequency features but studies have shown that models that can capture nonlinear motion patterns and that can handle large heterogeneous

datasets are more in demand. This paper contributes to the need for such models by evaluating a feature engineering driven approach with dimensionality reduction techniques such as PCA and t-SNE. This study applies the XGBoost classifier to smartphone accelerometer and gyroscope data. XGBoost was chosen as it is a model that can handle complex nonlinear relationships while also being robust to noise. The study yields strong performance across everyday and high intensity activities with 97% accuracy in basic recognition and over 92% accuracy in more demanding motion classification tasks. This study highlights the potential to replace traditional motion sensing systems with smartphone sensors.

[20] "Wearable Sensor-Based Human Activity Recognition: Performance and Interpretability of Dynamic Neural Networks."

Research in HAR has shown that deep architectures such as LSTMs and GRUs have shown that it is highly effective at modeling temporal dependencies in inertial sensor data. However, their black-box nature limits transparency in clinical and health monitoring applications. This shows the need for models that balance accuracy, Interpretability, and computational efficiency. This paper evaluates three neural network families: FIRNN, LSTM, and GRU over an extensive experimental design that involves 16,500 trained models. The paper demonstrates that LSTM achieved the highest accuracy of 98.76% over GRU (97.33%) and FIRNN. The paper highlights that FIRNN has an advantage of low resource deployment due to its lower computational complexity. A prominent topic in this paper is the application of LRP which was used to reveal which sensor dimensions and hidden units drive the model's decisions. The analysis shows that the gyroscope Y-axis signals are more important while the accelerometer Y-axis does not drive the model's decision as much. GRU models were found to distribute relevance more diffusely over FIRNN. This paper provides good guidance on selecting neural architectures for wearable HAR systems that need to balance accuracy, efficiency, and interpretability.

3 Methodology

3.1 *Dataset Exploration*

The WISDM-51 dataset is organized into a hierarchical folder structure where sensors are separated by device and sensor type. Each folder contains 51 files, one for each participant, and each record contains the subject ID, activity label, timestamp, and the x, y, and z coordinates. Sensor data was collected at a rate of 20 Hz or about every 50ms. Each participant was asked to perform 18 distinctive tasks for 3 minutes each, with 12 of the tasks being largely hand-based. The hand-oriented activities we considered were: dribbling, playing catch, typing, writing, clapping, brushing teeth, folding clothes, eating pasta, eating soup, eating sandwich, eating chips, and drinking. These parameters are summarized in Table 1.

For this study, each subject had a smartwatch placed on their dominant hand as well as a smartphone in their pocket. We used Python to load the raw watch accelerometer and gyroscope data and check for any null or corrupt values. No missing records were found and all the samples looked sufficient for analysis.

Here is a summary table of our data exploration:

Participants	Average Time	Demographics	Total Raw Samples	Unique Labels
51	61 minutes	Not provided	15,630,426	18

Table 1: **Recap of the results.** Summary of WISDM-51 dataset characteristics including participant count, data collection duration, sample size, and activity labels.

3.2 *Annotation and Signal Exploration*

To understand the physical characteristics of each hand-oriented activity, we extracted time windows around labeled events. The raw time series data for each sensor was divided into 9-second nonoverlapping segments. From there, we took a random 9-second window– about 180 samples at 20 Hz– from a random subject. Then, for each of the selected activities, we visualized the raw x, y, z watch accelerometer values through generated plots. We had in total 12 graphs, but we chose the following to analyze: Figure 1, 2, 3, and 4.

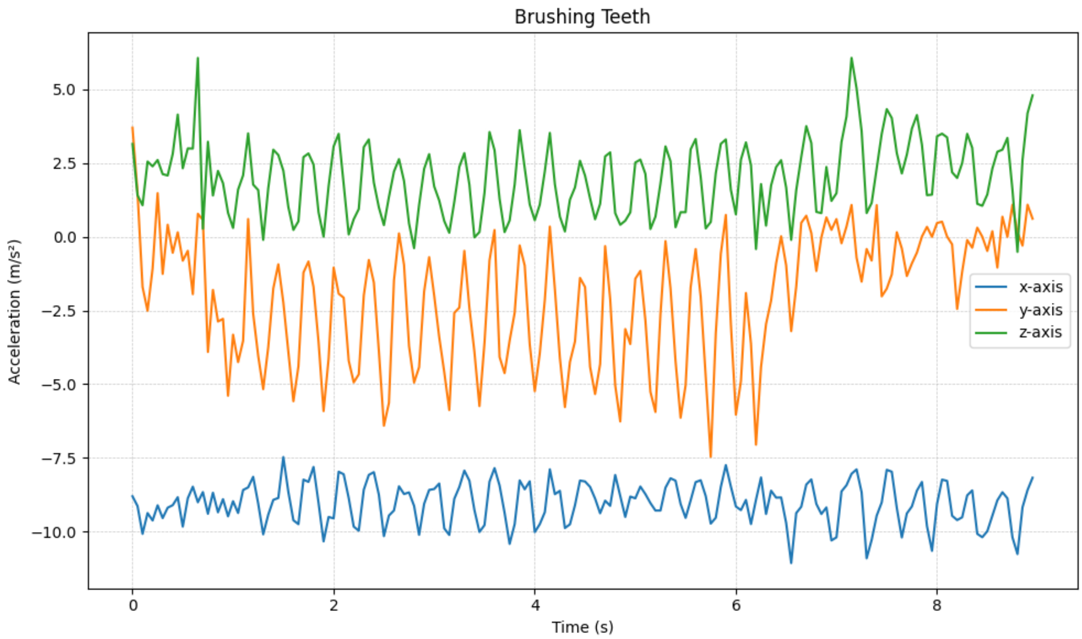

Figure 1: **Raw watch accelerometer signals for brushing teeth.** High-frequency, consistent signals reflect the repetitive back-and-forth brushing motion at a constant tempo with minimal pauses between strokes.

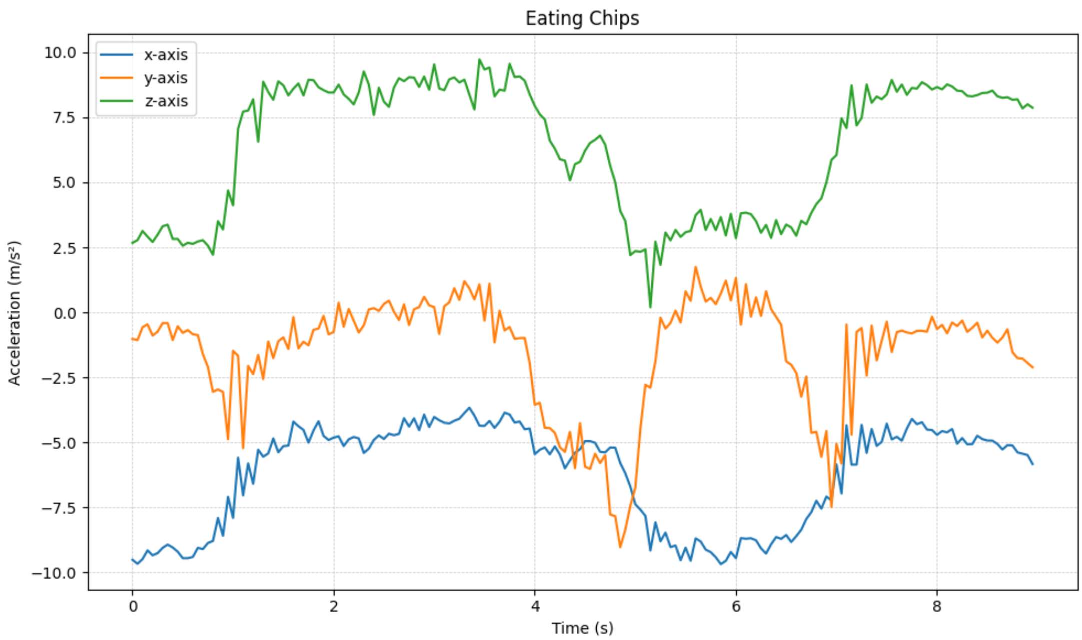

Figure 2: **Raw watch accelerometer signals for eating chips.** The signal shows short bursts of motion separated by pauses, reflecting the hand-to-mouth cycle: steady posture, smooth rotation to grab a chip, and rapid movement to the mouth.

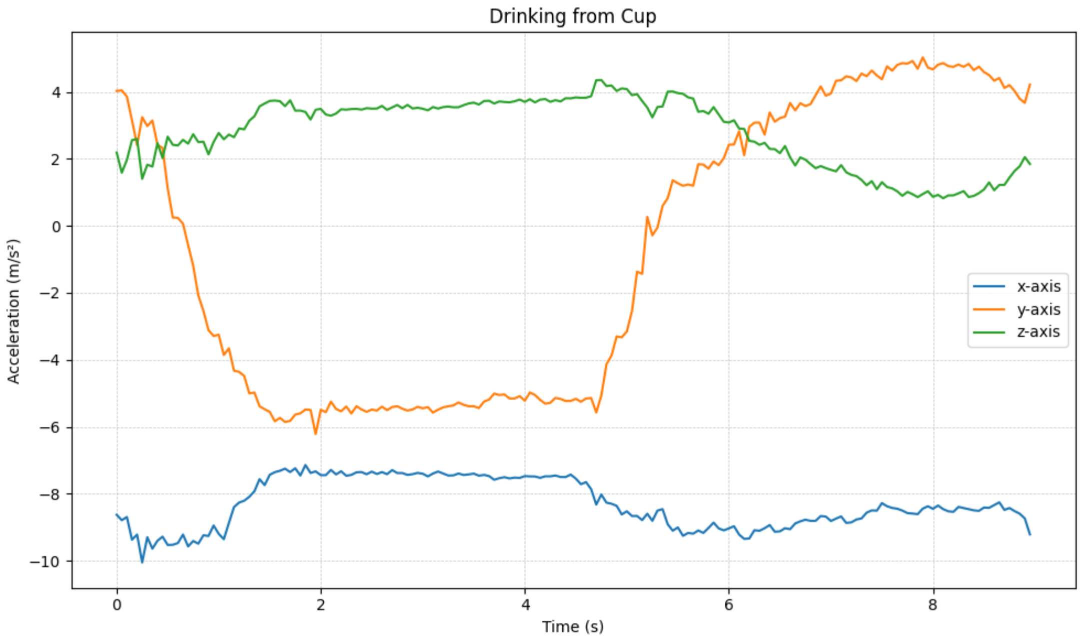

Figure 3: **Raw watch accelerometer signals for drinking.** The signal shows steady, smooth peaks reflecting the continuous hand rotation and gradual wrist orientation changes characteristic of drinking, with less sharp movements compared to eating activities.

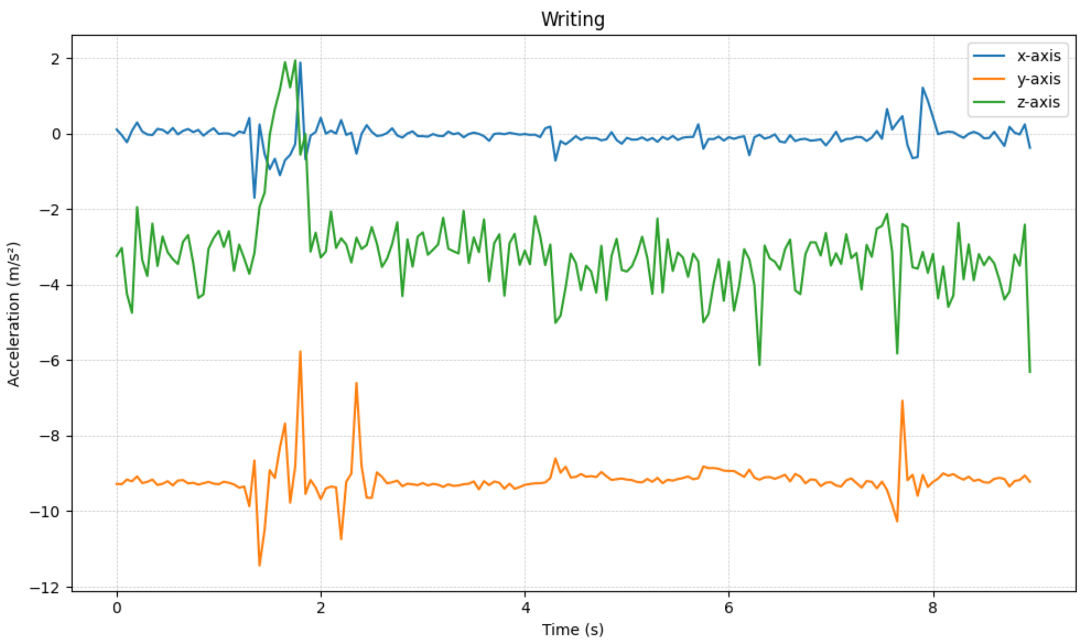

Figure 4: **Raw watch accelerometer signals for writing.** The signal exhibits steadiness with small vibrations, indicating controlled wrist movements while the arm remains mostly stationary, with minimal rotation throughout the activity.

3.3 *Signal Preprocessing*

Signal preprocessing is essential because raw accelerometer data contains sensor noise, vibration artifacts, and missing samples. To denoise the signal, we applied a 4th-order Butterworth low-pass filter with a 5 Hz cutoff. Hand-activity movements occur at relatively low frequencies, while noise typically appears above 5–10 Hz. Filtering reduces high-frequency spikes, producing smoother, more physiologically meaningful trajectories. Missing values were handled using linear interpolation to preserve continuous time-series structure, which is required for windowing and feature extraction. Figure 5 clearly shows the improvement: the filtered signals exhibit smoother patterns and reduce jitter without distorting the underlying motion.

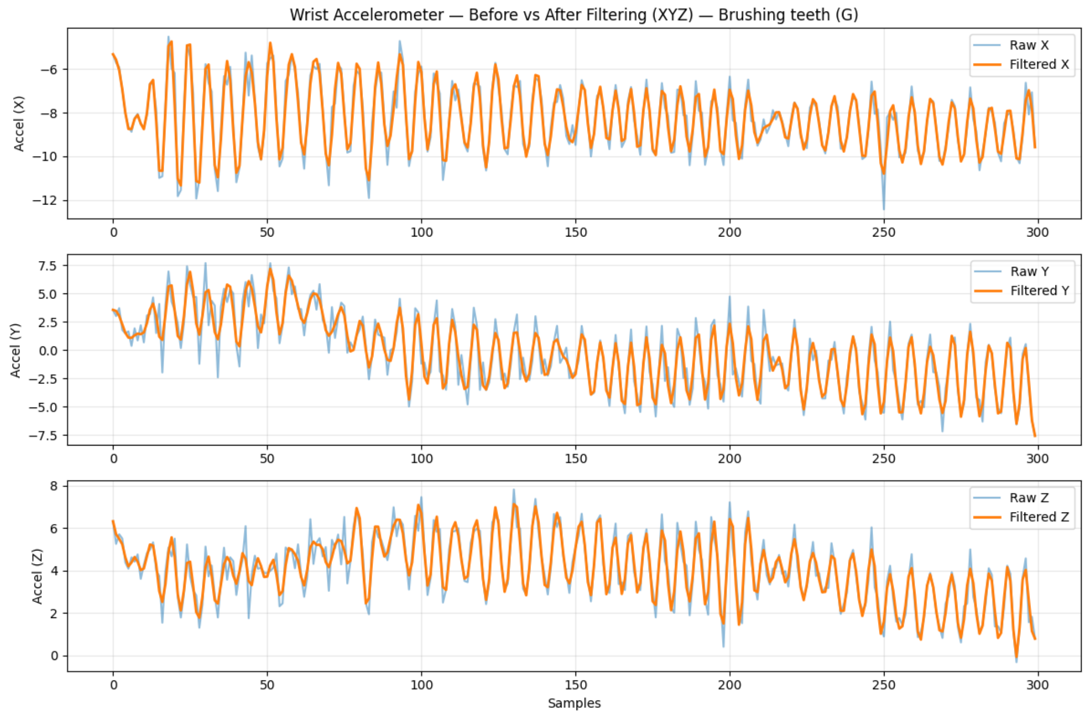

Figure 5: **Filtered signals from brushing teeth activity.** Wrist accelerometer data (X, Y, Z axes) during tooth brushing, showing raw measurements and filtered signals. The filtering effectively reduces high-frequency noise while preserving the underlying motion patterns characteristic of the brushing activity.

3.4 *Windowing Strategies*

```
# Path to one subject's watch accelerometer raw file
RAW_DIR = "wisdm-dataset/raw/watch/accel"
FILENAME = "data_1647_accel_watch.txt" # pick any subject
```

We chose a sliding-window segmentation strategy, which is the standard in human activity recognition. Because accelerometer data does not naturally contain distinct "event boundaries," fixed windows provide consistent temporal structure for feature extraction and machine learning. We tested 10 window sizes ranging from 20 to 200 samples, stepping 20 samples at a time. Given that sampling is 20 Hz, then these correspond to 1 to 10s windows. Smaller windows capture rapid motion transitions, while larger windows provide more stable, aggregated motion features. These windowed segments form the input to the feature extraction and modeling stages of the pipeline.

As seen in Figure 6, smaller windows capture short gestures, create more training examples, but are more noisy and less stable. On the other hand, larger windows provide smoother features, but provide fewer training samples and risk losing fast transient gestures. These windowed segments form the input to the feature extraction and modeling stages of the pipeline.

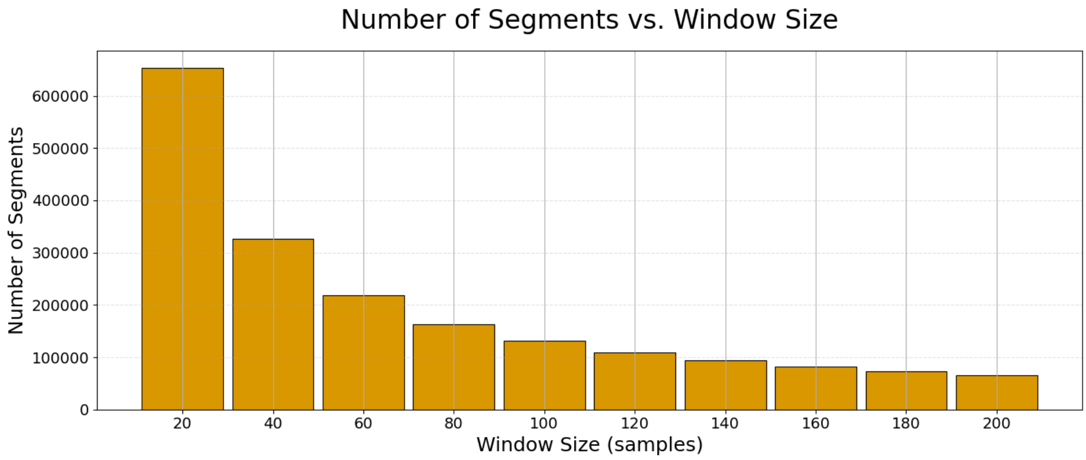

Figure 6: **Number of segments generated as a function of window size.** Smaller window sizes produce significantly more segments, with the count decreasing approximately inversely with window size.

3.5 *Feature Extraction*

3.5.1 Time-Domain Feature Extraction

After segmenting the dataset into 9-second non-overlapping windows (180 samples each), we computed several time-domain features for every sensor channel. These features describe the fundamental behavior of the motion signal within each window:

- **Mean:** Shows the average value of the signal, which reflects the general orientation or position of the wrist during the activity.
- **Standard Deviation (SD):** Measures how much the signal varies. Activities with more intense or inconsistent movements naturally produce larger variation.
- **Root Mean Square (RMS):** Represents the strength or energy of the movement within the window. It is useful for comparing high-intensity versus low-intensity actions.
- **Zero-Crossing Rate (ZCR):** Counts how frequently the signal changes sign. This feature reflects rhythmic or oscillatory patterns, such as the repetitive motion during brushing teeth.

These features give the model a clear statistical summary of wrist movement. They help describe how strong, how stable, and how patterned the motion is.

3.5.2 Frequency-Domain Feature Extraction

Besides examining the raw signals in the time domain, we also analyzed them in the frequency domain using the real Fast Fourier Transform (rFFT). This approach offers a more detailed view of the motion by revealing its periodic or oscillatory qualities. We calculated three key frequency-domain features:

- **Dominant Frequency:** The frequency with the highest energy. Many hand activities have a natural rhythm (brushing teeth, clapping), which this feature captures.

- **Spectral Energy:** Measures the total amount of energy across all frequencies. This is useful for identifying activities with strong or powerful motion.

- **Spectral Entropy:** Measures the unpredictability or complexity of the signal. Higher entropy suggests chaotic or irregular movement, while lower entropy indicates smoother, more repetitive motion.

These frequency-based features capture aspects of the motion that cannot be seen directly from the raw signal, making the dataset more informative and comprehensive.

3.5.3 Feature Matrix Construction and Random Forest Training

Each 9-second window contains 12 channels (watch + phone, accelerometer + gyroscope × 3 axes). For every channel, we extracted all seven features (four time-domain and three frequency-domain) across every sensor axis. This resulted in an 84-dimensional feature vector for each sample. After building the full feature matrix, we trained a Random Forest classifier using an 80/20 train-test split to measure how important each feature is for recognizing different hand activities.

We visualized the results in Figure 7 and Figure 8:

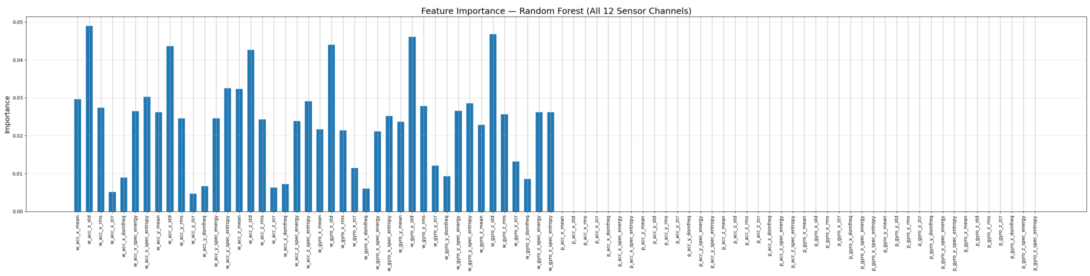

Figure 7: **A full bar chart showing feature importance scores from Random Forest classifier of all extracted features.** Standard deviation features of accelerometer and gyroscope data dominate, with spectral entropy and statistical measures also contributing to classification performance.

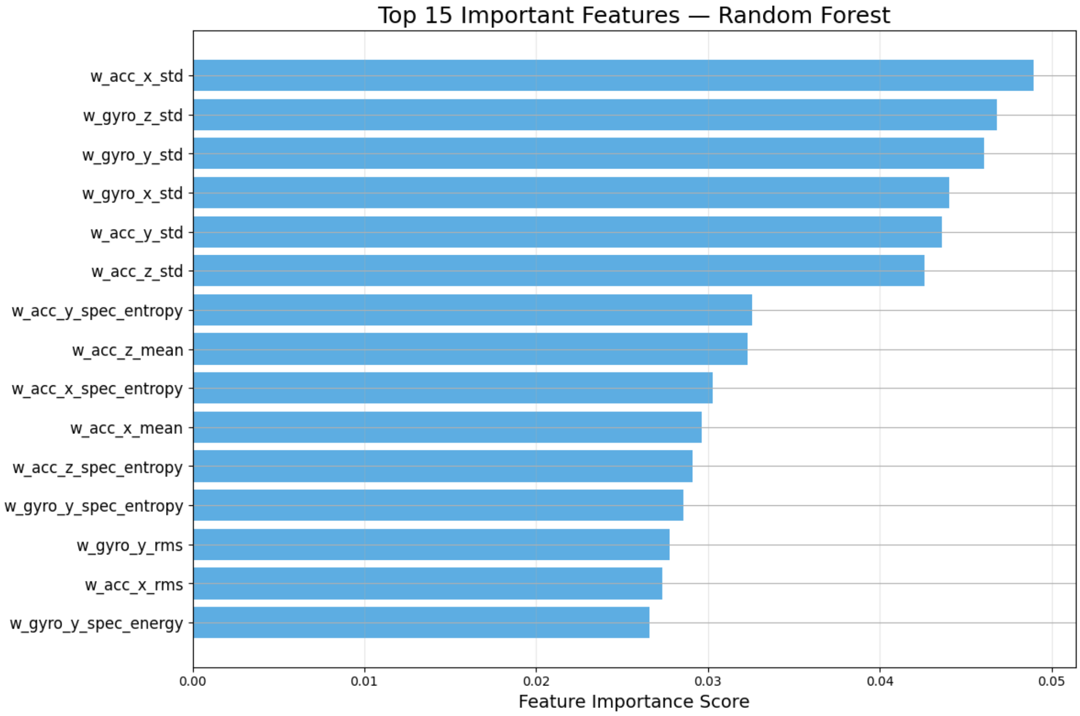

Figure 8: **Top 15 most influential features from Random Forest classifier.** Wrist accelerometer X-axis standard deviation is the most discriminative feature, followed by gyroscope Z and Y standard deviations, indicating that movement variability is key for activity recognition.

This analysis helps us understand which parts of the sensor data truly matter for classification, and which ones contribute very little.

3.5.4 Interpretation of Feature Importance

The analysis clearly shows that watch sensors dominate the top features. This makes sense because the watch is located directly on the wrist, which is the body part that performs almost all hand-related activities. In contrast, the phone tends to stay still, so its motion signal contributes very little.

Across all features, standard deviation (SD) consistently appears at the top for the X, Y, and Z axes of both the watch accelerometer and gyroscope. This means that the amount of variation in wrist movement is the most reliable indicator of activity type.

Frequency-domain features such as spectral entropy also appear high in the ranking, suggesting that the complexity of motion plays a meaningful role in separating activities like writing, eating, or brushing teeth.

Overall, the feature importance analysis confirms that wrist movement provides the most informative signals in this dataset, especially because the watch is positioned directly on the hand and captures subtle variations in motion. Measures of variability, such as standard deviation and RMS, play a crucial role in distinguishing fine-grained hand activities, since each action naturally produces different levels of fluctuation and movement intensity. Frequency-based characteristics, particularly spectral entropy, also contribute meaningful discriminative power by capturing the complexity and rhythm of each activity. In contrast, features from the phone sensors show very limited influence, likely because the phone remains relatively static during most tasks. These findings align with the real-world nature of the activities and reinforce the reliability of the extracted features for accurate activity recognition.

4 Results and Discussion

4.1 *Modeling*

Six classical machine-learning models were evaluated on the WISDM hand-activity dataset using engineered time and frequency-domain features extracted from 9-second (180-sample) sensor windows. An 80/20 train–test split was applied to ensure balanced class representation.

Table 2: Model Performance Leaderboard

Model	**Accuracy**	**Macro-F1**	**Weighted F1**
Random Forest	0.9498	0.9494	0.9497
XGBoost	0.9427	0.9422	0.9425
Decision Tree	0.8938	0.8933	0.8937
SVM (RBF)	0.7445	0.7448	0.7460
AdaBoost	0.3080	0.2931	0.2930
Gaussian Naive Bayes	0.1658	0.1331	0.1349

Random forest achieved the highest performance overall. XGBoost came close under Random forest. Decision Tree and SVM(RBF) achieved moderate accuracy while AdaBoost and Naive Bayes performed the worst out of the six models. This may be due to their sensitivity to noisy sensor features.

The activity labels were labeled as the following codes below:

F: Typing	**G:** Brushing Teeth	**H:** Eating Soup
I: Eating Chips	**J:** Eating Pasta	**K:** Drinking (from glass)
L: Eating Sandwich	**O:** Playing Catch	**P:** Dribbling (basketball)
Q: Writing	**R:** Clapping	**S:** Folding Clothes

Random Forest and XGBoost performed the best as shown in Figure 9. They received F1 scores higher than 0.93. For Random Forest, all activities achieve F1 scores around 0.9-1.0 which confirms that the model is balanced with no dominant and neglected classes. As the complexity of the model used decreases, the F1 score decreases which shows how multivariate sensor data benefit from methods that can capture nonlinear feature interactions and correlations between sensor channels.

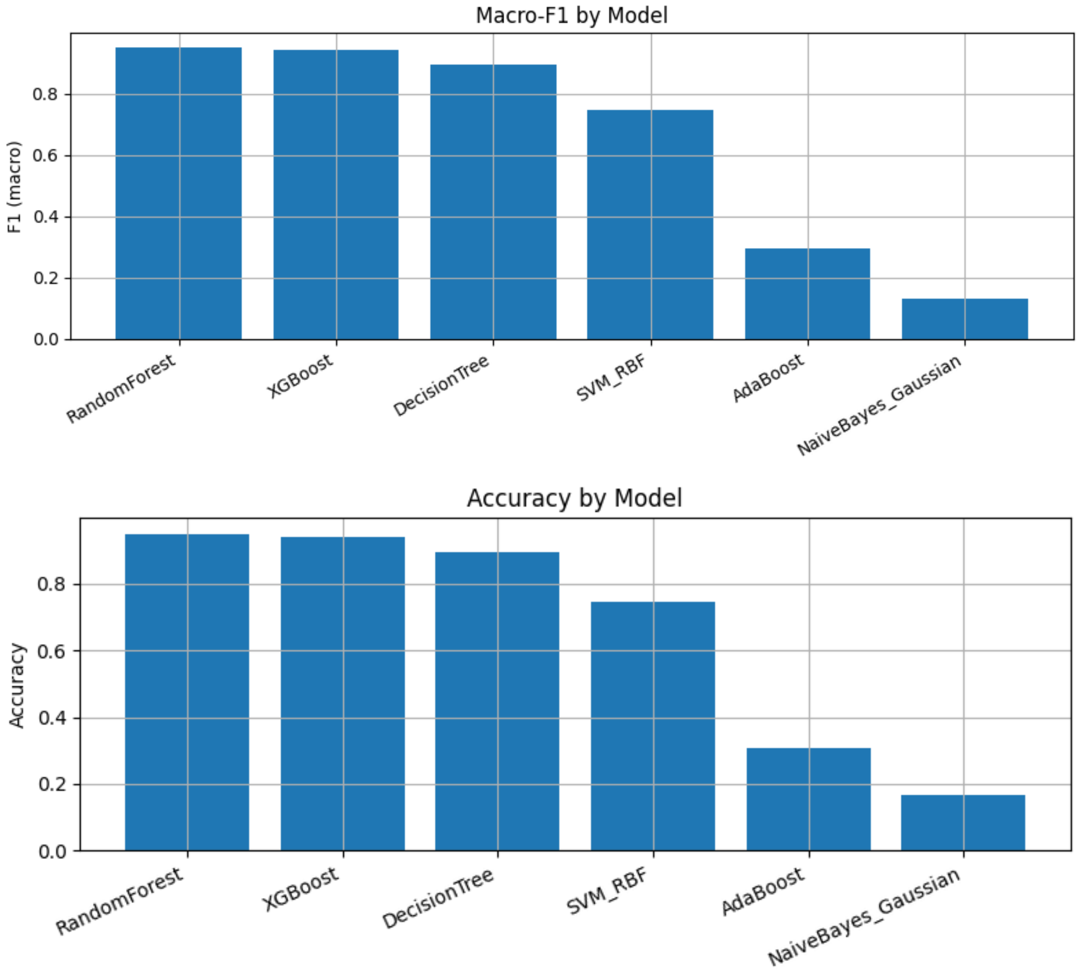

Figure 9: **Accuracy and macro F1-score by model.** Random Forest and XGBoost achieve the highest performance (greater than 94%), while Naive Bayes shows significantly lower scores, indicating poor suitability for this activity recognition task.

The confusion matrix in Figure 10 is a visualization used to evaluate the performance of a classification model by comparing predicted labels with actual labels. The matrix shows a strong dark blue diagonal across all 12 activity classes which indicate consistent correct classifications when Random Forest is used. Minor misclassifications primarily occur between activities which have similar motion-wise actions such as eating vs drinking.

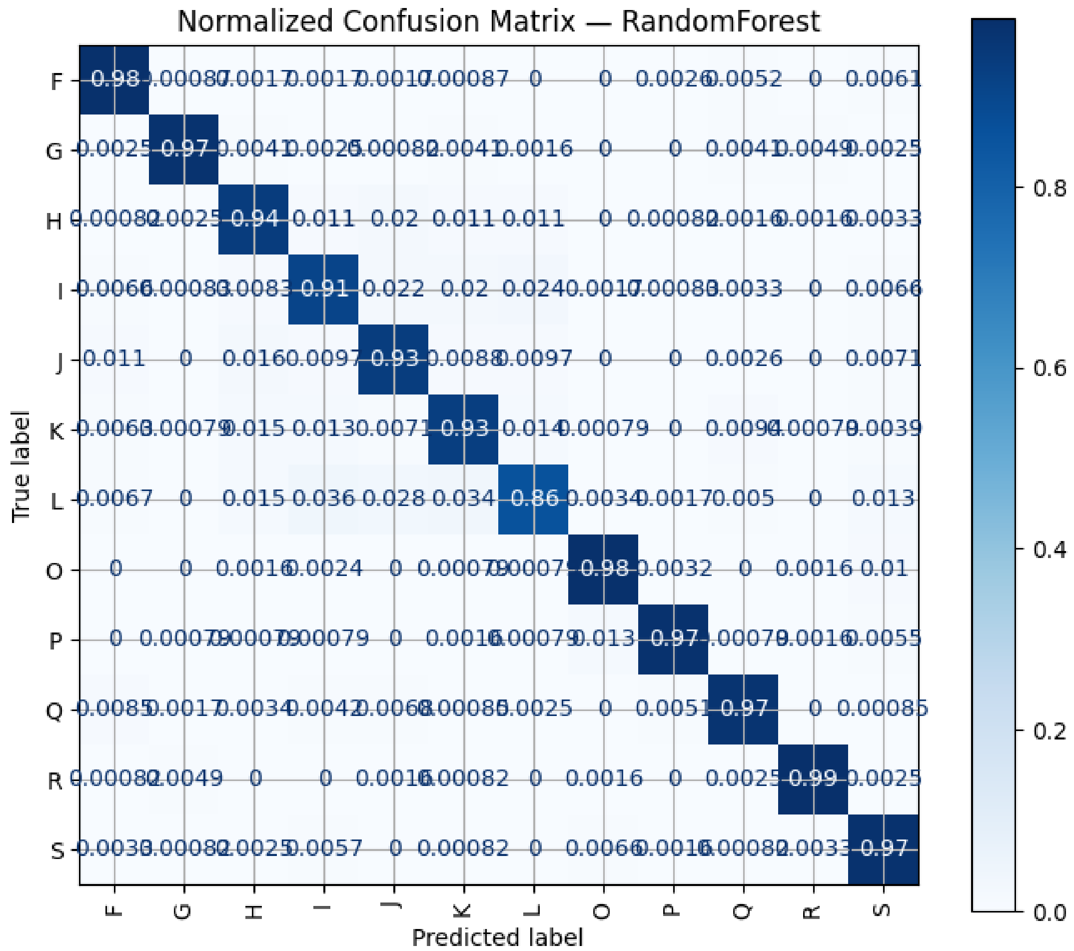

Figure 10: **Normalized confusion matrix for Random Forest classifier.** The model achieves high classification accuracy across all activities, with diagonal values consistently above 0.9, indicating strong discrimination between different activity types. Off-diagonal values are minimal, demonstrating low misclassification rates.

Generally, tree-based models like Random Forest and XGBoost outperform other models due to their ability to model nonlinear feature interactions. Naive Bayes and AdaBoost underperformed as they struggled with high variance sensor features. In all, Random Forest classifier provides the most accurate and reliable recognition of hand orientated activities in the WISDM dataset by achieving an accuracy and F1-score of around 95%.

4.1.1 ANN Implementation and Modeling

We also decided to implement an Artificial Neural Network (ANN) model in conjunction with our classical machine learning models. We did this to compare how different modeling paradigms perform on the WISDM activity-recognition dataset. Typically there are two ways to prepare data for an ANN model:

1) ANN on Raw Time- Series Sequences $\rightarrow$ flattening data:

The first approach involves keeping each activity segment as a sequence of sensor readings with the shape of (window length x number of channels).

In our dataset, this corresponds to:

- T = 180 time steps
- C = 12 sensor channels
- Sequence shape (180,12)

However, when using a plain feedforward ANN, the sequence must be flattened into a large vector of (180 x 12) = 2160 features since it cannot accept 2D sequences. Flattening destroys the temporal relationships which makes it difficult for the model to learn underlying motion patterns. When we attempted this approach, the ANN collapsed into predicting only one class for every sample with accuracy near random chance. This is expected since feedforward ANN models cannot model temporal dependencies after it has been flattened.

We decided to switch to the second approach.

2) ANN on Featured Engineered Inputs:

The second approach involves transforming each window into a feature vector just like how we did when preparing data for our classical machine learning models. In Task 6, we extracted features from the data across all 12 channels which created a low dimensional feature matrix of (n samples x n features). By summarizing each window with each descriptor, we eliminated the need for ANN to learn complex temporal patterns directly from the raw sensor sequences. This feature- based approach led to more stable learning behavior as it achieved an accuracy of 0.8134 and successfully predicted values across all activity values unlike the previous collapsed ANN model.

4.1.2 ANN Results

After training the ANN based on feature based inputs, the model achieved strong and stable performance. Overall, the final ANN achieved:

```
Test Accuracy (ANN-features):  0.8134
 Test Loss (ANN-features):  0.6146
  Macro F1 (ANN-features):  0.8128
 Weighted F1 (ANN-features):  0.8138
```

These metrics indicate that the ANN performs consistently across all activity classes.

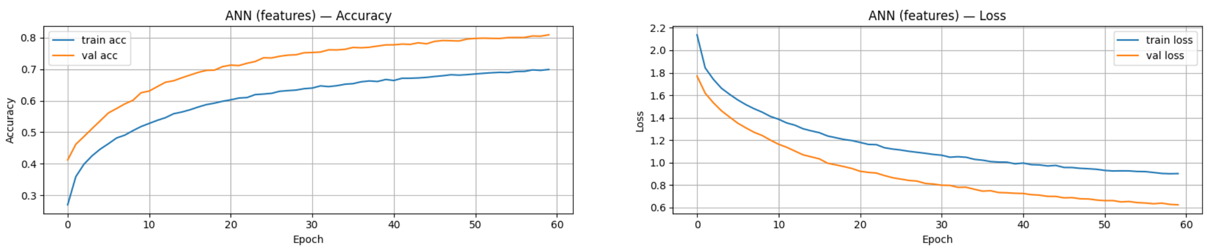

Figure 11: **ANN training history:** (left) accuracy progression showing training accuracy reaching approximately 70% and validation accuracy plateauing around 82%, and (right) loss curves demonstrating steady convergence with validation loss stabilizing near 0.6, indicating effective model learning with minimal overfitting

Based on the accuracy curve in Figure 11, we saw that both the training and validation accuracy curves increase rapidly and converge near the 0.8. We also noticed that the train and validation curves both follow closely which indicate that there was no overfitting. This means that the ANN was well regularized for the feature based input.

For the loss curve, we saw that the training and the validation loss continuously decreased. The validation loss curve was consistently lower than the training loss. This pattern indicates that the model is learning efficiently which is a sign of a stable ANN.

```
Classification report (ANN_features):
              precision    recall  f1-score   support

           F     0.7888    0.8915    0.8370      1152
           G     0.8881    0.8046    0.8443      1213
           H     0.7502    0.7422    0.7462      1218
           I     0.7003    0.7183    0.7092      1207
           J     0.7101    0.7690    0.7384      1134
           K     0.6734    0.8190    0.7391      1271
           L     0.7667    0.6071    0.6776      1191
           O     0.9290    0.8951    0.9117      1258
           P     0.9446    0.9246    0.9345      1273
           Q     0.8661    0.8434    0.8546      1181
           R     0.8921    0.8569    0.8742      1216
           S     0.8954    0.8785    0.8869      1218

    accuracy                         0.8134     14532
   macro avg     0.8171    0.8125    0.8128     14532
weighted avg     0.8181    0.8134    0.8138     14532
```

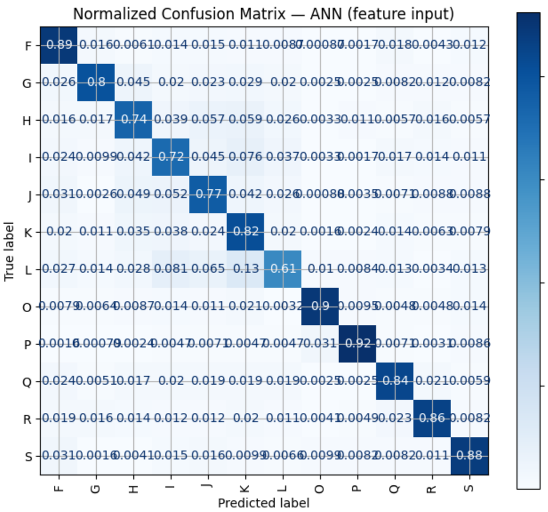

Figure 12: **ANN model performance:** (left) classification report showing per-class precision, recall, and F1-scores with overall accuracy of 81.34%, and (right) normalized confusion matrix demonstrating strong diagonal values with moderate inter-class confusion, particularly among similar eating and hand-motion activities.

The classification report [12] shows strong performance across all classes. High performing activities such as O (playing catch), P (dribbling), R (clapping), and S(folding clothes) achieve particularly strong results of precision and recall above 0.89 and F1 scores between 0.87 - 0.94. Activities such as H(eating soup), I (eating chips), J (eating pasta), and L (folding clothes) show worse performance with mid-range F1 scores between 0.67- 0.75. The model could easily confuse these activities as they are similar in wrist and hand motions. Overall, these reports only add on to how well the ANN is able to generalize across all activity types.

The confusion matrix visually confirms the model's consistent performance as well. The deep blue diagonal line shows that the ANN correctly classifies most samples within each activity. As mentioned previously, the misclassifications are reasonable as the activities misclassified are similar in motion.

Table 3: Updated Model Performance Leaderboard

Model	Accuracy	Macro-F1	Weighted F1
Random Forest	0.9498	0.9494	0.9497
XGBoost	0.9427	0.9422	0.9425
Decision Tree	0.8938	0.8933	0.8937
ANN	**0.8134**	**0.8128**	**0.8138**
SVM (RBF)	0.7445	0.7448	0.7460
AdaBoost	0.3080	0.2931	0.2930
Gaussian Naive Bayes	0.1658	0.1331	0.1349

In the updated leaderboard, the ANN achieves 4th place overall. The ANN model performs worse than the tree-based models but performs better than traditional linear or probabilistic models like SVM or AdaBoost. This implies that the ANN is decently competitive and is a contender to alternative classical models. However, tree-based models are more suited for the WISDM dataset.

4.2 *Advanced Modeling and Evaluation*

As seen earlier, we decided that the standard 80/20 split was ideal for evaluating our models. We chose the 80/20 split because it comes with advantages such as more training data for the model to fully understand complex patterns and better generalization which reduces overfitting. While the 80/20 split has disadvantages including less testing data and risk of overfitting, since our dataset is on the larger side, it is less likely to encounter these issues.

Along with the standard 80/20 split, we also utilized a LOSO evaluation that prevents data leakages by ensuring independence of all samples to evaluate the models' ability to generalize new, unseen individuals. This way, models can be assessed efficiently since they will encounter real-world scenarios.

4.2.1 *LOSO Evaluation*

Model	Accuracy (Mean)	Precision (Mean)	Recall (Mean)	F1–Macro (Mean)
Random Forest	0.448768	0.486951	0.452220	0.437843
XGBoost	0.424375	0.460077	0.427248	0.412767
Decision Tree	0.287512	0.299574	0.289735	0.278226
RBF SVM	0.277843	0.279941	0.277518	0.251198
AdaBoost	0.194326	0.155761	0.191296	0.127509
Gaussian Naive Bayes	0.159545	0.137601	0.158075	0.105032

Figure 13: **LOSO cross-validation performance comparison of classification models.** Random Forest achieves 44.88% accuracy despite the challenging task of generalizing to completely unseen subjects, significantly outperforming Gaussian Naive Bayes at 15.95%, highlighting the difficulty of cross-subject activity recognition..

We grouped by subject IDs and applied LOSO to assess each model's ability to generalize new, unseen individuals, mimicking what the models would encounter in real-world situations.

4.2.2 Window Sizes

Table 4 details the window size in relation to the number of segments per unit length. The segments per unit length normalizes the number of segments by the window size, which tells us how many segments you would get per data point in a window. In short, it measures the resolution against the window size. This helps us compare the different window lengths more equally and how much resolution is lost as window size increases. Resolution demonstrates how detailed each window size is, with higher resolutions being able to detect sudden changes in the data, while low resolution captures longer-term trends.

As window size increases, the segments per unit window decreases. The smaller windows have many segments which indicate that there is higher resolution and more variation captured. On the contrary, the large windows have fewer segments, revealing that there is less resolution and smoother trends.

Table 4: Windowing Statistics: Generated Windows and Segments per Unit Window

Window Size	Windows Generated	Segments per Unit Window
20	653940	32697
40	326969	8174
60	217979	3632
80	163484	2043
100	130787	1307
120	108989	908
140	93419	667
160	81741	510
180	72659	403
200	65393	326

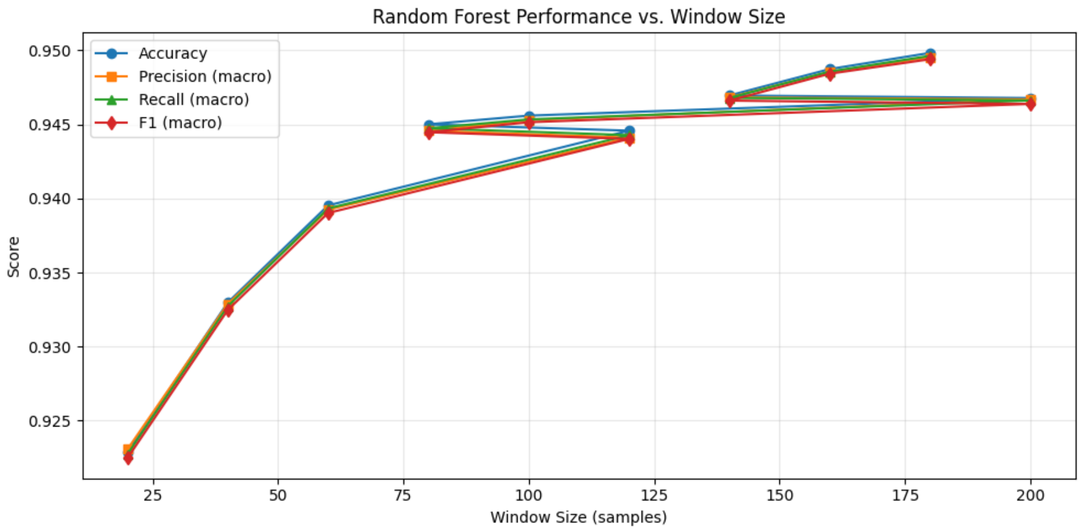

Figure 14: **Random Forest performance metrics as a function of window size.** All metrics (accuracy, precision, recall, and F1-score) improve with increasing window size, plateauing around 140–180 samples. This suggests that larger windows capture more discriminative temporal patterns for activity recognition.

In Figure 14, We decided to further test window sizes against the Random Forest model performance and saw a trend of higher window sizes resulting in higher accuracy. However, the model seems to perform the best at a window size of 180, so we chose this as our window size for training and testing our model. Since a window size of 180 is considered a larger window size, it results in many benefits such as reducing noise, computation, overfitting, and better overall stability. A larger window size of 180 reduces noise because larger windows tend to smooth out noise since data is summarized over a longer span. It also created fewer segments, meaning there would be faster feature extractions, faster training, and smaller memory usage. Lastly, larger windows reduce overfitting and have overall better stability in model predictions since smaller windows tend to output rapidly changing predictions while larger models are stable because each prediction captures entire meaningful sequences.

4.2.3 Metrics Table

window_size	segments	accuracy	precision_macro	recall_macro	f1_macro
180	72659	0.949835	0.949544	0.949637	0.949409
160	81741	0.948743	0.948547	0.948573	0.948430
140	93419	0.946960	0.946828	0.946797	0.946611
200	65393	0.946785	0.946695	0.946620	0.946389
100	130787	0.945600	0.945295	0.945343	0.945152
80	163484	0.945010	0.944582	0.944743	0.944476
120	108989	0.944582	0.944116	0.944280	0.944027
60	217979	0.939536	0.939248	0.939333	0.939023
40	326969	0.932976	0.932881	0.932726	0.932475
20	653940	0.922890	0.923067	0.922728	0.922476

Figure 15: **Random Forest performance metrics across different window sizes.** Performance improves with larger window sizes, peaking at 180 samples with 94.98% accuracy and 94.94% macro F1-score, while smaller windows generate significantly more segments at the cost of lower classification performance.

In Figure 15, the table is sorted by highest to lowest F1-Macro score, with the window size of 180 achieving the highest average F1-macro score of 0.949. This tells us that the window size captures the right amount of motion context, giving the Random Forest model the clearest view of each activity pattern. Overall, it demonstrates that the Random Forest model performs best on average across all activity classes using the window size of 180. This means that the model performs well on minority classes, not just majority ones.

4.2.4 Confusion Matrix

As concluded from the LOSO Evaluation, the Random Forest model is the best fit for our database. To further support this statement, we can analyze the confusion matrix for the Random Forest model. In Figure 16, we see that the diagonal is very strong. This proves that the Random Forest model is able to classify the activities well, with around 900 to 1100 correct predictions, making the overall accuracy of the model extremely high.

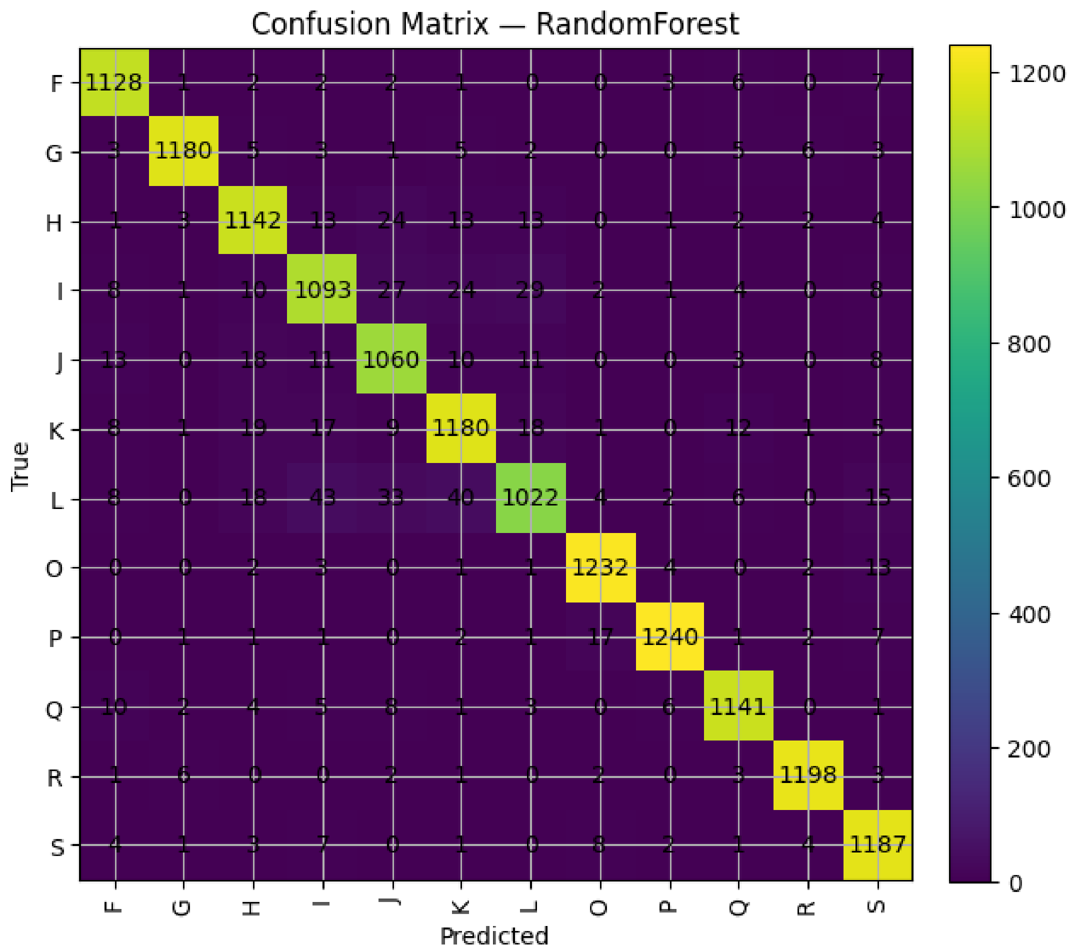

Figure 16: **Confusion matrix for Random Forest classifier showing raw counts.** Strong diagonal values indicate excellent classification performance across all 12 activity classes, with minimal misclassification between activities.

4.2.5 Error Analysis

While the confusion matrix for the Random Forest model has higher accuracy overall, we can also see that there are a few misclassification clusters between activities.

There is a visible trend of misclassifications between the following eating activities: eating soup, eating chips, eating pasta, and eating sandwich. We deduce that these misclassifications occurred because of similar arm-to-mouth motion and similar wrist rotation movements that confused the evaluation model. For example, class I or eating chips and class J or eating pasta, both involve the repetitive scooping of food and bringing the food to the mouth. Since these gestures are similar, it explains why the evaluation model would occasionally confuse the two classes with each other.

Another visible trend of misclassifications is between the following hand activities: typing and writing. Since both involve small, localized wrist motions on a table, it explains why the evaluation model would mistake one for the other.

There are a few other reasons why these errors occurred other than certain activities being inherently similar. One reason would be the large window size that we chose to test the evaluation models on. A window size of 180 paired with a sampling rate of 20 Hz means that we capture 9 seconds of activity per window. A 9-second window can be on the larger side for human-activity recognition, since each window likely contains multiple phases of an action or transitions between activities. This can blur distinctions between similar motions, leading to misclassifications. Furthermore, the Random Forest model does not have temporal logic, meaning it has no memory and no awareness of time. The model treats each window as independent, not linking its activities to windows before it or after. If activities overlap between windows, the model would have a difficult time learning sequences of repeated motion and end up misclassifying an activity.

4.3 *Final Discussion*

To improve upon our model, some recommendations include adding transition classes and focusing on clean labeling.

A transition class is a label used when a window contains multiple activities. This is especially helpful when subjects switch between activities within a short period, resulting in multiple activities within a window. For example, if our 9-second window contains three activities: eating chips, eating a sandwich, and eating pasta, the window should not be labeled as only one of the activities that occurred. Instead, it can be assigned a label like "transition-eating." Some of the benefits of using transition class labels include preventing mixed signals and reducing confusion between similar classes. If windows containing multiple activities are incorrectly labeled as a single activity, the model would assume two activity sequences into one, resulting in the wrong activity classification. Transition classes can isolate that noise. Transition classes also help solve one of the biggest errors we encountered with the Random Forest model. It can reduce confusion between classes by absorbing ambiguous windows that contain more than one activity, which leads to less overlap and improves the accuracy of classifying each activity. It can also help the model learn to detect ambiguous moments in the dataset, which will make the model's final predictions more stable.

Clean labeling can improve the quality of class distinctions through implemented rules. For example, we can set a rule where a window can not contain less than 80% of a single activity, or it will be marked as a transition. This would teach the model to understand that the window doesn't contain one activity and can help the model accurately detect activity changes. We can also set a rule where ambiguous windows are deleted to strengthen the model's clarity. In short, clean labeling helps purify the data to a cleaner version in order to train the model better.

Although we pointed out many errors generated by the Random Forest model, we stand by our decision that it is still the best choice out of all the models we tested. It remains one of the best models for our dataset because of its exceptional ability to handle noise in a dataset and its low maintenance feature of being fast to train, run, and perform computational loads. Despite having no temporal logic, the confusion matrix for the Random Forest model displays high diagonal dominance and strong accuracy across classes, demonstrating its ability to differentiate activities well. Overall, most errors stem from natural similarity between certain activities, further proving Random Forest as a practical choice for our dataset.

5 Conclusion

In conclusion, we visualized and interpreted raw signals of a select number of activities in the WISDM-51 dataset. Afterwards, we performed signal preprocessing to denoise signals, resulting in a reduction in high-frequency spikes and the production of smoother patterns. We chose the sliding-window segmentation strategy over the fixed window strategy and tested ten different window sizes. Our results prompted us to train our evaluation models with a window size of 180, giving us 9 seconds to capture activity per window. We chose to train each model with an 80/20 split, and the resulting accuracy, precision, recall, and F1-macro that was generated led us to conclude that the Random Forest model was the best choice for training. With the Random Forest model, which demonstrated high diagonal dominance and strong accuracy when classifying classes, we were able to differentiate between different hand tasks at a high accuracy rate, as shown in the confusion matrix.

5.1 *Team Contributions*

Member 1 – Sandy Tam:

- Dataset exploration and annotated signal exploration
- Introduction and methodology sections

Member 2 – Nathan Dinh:

- Signal processing and windowing strategies
- Results and discussion sections

Member 3 – Angie Do:

- Feature extraction and analysis
- Literature review section

Member 4 – Hannah Tran:

- Modeling
- Results plots and model descriptions

Member 5 – Emily Liu:

- Advanced modeling and evaluation
- Final discussion and conclusion sections

References

[1] Gary M. Weiss, Jessica L. Timko, Catherine M. Gallagher, Kenichi Yoneda, and Andrew J. Schreiber. Smartwatch-based activity recognition: A machine learning approach. In *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 426–429, 2016.

[2] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, pages 1–17, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[3] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. 46(3), January 2014.

[4] Yujie Dong, Adam Hoover, Jenna Scisco, and Eric Muth. A new method for measuring meal intake in humans via automated wrist motion tracking. *Applied Psychophysiology and Biofeedback*, 37:205–215, 2012.

[5] Yongchun Gu, Yufeng Xu, Yun Shen, Hao Huang, Ting Liu, Linhong Jin, Hongliu Ren, and Jia Wang. A review of hand function rehabilitation systems based on hand motion recognition devices and artificial intelligence. *Brain Sciences*, 12(8):1079, August 2022.

[6] Shreehar Joshi and Eman Abdelfattah. Deep neural networks for time series classification in human activity recognition. In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0559–0566, 2021.

[7] Kyung Rok Pyun, Kangkyu Kwon, Myung Jin Yoo, Kyun Kyu Kim, Dohyeon Gong, Woon-Hong Yeo, Seungyong Han, and Seung Hwan Ko. Machine-learned wearable sensors for real-time hand-motion recognition: toward practical applications. *National Science Review*, 11(2):nwad298, February 2024.

[8] Md Mijanur Rahman, Ashik Uzzaman, Fatema Khatun, Md Aktaruzzaman, and Nazmul Siddique. A comparative study of advanced technologies and methods in hand gesture analysis and recognition systems. *Expert Systems with Applications*, 266:125929, 2025.

[9] Caminate Na Rang, Paulo Jerónimo, Carlos Mora, and Sandra Jardim. Hand gesture recognition using machine learning. *Procedia Computer Science*, 256:198–205, 2025. CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies.

[10] Nisha Dangol and Sazia Mahfuz. A review on sensor-based har models using gnn: Ai in healthcare. *Procedia Computer Science*, 272:368–375, 2025. 16th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 15th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare.

[11] Ravi Kumar Athota and D. Sumathi. Human activity recognition based on hybrid learning algorithm for wearable sensor data. *Measurement: Sensors*, 24:100512, 2022.

[12] Gabriela Ciortuz, Hawzhin Hozhabr Pour, Muhammad Tausif Irshad, Muhammad Adeel Nisar, Xinyu Huang, and Sebastian Fudickar. Machine learning models for wearable-based human activity recognition: A comparative study. *Neurocomputing*, 650:130911, 2025.

[13] Matin Kheirkhahan, Sanjay Nair, Anis Davoudi, Parisa Rashidi, Amal A. Wanigatunga, Duane B. Corbett, Tonatiuh Mendoza, Todd M. Manini, and Sanjay Ranka. A smartwatch-based framework for real-time and online assessment and mobility monitoring. *Journal of Biomedical Informatics*, 89:29–40, 2019.

[14] Sakorn Mekruksavanich, Anuchit Jitpattanakul, Preecha Youplao, and Preecha Yupapin. Enhanced hand-oriented activity recognition based on smartwatch sensor data using lstms. *Symmetry*, 12(9):1570, 2020.

[15] Farhad Shahmohammadi, Anahita Hosseini, Christine E. King, and Majid Sarrafzadeh. Smartwatch based activity recognition using active learning. In *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 321–329, 2017.

[16] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, 22(4):1476, 2022.

[17] Licheng Zhang, Xihong Wu, and Dingsheng Luo. Real-time activity recognition on smartphones using deep neural networks. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1236–1242, 2015.

[18] S. Kundu, M. Mallik, J. Saha, et al. Smartphone based human activity recognition irrespective of usage behavior using deep learning technique. *International Journal of Information Technology*, 17:69–85, 2025.

[19] Ruikang Hu. Human activity recognition using smartphone sensors based on xgboost model. In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 286–292, Chengdu, China, 2024. SCITEPRESS – Science and Technology Publications, Lda. Paper published under CC license (CC BY-NC-ND 4.0).

[20] Dalius Navakauskas and Martynas Dumpis. Wearable sensor-based human activity recognition: Performance and interpretability of dynamic neural networks. *Sensors*, 25(14), 2025.